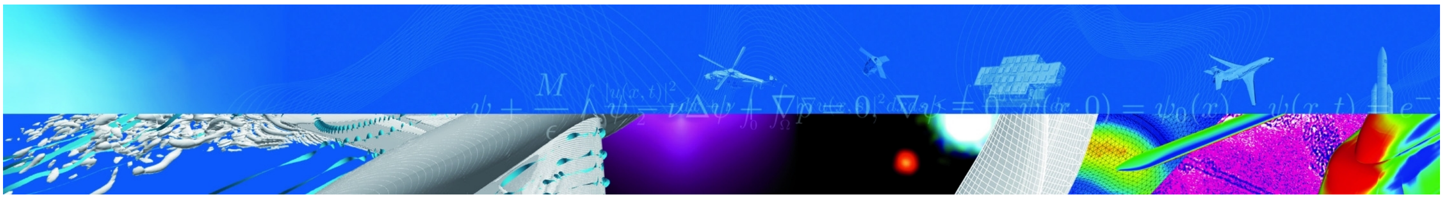


ONERA
THE FRENCH AEROSPACE LAB

return on innovation

TECHNICAL REPORT

SPIS-DUST
Detailed Design Document
and
Software User Manual_v2

Authors : P. Sarrailh (ONERA) ; S. Hess (ONERA) ;

J.-C. Mateo Velez (ONERA) ;

B. Jeanty Ruard (Artenum) ; J. Forest (Artenum)

SPACE ENVIRONMENT DEPARTMENT

RT 1/21505 DESP - June 2016

UNCLASSIFIED
(SANS MENTION DE PROTECTION)

ONERA
THE FRENCH AEROSPACE LAB

BP 4025 - 2, avenue Edouard Belin
31055 Toulouse Cedex - FRANCE
Tél. : 05 62 25 25 25 - Fax : 05 62 25 25 50

SPACE ENVIRONMENT DEPARTMENT

Technical Report N° RT 1/21505 DESP

June 2016

SPIS-DUST Detailed Design Document and Software User Manual_v2

Written by :

P. Sarrailh; S. Hess; J.-C. Mateo Velez; B. Jeanty Ruard (Artenum); J. Forest (Artenum)

Approved by :

Director
Space Environment Department
J.F. Roussel

IDENTIFICATION CARD of ONERA REPORT N° RT 1/21505 DESP

Issued by : **SPACE ENVIRONMENT DEPARTMENT**	Contracting Agency : **ESA**	Contract Number : **4000107327/12/NL/AK**
		Date : **June 2016**

Title :	**SPIS-DUST Detailed Design Document and Software User Manual_v2**
Author(s) :	**P. Sarrailh; S. Hess; J.-C. Mateo Velez; B. Jeanty Ruard; J. Forest**

SECURITY CLASSIFICATION : Civil Title : UNCLASSIFIED (SANS MENTION DE PROTECTION) ID Card : UNCLASSIFIED (SANS MENTION DE PROTECTION) Report : UNCLASSIFIED (SANS MENTION DE PROTECTION)	Timing Classification Off Title : Without object ID Card : Without object Report : Without object

Abstract :

This report details the physical and numerical models implemented in the SPIS-DUST tool in the frame of the ESA Co 4000107327/12/NL/AK. The software simulates the lunar surface and the exploration unit interactions with the space environment, including plasma physics, lunar regolith charging, forces acting on dust, dust ejecta and trajectories. It also simulates the obscuration of lander surfaces. This document also gathers the Software User Manual including Settings and Outputs.

Key words :
LUNAR DUST ; SPIS-DUST

DISTRIBUTION LIST of ONERA REPORT N°RT 1/21505 DESP

Distribution of report

- **Outside ONERA :**

ESA	F. Cipriani	4 ex.
Artenum	B. Jeanty Ruard - J Forest	2 ex.

- **Inside ONERA :**

DESP	A. Champlain - J.C. Matéo Vélez - P. Sarrailh - S. Hess - P. Oudayer	5 ex.
DCT/CID	Documentation	1 ex.

Distribution of identification card

Systematic distribution : DSG, DTG, DAI, DCV/2I .... 4 ex.

URL du fichier = http://ispserveur.onera/onera/upload/RT_1-21505_DESP.1447345710.pdf

Detailed Design Document User Manual

June 2016

Contributors

Pierre Sarrailh
Sébastien Hess
Jean-Charles Matéo-Vélez

ONERA/DESP
2 Av. Edouard Belin
31055 Toulouse Cedex, France

Benjamin Jeanty-Ruard
Julien Forest

ARTENUM
24 rue Louis Blanc
75010 Paris

Document Status Sheet			
Document Title:			
Issue	**Date**	**Author(s) of document/change**	**Reason of change**
1.0	03/07/2014	SH,PS,JCMV	Creation
1.1	01/09/2014	BJR	Section 3 updates
1.2	03/09/2014	SH, JCMV	Typo corrections, small updates
1.3	20/04/2015	SH	Update after Validation and consolidation phases.
2.2	24/06/2016	BJR, JCMV	Includes CCN activities and typos corrections

SUMMARY

1. SUMMARY

1.1. Objectives

This document describes the model developed in the frame of the “Dusty plasma environments: near-surface characterization and modeling” activity under ESA contract number 4000107327/12/NL/AK, whose objective is to set the foundations for the development of a comprehensive suite of near surface dusty plasma environment models applicable to airless bodies of interest for future exploration missions such as Asteroids, Moon, etc... These models shall allow the definition of charged dust environment specifications.

The literature on existing literature and review of harmful effects (TN1/TN2) and the modelling requirement of [MRAD] were used to develop a new set of numerical models, added to the SPIS suite [SPIS].

1.2. This report structure

This document is composed of three parts. Section 3presents the changes made to the SPIS-NUM module and Section 4 those made to the UI module as well as how to use it and guidelines for selecting parameters. Section 5describes the instruments and outputs.

1.3. Acronyms

distribution function	Distribution Function
ESN	Electrical Super Node
ONERA	Office National d'Etudes et de Recherches Aérospatiales
S/C	Space craft
SE	Secondary Electron
SEEE	Secondary Electron Emission from Electron impact
SOW	Statement Of Work
SPIS	Spacecraft Plasma Interaction Software
wrt	with respect to
GIS	Geographic information system

1.4. Reference Documents

[TN1/TN2]	this issue, "Summary of Existing Work, Harmful Effects and Reference Library", July 2013
[MRAD]	this issue, "Modelling requirements", version 1.0, June 2013
[SPIS]	Spacecraft Plasma Interaction Software, www.spis.org
[LSB-1991]	The Lunar Source Book, Cambridge University Press, 1991.
[W-2007]	Adhesion of Lunar Dust, Walton 2007

[S-2002] Sternovsky et al. 2002

[D-2013] Devaud et al., 2013, Lunar Science Institute virtual forum

[UI-TN4] this issue, SPIS-UI Technical note 4.

[Gift wrapping algorithm] Gift wrapping algorithm[electronic resource], Wikipedia, http://en.wikipedia.org/wiki/Gift_wrapping_algorithm, update June 22th, 2013 at 10:29

[Olea 2003] Ricardo A. Olea (2003), Geostatistics for engineers and earth scientists, Norwell, Massachussetts02061 USA, 303p

[Myers 1991] Myers D.E., (1991), Interpolation and estimation with spatially located data: Chenometrics and Intelligent Laboratory Systems, v.11,no. 3,p. 209-228

[Gmsh doc] Gmsh 2.8 [electronic resource], Gmsh, http://geuz.org/gmsh/doc/texinfo/gmsh.html, update July 9th, 2014

[Numerical Recipe] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, “Numerical Recipes Third Edition”, Cambridge University Press, 2007, ISBN : 978-0-521-88407-5, chapter 14 : « Statistical description of data », page 722

2. SPIS-DUST DESCRIPTION

The model developed for plasma, surface and dust interaction is summarized in Figure 1. Inputs and outputs, as well as requirements for each component of the model were described in [MRAD] and divided in six sub-groups:

- AE: Ambient Environment model at mesoscale
- SC: Surface Charging
- DI: Dust Injection from the lunar and asteroid surfaces
- DC: Dust Charging and dynamics in gaseous phase
- MT: Multi Time scale modelling
- OG: Outputs Generation

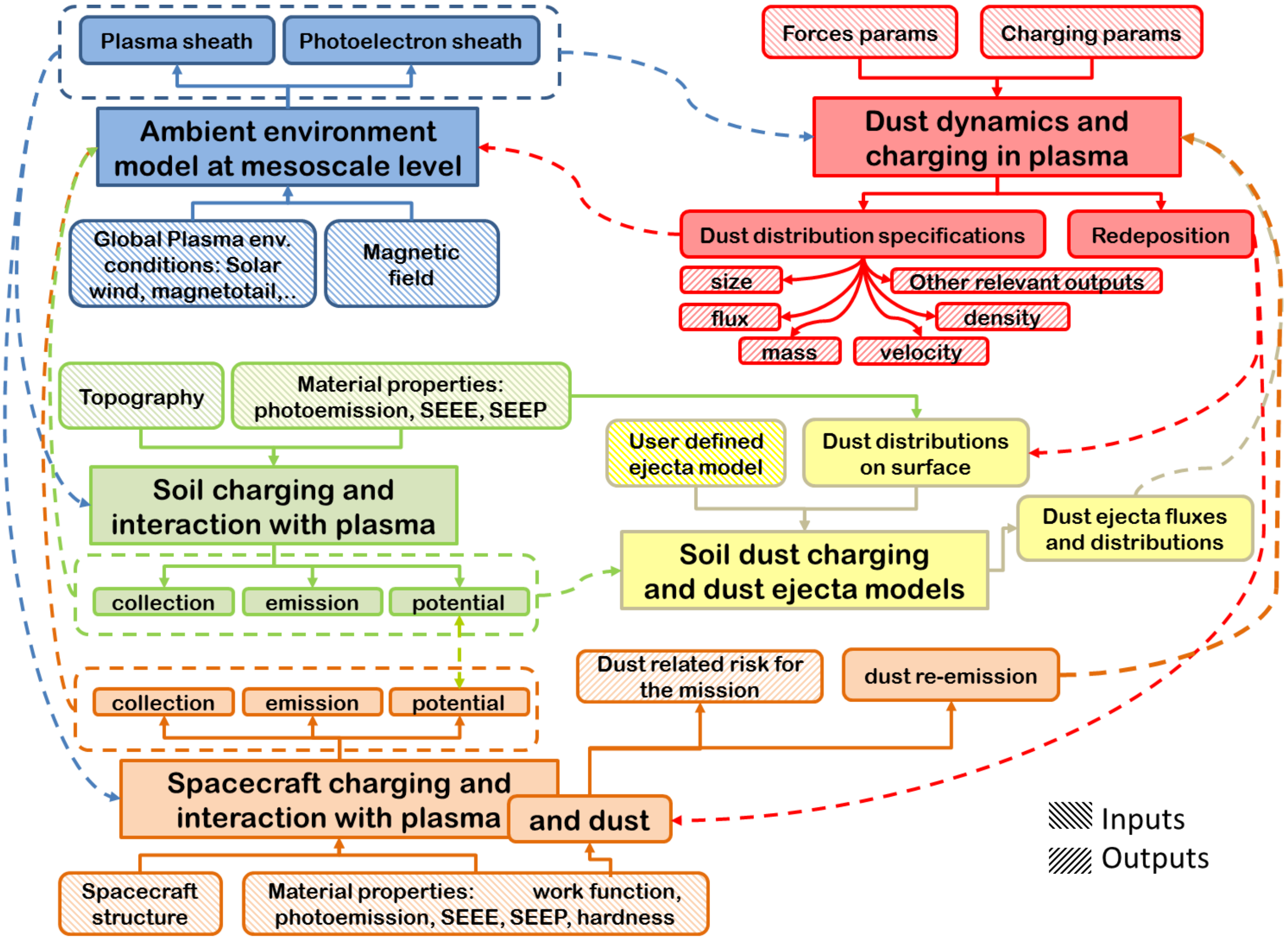

Plasma sheath
Photoelectron sheath
Ambient environment model at mesoscale level
Global Plasma env. conditions: Solar wind, magnetotail,..
Magnetic field
Forces params
Charging params
Dust dynamics and charging in plasma
Dust distribution specifications
Redeposition
size
flux
mass
velocity
Other relevant outputs
density
Topography
Material properties: photoemission, SEEE, SEEP
Soil charging and interaction with plasma
collection
emission
potential
User defined ejecta model
Dust distributions on surface
Soil dust charging and dust ejecta models
Dust ejecta fluxes and distributions
Dust related risk for the mission
dust re-emission
collection
emission
potential
Spacecraft charging and interaction with plasma
and dust
Spacecraft structure
Material properties: work function, photoemission, SEEE, SEEP, hardness
▨ Inputs
▨ Outputs

Figure 1 - Overview of the physical model

Details of numerical developments and model use are provided in the next sections. We choose to present them in the order of appearance of the SPIS modelling chain, which is currently used to define pre-processing settings, run the simulation and present the results.

3. DEFINITION AND SETTINGS OF THE SIMULATION DOMAIN

With respect to classic surface charging modelling, the definition and the settings of the computational domain plasma near a dusty surface presents several differences due to the presence of the ground surface. In order to allow a fine modelling of this surface, extra functions have been developed where the ground surface can be re-built from a cloud of elevation points, coming from a GIS model for instance, as well classic CAD model. A fine kriging algorithm is used to rebuild a consistent tessellated CAD model of the ground by interpolation of the elevation of points. A precise modelling of the lander/ground surface contact points appears also as an important issue. In order to facilitate the point, a semi-automatic procedure of computation of such lander/ground contact point has been introduced. The present section detailed the implemented algorithms and the way to use these new functions.

3.1. Import of the ground topography from a set of points by tessellation and kriging

In this approach and in order to provide the finest description possible, the ground geometry is defined from a cloud of points where z-coordinate corresponds to the local elevation. The rebuilt surface corresponding to this cloud is built by tessellation and interpolation, or kriging, of theses local elevations. This mesh surface is then reconverted into an equivalent tessellated (i.e. triangulated) CAD model than can be used as border of the computational volume. This approach opens the possibility in the future to import realistic ground models coming from GIS system or digital terrain model (DTM). The tessellation and kriging operations have been implemented in the Keridwen tools package as Keridwen commands in normalized bundles.

3.1.1. Loading of the cloud of points

The ground surface is described through a set of (*x*, *y*, *z*) sounding points, where *z* corresponds to the local elevation w.r.t a reference altitude.

3.1.1.1. Data structure and file format

The input cloud of points is defined in an ASCII file and according the format given in the annex 1 of *Physical model and algorithm* [TN2]. The extension of this file is dtm (Digital Terrain Model). This format is summarized here below:

- Meta data: Each word preceded by arrobas, like @keyword, is considered as a meta-data. It can be typically used to store information like author, name, description, units... However, in the current implementation these meta-data are not processed yet;
- Comments: All text following a # character is considered as a comment;
- Node coordinates: Nodes are defined in an orthonormal Cartesian coordinate system (i.e. x y z), z correspond to the local elevation of ground. Each value is separated with a space character. By default, units should be in meters to be compliant with SPIS-NUM.

An example is here below:

```
#start of the file
@author:Artenum
@description:sample of data terrain model of the moon
@name:sample_file
@date:2013-09-12
@unit:m
#nodes coordinates
2 3 8
4.2 6.1 18.0
2.2 3.1 8.8
5.1 2.114 6.4
#end of the file
```

3.1.1.2. Architecture

A dedicated reader has been implemented in the CAD module of Keridwen tools. This reader will temporary convert the loaded data into a list of Penelope’s Points objects. In a second type, these Penelope’s points are converted into GeoPoint objects, as implemented in the Keridwen's CAD module, in order to add extra information, like the local refinement value, needed to perform meshing later on.

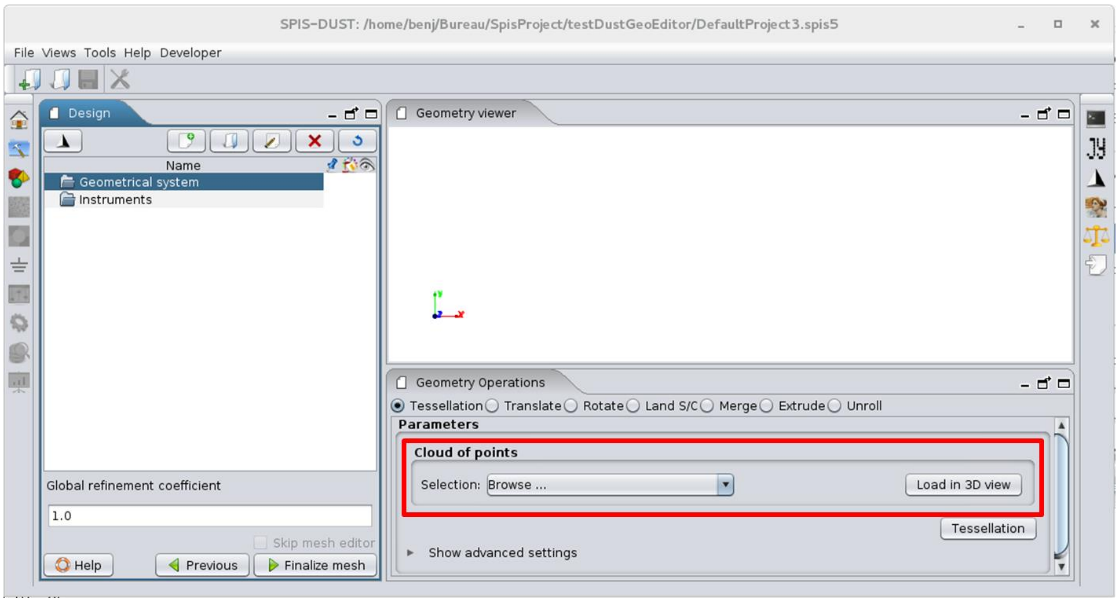
3.1.1.3. Use

The importer is available through the *Geometry Operation* panel of the *Geometry Editor* module and in the *Tessellation* index. Figure 2 shows where the user must define the file where is defined the cloud of points. The selection combo-box indicates let the select the cloud that will be used later for the tessellation. A file chooser in the combo-box allows selecting files on the disk. The selected file should then appear in the combo-box.

The CAD model is loaded in memory after selection in the combo-box only. Default parameters are proposed to generate a rapid tessellation. All of these default values are available in the *"show advanced settings"* panel. It is outlined that this first tessellation can be too rough for a proper modelling of the ground topography and users are allowed to change them to custom/adapt the tessellation.

As soon as the "*Load in 3D view*" button is clicked, the selected cloud of points is displayed in the *Design* panel and in the 3D *geometry viewer*.

The tessellation is performed when the "*Tessellation*" button is clicked. The parameters used by the tessellation are defined in the *show advanced settings* panel.



Global refinement coefficient: 1.0

● Tessellation ○ Translate ○ Rotate ○ Land S/C ○ Merge ○ Extrude ○ Unroll

Selection: Browse ...

Figure 2: selection of file where is stored a cloud of points

3.1.2. *Identification of the external boundary of the cloud*

During the tessellation, triangles will be created to generate a surface. In a first step, the cloud of points is considered in 2D (altitude z is 0). Then Gmsh software will be used to generate a triangulated surface where all points will be defined with a null altitude (z = 0). Nevertheless, a closed surface must be defined to compute a triangulate surface with this software. The current step will describe the computation of the boundary of the 2D cloud of points.

3.1.2.1. *Algorithm and theory*

The boundary of the cloud of points can be computed with the Gift wrapping algorithm [Gift wrapping algorithm]. This algorithm has a complexity in O(nm), where n is the number of points defined in the cloud of points and m the number of point on the boundary. The Gift wrapping algorithm is composed of the following steps:

- Select the point of the cloud with a minimal abscissa and put it in the boundary
- While the boundary is not closed
    - Loop on all points of the cloud points, search the next point on the boundary
        - Select 2 points from the cloud of points
        - Compute the direction of the triangle
        - If the triangle is clockwise, the second point of the triangle is considered on the boundary
        - Else if the triangle is counter clockwise the third point of the triangle is on the boundary
    - End loop
- End while

This algorithm is implemented in the CAD module of Keridwen tools. The computation of the boundary of the 2D cloud of points is automatically performed during the tessellation and like the cloud loading does not required any specific action from the user.

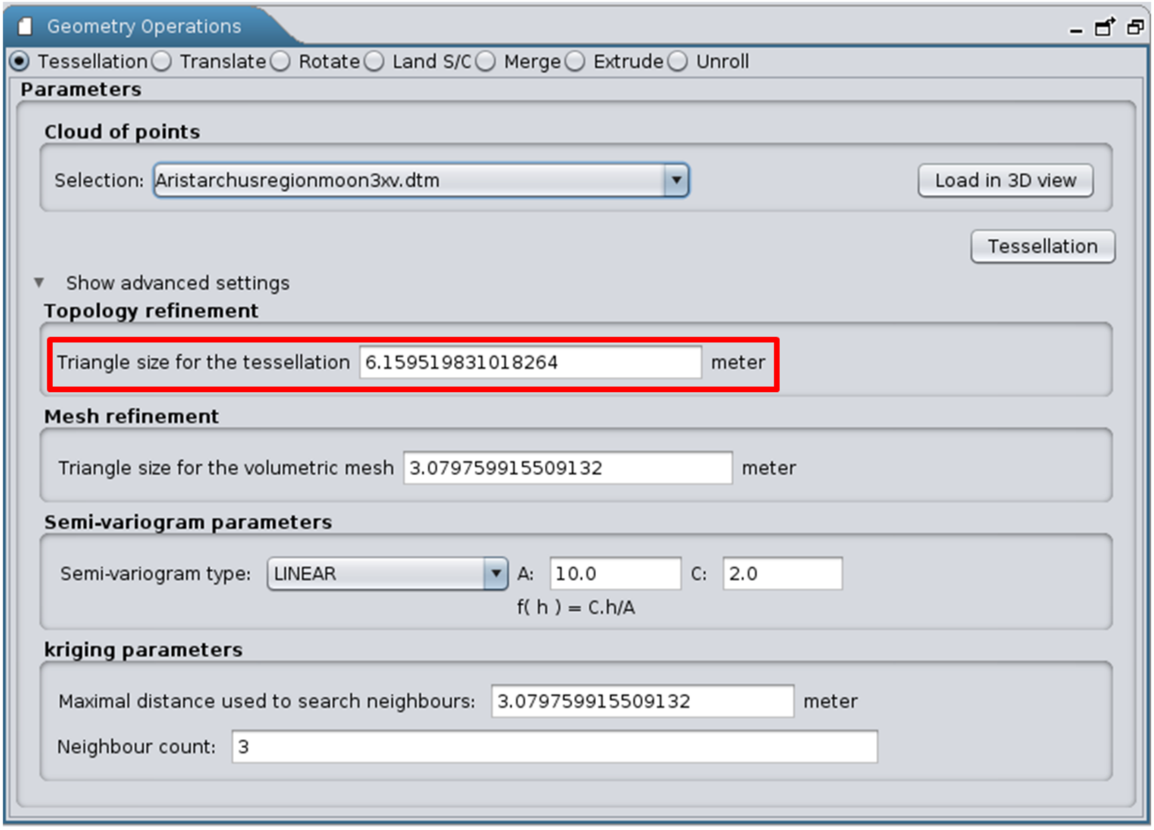
3.1.2.2. *Limitations of use*

The present algorithm requires at least 3 points to identify and define the boundary.

3.1.3. Tessellation refinement

3.1.3.1. *Principle*

During the tessellation, triangles will be created to generate the tessellated surface. A surface is defined after the computation of the boundary of the cloud of points. Gmsh software will be used to triangulate this surface. This software will generate a mesh to accomplish this task. Gmsh used a refinement coefficient associated to all points of the cloud of points to compute the size of all triangles of the mesh. This value is defined by the user. Figure 3 shows where the user must define this tessellation refinement in the extended panel after having selected “*show advanced settings”*.



● Tessellation ○ Translate ○ Rotate ○ Land S/C ○ Merge ○ Extrude ○ Unroll

Parameters

Cloud of points

Selection: Aristarchusregionmoon3xv.dtm

Topology refinement

Triangle size for the tessellation 6.159519831018264 meter

Mesh refinement

Triangle size for the volumetric mesh 3.079759915509132 meter

Semi-variogram parameters

Semi-variogram type: LINEAR A: 10.0 C: 2.0

f(h) = C.h/A

kriging parameters

Maximal distance used to search neighbours: 3.079759915509132 meter

Neighbour count: 3

Figure 3: definition of tessellation refinement

3.1.3.2. Remarks

- The local mesh size might be locally impacted by the distance between two neighbour points on the border. If two points of the boundary are closer than the given refinement factor, the local cell size will be equal to the distance between these points.
- More the triangle size is small more the CPU time used to perform the tessellation will be high.

3.1.4. Kriging interpolation

The kriging interpolation constitutes one of the key operations of construction of the geometrical model of the ground. Its purpose is to apply the relevant z-axis value on each vertices of the mesh according the local elevation of the nearest and most relevant points.

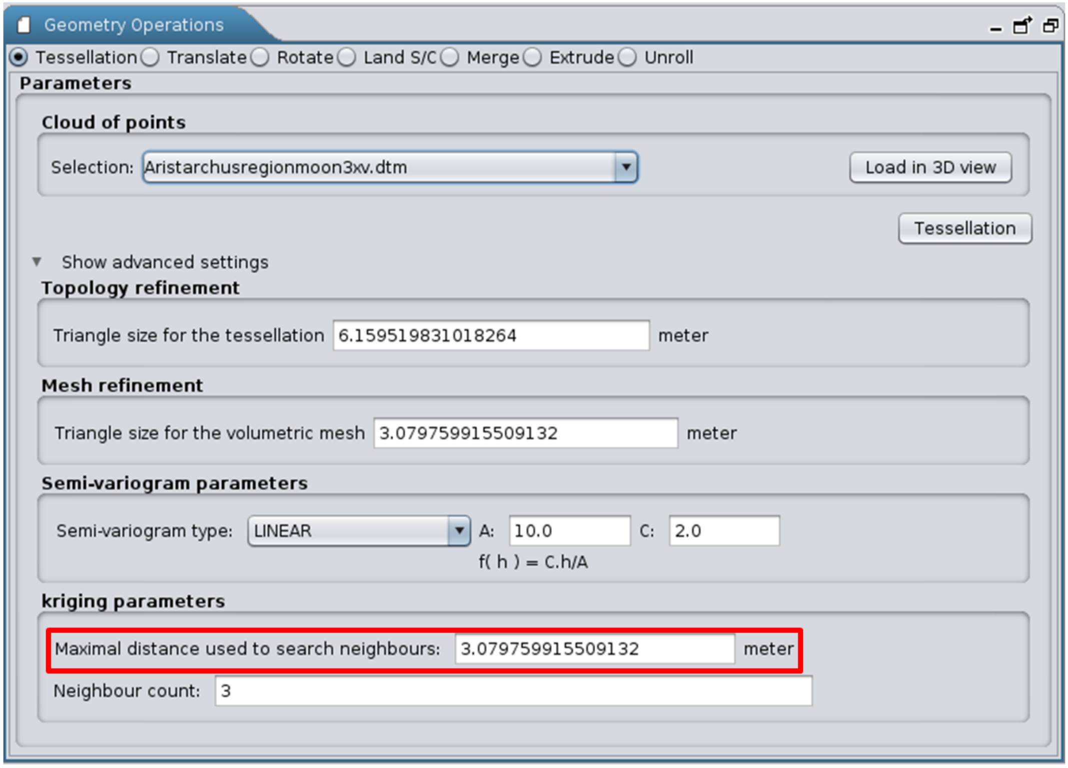
3.1.4.1. Algorithmic and theory

The kriging interpolation is an interpolation technique regularly used in geostatistics. The interest of this interpolation method is that it takes into account both the distance between the point where the altitude must be computed and its neighbours but also the distance between neighbouring pairs. To precisely achieve this interpolation, the algorithm described page 51 in [Olea, 2003] is applied and is decomposed in the following major steps, considering that user's inputs might set through the GUI as illustrated in Figure 4.

- First, the altitude of the point x0 is searched;

- Neighbours of x0 are searched. The aim is to select only relevant observations to compute the altitude of each vertex. This research is more subtle than search the n neighbours the closest of x0. Indeed, in order to improve the radial data distribution selection of a search neighbourhood can be based on a more elaborate scheme. Quadrants are used to select neighbours. A maximum distance for the research of neighbours and a maximal number of neighbours searched by quadrants should be manually set by the user. In very general terms, the neighbourhood must be large enough to contain at least 3 observations and no more than 25 considered neighbours [Myers 1991]. Take care that the increase of the number of neighbour, as well the distance used to search neighbours, will significantly impact the total CPU time used to perform the tessellation. The user is warned to modify these parameters with care. If no neighbour is found, then the vertex associated to x0 is removed from the mesh. There are several possibilities to find neighbours. First, the distance between the point x0 and each point defined in the cloud of points is computed. This computation is not optimal. Here a virtual regular grid is created.. For each cell, the list of inside points is built. The virtual grid is used to search the neighbour from the coordinate of x0.
- A covariance matrix K is computed where each value (i,j) correspond to a covariance factor function of the distance between the neighbours x_i and x_j. This term takes into account in the interpolation the distance between neighbouring pairs.
- A covariance vector K0 is computed where the component i correspond to a covariance factor function of the distance between the point x0 and its neighbour xi. This term takes into account in the interpolation the distance between the point where the altitude is computed and its neighbours.
- An altitude vector Z is defined where the i-component corresponds to the neighbour elevation xi.
- The altitude of the point x0 is computed with the formula:
  $$z_0^* = Z^t K^{-1} K_0$$
- Matrices K and K0 are defined from a covariance factor function. In our case, a semivariogram is used because the semivariogram is a statistic that assesses the average decrease in similarity between two random variables as the distance between the variables increases, leading to some applications in exploratory data analysis. Moreover, the semivariogram requires a less stringent assumption for its existence than the covariance. To finish, estimation of the semivariogram is insensitive to the addition of a constant to the random function, whereas the covariance estimator depends on such a constant [Olea 2003]. There are several semivariogram types. Two semivariogram types are possible:
  - Linear semivariogram: $\gamma(h) = C\dfrac{h}{a}$ where h is a distance, C is the sill and a the range;
  - Exponential semivariogram: $\gamma(h) = C(1 - e^{-\frac{h}{a}})$ with*h*a distance, *C*the sill and *a* the range.

Some of the inputs may depend on the characteristics of the input cloud of points and an iterative process might be needed to find the best setting. This algorithm is implemented in the CAD module of Keridwen.



Geometry Operations

● Tessellation ○ Translate ○ Rotate ○ Land S/C ○ Merge ○ Extrude ○ Unroll

Parameters

Cloud of points

Selection: Aristarchusregionmoon3xv.dtm ▾ Load in 3D view

Tessellation

▾ Show advanced settings

Topology refinement

Triangle size for the tessellation 6.159519831018264 meter

Mesh refinement

Triangle size for the volumetric mesh 3.079759915509132 meter

Semi-variogram parameters

Semi-variogram type: LINEAR ▾ A: 10.0 C: 2.0

f(h) = C.h/A

kriging parameters

Maximal distance used to search neighbours: 3.079759915509132 meter

Neighbour count: 3

Figure 4: Setting panel of the kriging and semivariogram parameters

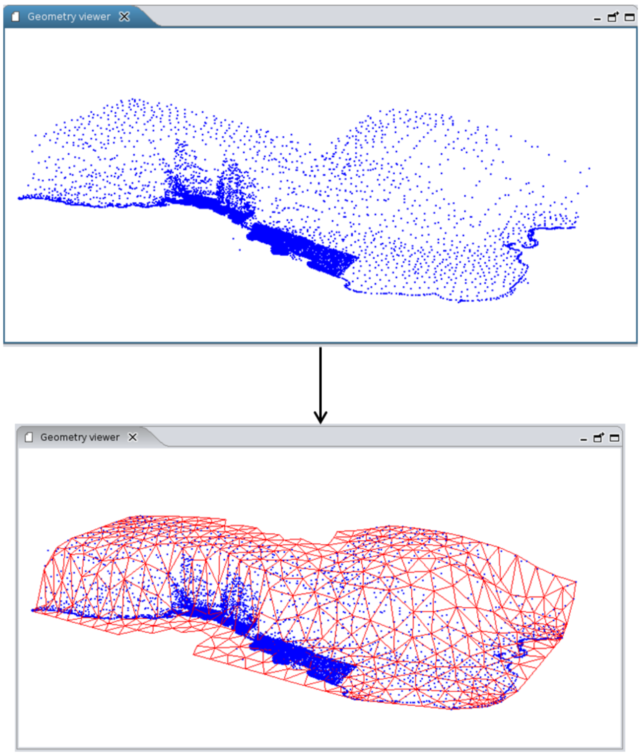


Figure 5: sample of kriging result

3.1.5. Mesh to tessellated CAD model conversion

In order to be used as boundary support for the meshing of the computational volume, the above generated ground mesh is converted into an equivalent CAD model using the Penelope's mesh to convert capabilities (see MeshToGeoExporter class). Figure 6 shows an example of generated CAD model. A global refinement factor can be set by the user and applied to whole generated model. More this parameter is small, more the mesh generated from the CAD model will be big. A Gmsh physical [Gmsh doc] is automatically built for the entire created surface.

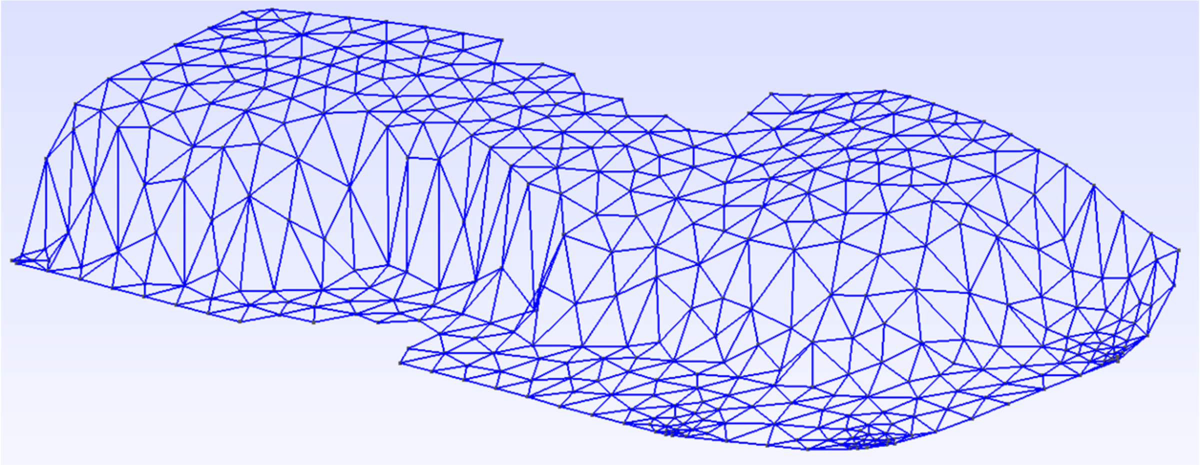
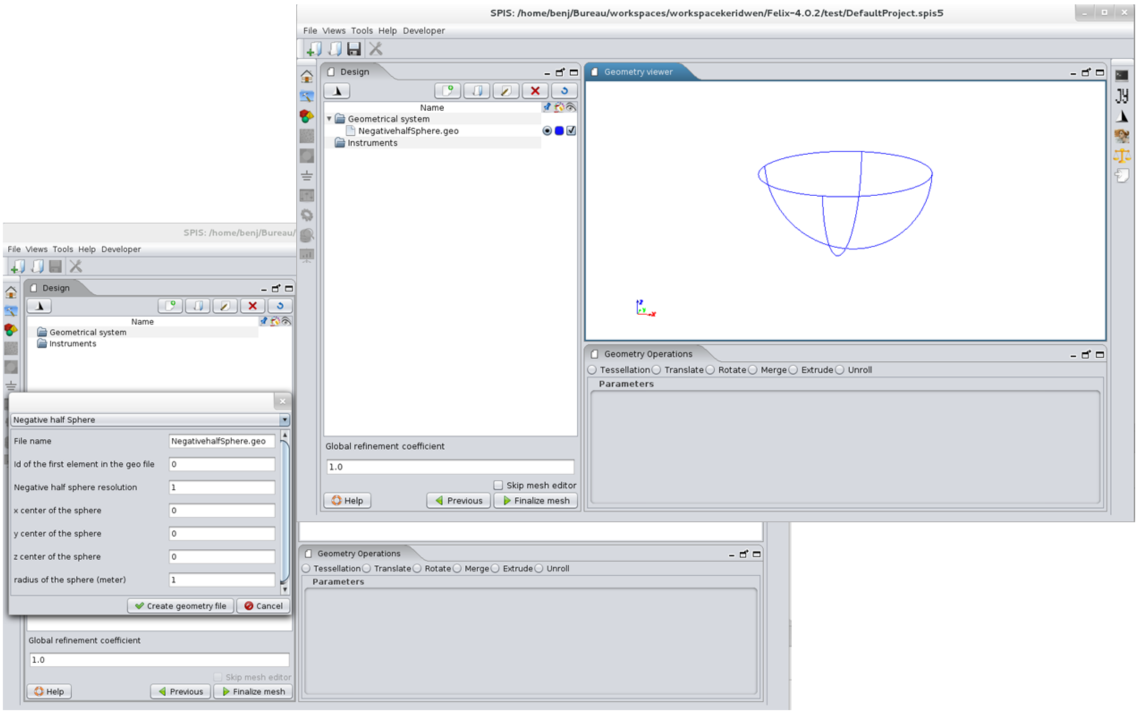


Figure 6: equivalent CAD model computed from ground mesh

3.2. Direct import of the ground geometry from a CAD model.

A direct import of ground geometry from a CAD model remains possible by loading of a classic Gmsh geometry file (.geo). Such structured can be modelled by hand following the classic Brep approach as illustrated in Figure 7or issued from the output of the above approach defined in section 3.1.

To help the user in the modelling of most critical Lunar ground configuration (e.g. plane surface, simple craters…) a set of specific templates has been defined and added to the default ones.



Negative half Sphere

File name	NegativehalfSphere.geo
Id of the first element in the geo file	0
Negative half sphere resolution	1
x center of the sphere	0
y center of the sphere	0
z center of the sphere	0
radius of the sphere (meter)	1

Global refinement coefficient: 1.0

Figure 7: sample of template utilization

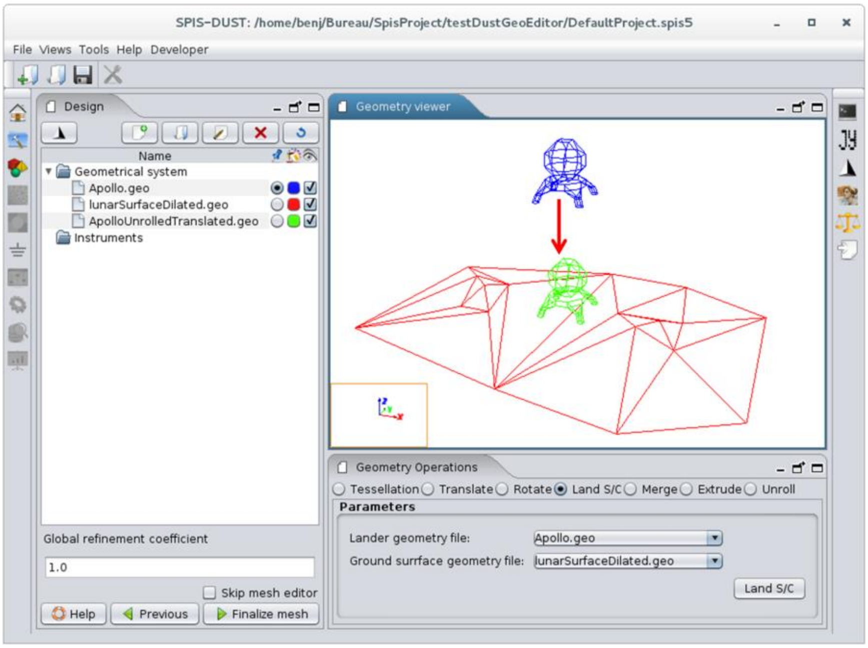
3.3. Integration of the lander in the modelling volume

To simplify the modelling of the lander and its interaction with the ground a set of additional CAD and geometrical operations have been introduced. Practically, these operators are generic and can be used for other applications cases and the edition of all CAD models in SPIS.

3.3.1. Geometrical operators: translation, rotation and spacecraft landing

Translation rotation and landing operators allow placing easily the lander at the right place in the computational volume. Geometrical operators take in input all standard Gmsh .geo file. Operations are performed directly in memory through a Keridwen GeoModel. By default, operators “unroll” the generated geometries, i.e. perform the operations, and generate a `flat' geometry in output, with an explicit list of all elementary entities [Gmsh doc]. The final output appears into a new file with “Unrolled” and the operation name happened to the initial name and loaded in the global geometrical model of SPIS. The original file remains unchanged. The unroll operator is also available by its own button. The following figure Figure 8 shows an example of geometrical operations, here a landing. Settings of all of these operators are available through the *Geometry Operation* panel in the *Geometrical Editor*. All operators are based on the generic geometrical operators of the CAD and org-Keridwen-modelling-gmsh modules of Keridwen Tools.

The *land S/C* operator uses the ray-tracer engine of the Penelope library to compute the minimal distance between the lander and the ground surface. After computation of the minimal distance between the two geometrical models, an automated translation along the z-axis is performed.


SPIS-DUST: /home/benj/Bureau/SpisProject/testDustGeoEditor/DefaultProject.spis5
File Views Tools Help Developer
Design
Name
Geometrical system
Apollo.geo
lunarSurfaceDilated.geo
ApolloUnrolledTranslated.geo
Instruments
Geometry viewer
Geometry Operations
○ Tessellation ○ Translate ○ Rotate ● Land S/C ○ Merge ○ Extrude ○ Unroll
Parameters
□ Skip mesh editor
Help
Previous
Finalize mesh
Land S/C

Global refinement coefficient: 1.0

Lander geometry file: Apollo.geo

Ground surrface geometry file: lunarSurfaceDilated.geo

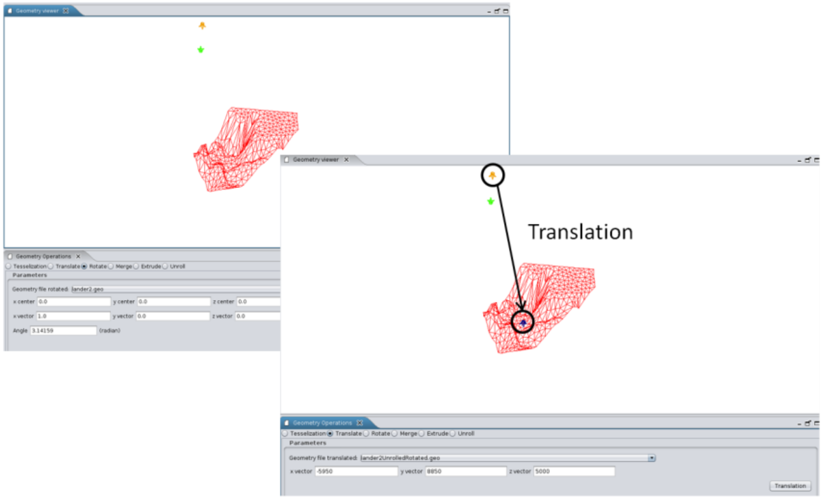
Figure 8: Example of landing operation.

Important remark:

- Translation and rotation operations are computed on the surface of the Gmsh .geo file. All plane and ruled surfaces are computed from the GeoModel previously built. However, the unrolled function fails in Gmsh if a hole is defined on a ruled surface. The unroll operation is automatically called before translation and rotation operations and can introduce some problems on these surfaces.
- Please take into consideration that if a geometry file external to the spis-project is loaded to be edited, the output files resulting of geometrical transformation requested by the user, will be saved in the same source directory. This means that, first, this source directory must be writable and, in second, source and edited geometry files will be saved into the current spis-project only when this one will be saved on user’s demand (button with the floppy icon).
- The *land S/C* operator is based on the Penelope ray tracer where the minimal distance is computed between the lander surfaces and the ground surfaces. The source of the ray tracer is the lander surfaces and the target is the lunar surface. The source is a mono directional source along the minus z-axis.

3.3.1.1. Limitations of use

The present version of the Gmsh .geo format of the Keridwen's CAD module still presents a few limitations and operations declared inside the input files are not supported. We recommend in input only flat geometries.


Geometry viewer
Geometry Operations
○ Tesselization ○ Translate ● Rotate ○ Merge ○ Extrude ○ Unroll
Parameters
Translation
Geometry Operations
○ Tesselization ● Translate ○ Rotate ○ Merge ○ Extrude ○ Unroll
Parameters
Translation

Geometry file rotated: lander2.geo
x center 0.0 y center 0.0 z center 0.0
x vector 1.0 y vector 0.0 z vector 0.0
Angle 3.14159 (radian)

Geometry file translated: lander2UnrolledRotated.geo
x vector -5950 y vector 8850 z vector 5000

Figure 9: translation sample

3.3.2. Union of the lander and the simulation domain

We remind that to build a self-consistent volume mesh, the external boundary should be continuous. Because the lander and/or the exploration units are positioned on the lunar surface, the intersection of surfaces of both geometries should be carefully computed. To do it, a specific merge operator has been developed on the basic of generic operators of Keridwen Tools.

From two different CAD models (i.e. the lander and the ground surface) defined in two different files, the merge operator will identify facing face-elements and try to fuse them in order to build-up a continuous surface. Various criteria of facing faces have been integrated to address the various ground configurations. The operator tries to avoid small voids or sharp edges to be left between the surfaces that could lead to numerical difficulties.

3.3.2.1. Process

Firstly, lander/exploration unit and the surface ground should be defined in two different CAD models, i.e. two different Gmsh .geo files, and are selected by the user.

Secondly, the lander faces to merge must be defined in a specific Gmsh physical surface and manually selected. As soon as the Gmsh physical surfaces to merge are selected in the relevant combo-box, the Gmsh physical surface is automatically highlighted in the *3D view*, as illustrated in Figure 10.

Thirdly, the detection of the face couples to be merged is done by clicking on the "*Compute faces to merge*" button. Each computed face pair to merge is highlighted with the same colour.

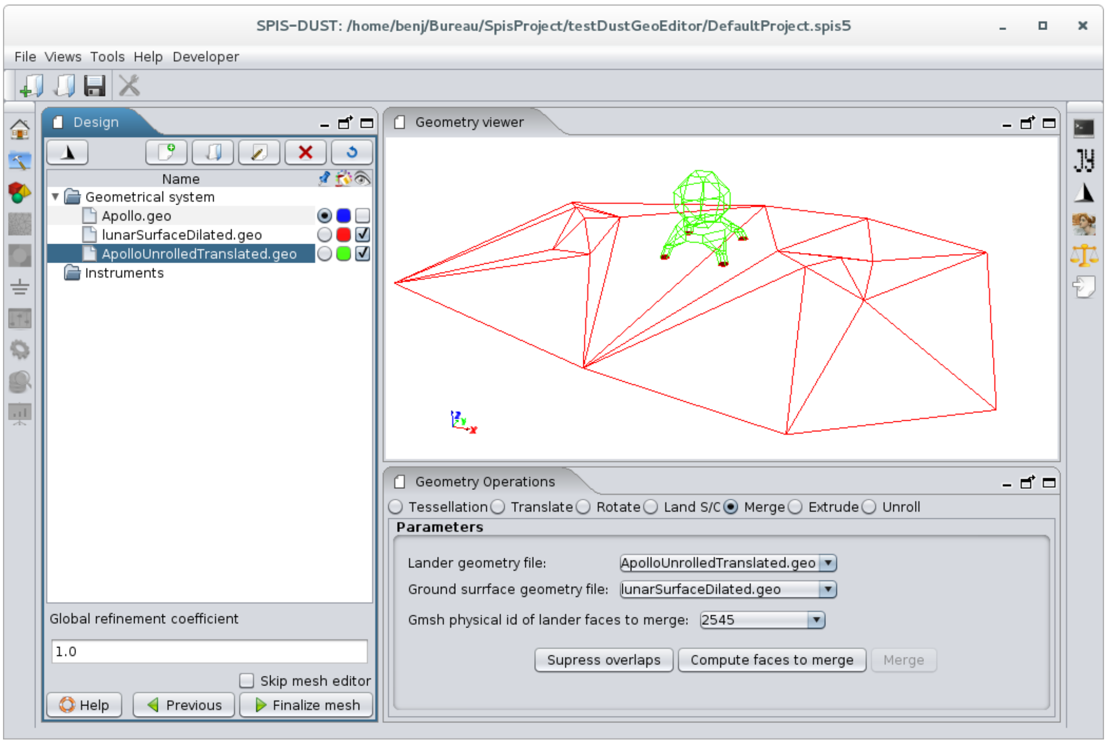
If the projection of a lander face seems to overlap several ground faces the "*Suppress overlaps*" button will detect and suppress them.

Last, the user applies the merging by clicking on the "*Merge*" button.

3.3.2.2. Algorithmic and theory

Distances between faces from the selected Gmsh physical surface and ground faces are computed the ray-tracing engine of Penelope. Please notice that ruled surfaces are considered as plane ones so some unrealistic distance computation may happen.

A couple of faces to merge is created for each face from the Gmsh physical surface. This pair of faces stores the lander face to merge and its closest ground surface. Both surfaces are suppressed. A new ruled surface is created considering the line loop of the inital ground surface as external boundary and the line loop of the lander face to merge as the inner loop. The generated ruled surface will correspond to a fillet making the link between both facing surfaces. To finish a Gmsh physical is associated to the generated fillet. The whole system is saved in Gmsh .geo file called "merged.geo" containing both initial files to be merged and the ruled face of the fillet. Figure 10 here below summarizes this approach.



○ Tessellation ○ Translate ○ Rotate ○ Land S/C ● Merge ○ Extrude ○ Unroll

Parameters

Lander geometry file: ApolloUnrolledTranslated.geo

Ground surrface geometry file: lunarSurfaceDilated.geo

Gmsh physical id of lander faces to merge: 2545

Global refinement coefficient: 1.0

□ Skip mesh editor

Figure 10: highlighted Gmsh physical surface to merge

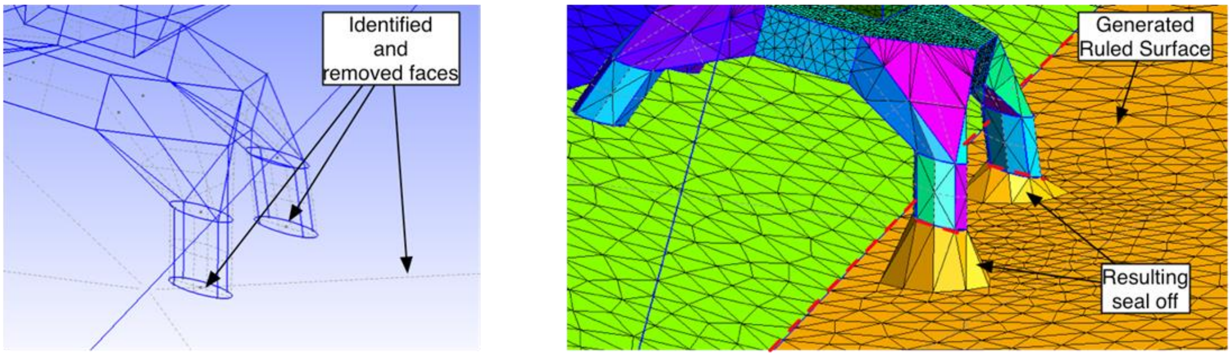


Figure 11: Principle of the merge operation and creation of the fillet at the connection of both models.

As illustrated here after, tests have confirmed the global robustness of the algorithm. However, the validation of the merge remains on the responsibility of the user. More especially, it is recommended to avoid situations where the selected face on one CAD model is almost best rides two faces of the other model. The fillet might be strongly twisted or malformed and inducing anomalies in the volume mesh. Use the "*Suppress overlaps*" button to avoid these potentially dangerous overlaps.

NB:

- We remind the current implementation of the Gmsh .geo file loader of Keridwen Tools still presents several limitations and it is recommended to load only "flat" geometry without operations declared. The merging operator is part of the Boolean operators of the CAD module of Keridwen Tools.
- The definition of spacecraft physicals on the inputs Gmsh's geo files is the responsibility of the user and must be defined BEFORE the merging operation.
- You must not have any other contact point between the spacecraft and the ground surface, excepted the surface elements resulting from the merge operation. Otherwise, the volume mesh will be inconsistent and, in most of the cases, the Gmsh will produce an error at the meshing phase.
- The overlaps detection and the computation of faces to merge are based on the ray tracer implemented on Penelope. The source of the ray tracer is the lander surfaces to merge and the target is the lunar surface. The source is a mono directional source along the minus z-axis so if the z-coordinates of lander faces to merge are lowest than ground faces, no merging is possible.
- It is possible to define one or several Gmsh physical surfaces where the faces to merge are defined. If n Gmsh physical surfaces to merge are defined then users have to process n times the merging operation.

3.3.2.3. Use

Figure 12 gives an example of the control GUI of the merging function. The whole process is decomposed into the following steps:

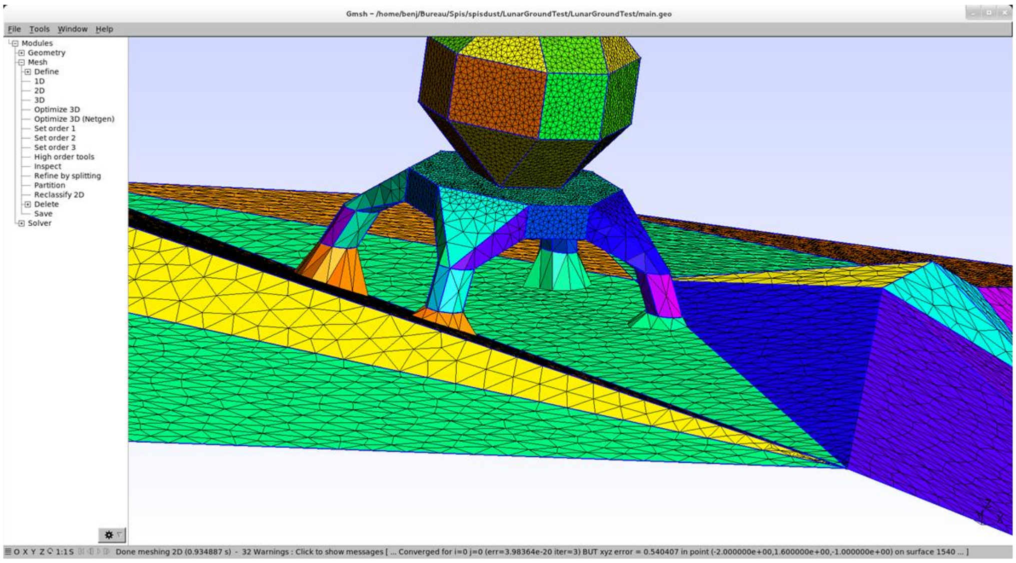
1. Selection of the input lander CAD model;
2. Selection of the input ground surface CAD model;
3. Selection of the Gmsh physical lander surface to merge;
4. Detection and suppression of overlapped ground surface by clicking on the "*Suppress overlaps*" button (optional);

5. Compute and 3D visualization of faces to merge by clicking on the "*Compute faces to merge*" button;
6. Validation of the merging by clicking on the "merge" button.

The generated geometrical structure should appear in the Design panel on the left.

Figure 12 gives an example of merged surface in the Gmsh mesher. The generated fillets at the feet of the landing gear are clearly visible.

Caution! The merging is possible only between two Gmsh .geo files defining surface and which do not contain any volume.



```
Done meshing 2D (0.934887 s) - 32 Warnings : Click to show messages [ ... Converged for i=0 j=0 (err=3.98364e-20 iter=3) BUT xyz error = 0.540407 in point (-2.000000e+00,1.600000e+00,-1.000000e+00) on surface 1540 ... ]
```

Figure 12: Visualization of merging result in Gmsh with the generated fillets.

3.4. Finalisation of the computational volume definition: extrusion of the ground surface

The computational volume can be defined manually as usual. However, it is also possible to define it by simple vertical extrusion of the ground surface with the introduction of the extrusion operator of Keridwen.

3.4.1. Algorithm

The algorithm used to perform the extrusion of a surface is not based on Gmsh internal transformations, like for rotations or translations, but is directly performed in Keridwen Cad Model. This is due to the necessity to create additional Gmsh's physical groups [Gmsh doc] for the sides and the top of the computational volume. The algorithm is based on the following approach:

1. All edges which define the boundary of the surfaces are identified;
2. For each point defined on the boundary, a new one is created with the same x and y coordinates but with the altitude z defined by the user. A vertical edge line is then created between the new point and the original point to draw the vertical border. A horizontal line is created between two successive newly created points at the top.

3. A vertical plane surface is created on the basic of the constituted line loop.
4. The top surface is created from all lines created between points of the top.
5. Last, a volume is automatically created from these whole surfaces.

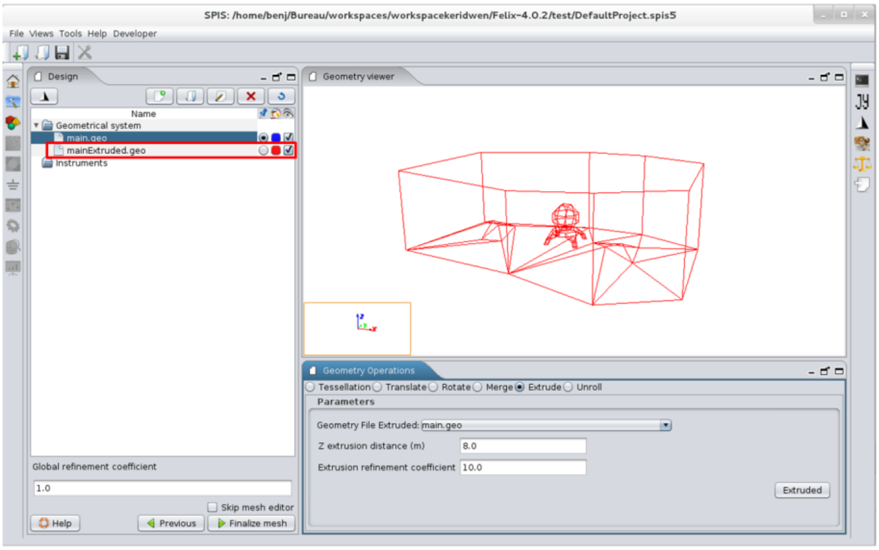
A first Gmsh physical surface is created for all lateral faces newly created, a second one is created for the top surface and a third one is created for the volume newly created.

Remarks:

- As for previous operation, the input file is considered as an unrolled geometry. This step is usually performed after the merging operation. We remind that the unroll operation performed by Gmsh currently presents a bug a hole is still present in the geometry to be unrolled. So unlike rotation or translation operations, there is no automatic unroll operation that is performed at the beginning of the algorithm.
- The merging operation can generate some specific tetrahedra not supported by Spis Num (low thickness but high height) and it can be necessary to use Delaunay algorithm in Gmsh software to generate the mesh.

3.4.2. Use

Figure 13 shows the setting panel for the extrusion and an example of result. The first parameter used is the Gmsh .geo file to extrude. The second is the z extrusion distance defined in meter. The extrusion is performed along the z-axis. In order to adapt the mesh refinement in function of the altitude, it is possible to define a different mesh refinement factor for elements at the top of the volume.



SPIS: /home/benj/Bureau/workspaces/workspacekeridwen/Felix-4.0.2/test/DefaultProject.spis5

Geometrical system
- main.geo ● ☑
- mainExtruded.geo ○ ☑

Instruments

○ Tessellation ○ Translate ○ Rotate ○ Merge ● Extrude ○ Unroll

Parameters

Geometry File Extruded: main.geo

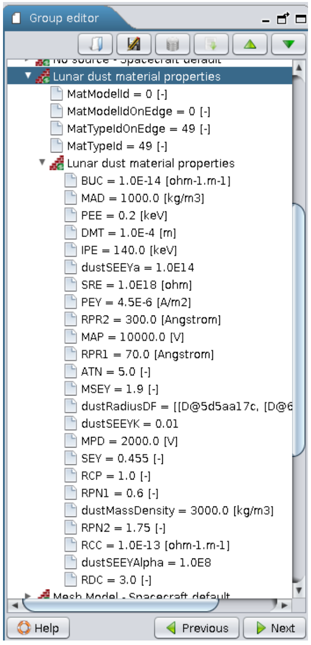
Z extrusion distance (m) 8.0

Extrusion refinement coefficient 10.0

Global refinement coefficient 1.0

□ Skip mesh editor

Figure 13: extrusion result



```
No source - Spacecraft default
Lunar dust material properties
    MatModelId = 0 [-]
    MatModelIdOnEdge = 0 [-]
    MatTypeIdOnEdge = 49 [-]
    MatTypeId = 49 [-]
  Lunar dust material properties
      BUC = 1.0E-14 [ohm-1.m-1]
      MAD = 1000.0 [kg/m3]
      PEE = 0.2 [keV]
      DMT = 1.0E-4 [m]
      IPE = 140.0 [keV]
      dustSEEYa = 1.0E14
      SRE = 1.0E18 [ohm]
      PEY = 4.5E-6 [A/m2]
      RPR2 = 300.0 [Angstrom]
      MAP = 10000.0 [V]
      RPR1 = 70.0 [Angstrom]
      ATN = 5.0 [-]
      MSEY = 1.9 [-]
      dustRadiusDF = [[D@5d5aa17c, [D@6
      dustSEEYK = 0.01
      MPD = 2000.0 [V]
      SEY = 0.455 [-]
      RCP = 1.0 [-]
      RPN1 = 0.6 [-]
      dustMassDensity = 3000.0 [kg/m3]
      RPN2 = 1.75 [-]
      RCC = 1.0E-13 [ohm-1.m-1]
      dustSEEYAlpha = 1.0E8
      RDC = 3.0 [-]
Mesh Model - Spacecraft default
```

Figure 14: Lunar dust material properties

3.5. Definition of the materials

3.5.1. Dusty surfaces

3.5.1.1. Default materials

The object surface properties are defined at group editor level. After selecting a physical group (several groups are possible for the surface, as e.g. the crater and the flat sides), the user must select the electrical super node (possibly different from one surface to another) and the material. We recommend using the default material properties listed as:

- Lunar dust material
- Conductive lunar dust material

They have the same properties except their conductivities. The conductive lunar dust material is a perfect conductor, allowing to greatly reducing the computational time.

3.5.1.2. Radius and mass density

The default mass density of dust is taken as 3000 kg/m^3. This value can be modified using the **dustMassDensity** local parameter in the group editor.

The default dust size distribution is taken from [LSB-1991], p306, referred as 71501,1 Mare and presented in Figure 15. The initial plot did not clearly present particles of radii lower than 1 µm. As a result, we extrapolated the radius distribution function (distribution function) with a fit distribution function(r) = A.r^{2}exp(-B.r^2).

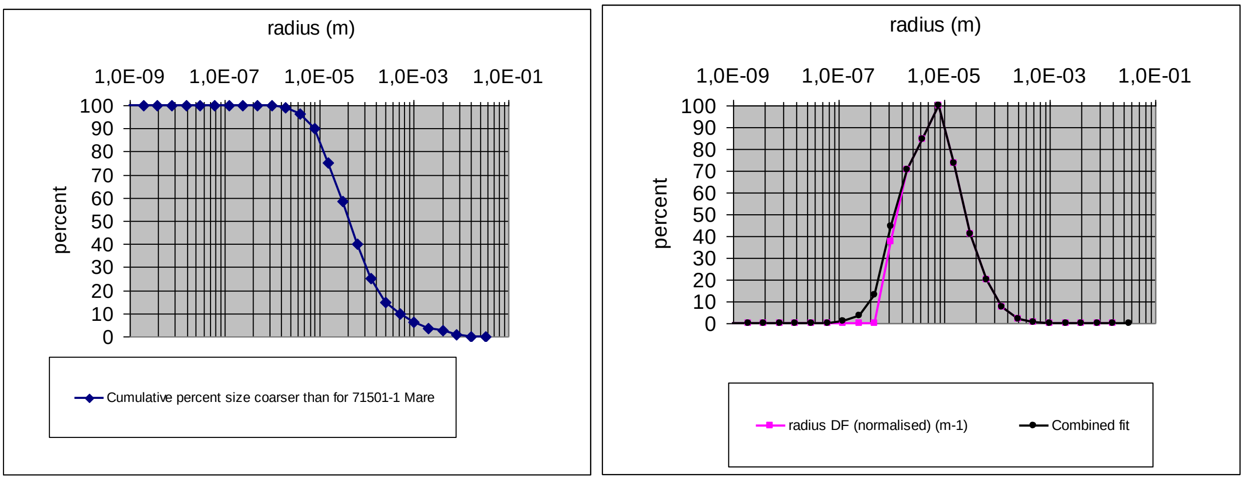
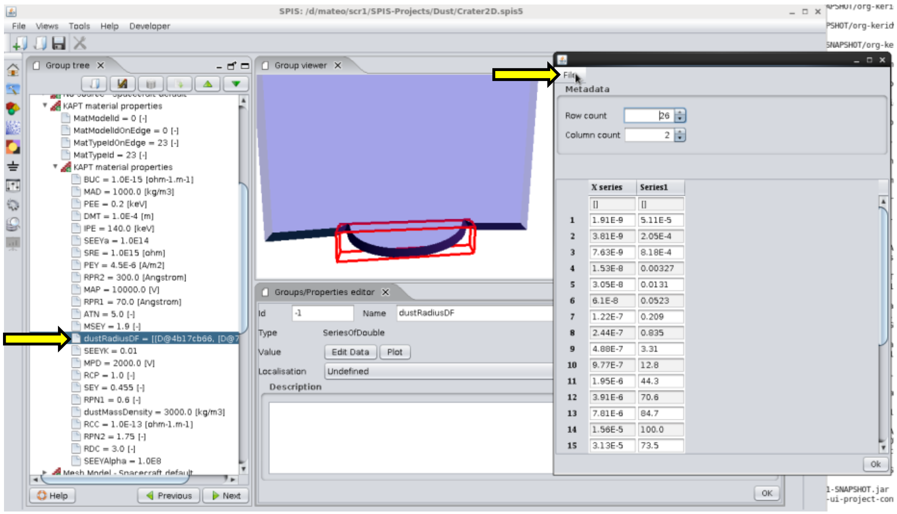


Figure 15 - Dust size distributions. left: cumulative percent of dust coarser than. Right: differential distribution function normalized.

It is possible to edit and plot the distribution function in the group editor as shown in Figure 16.



KAPT material properties
MatModelId = 0 [-]
MatModelIdOnEdge = 0 [-]
MatTypeIdOnEdge = 23 [-]
MatTypeId = 23 [-]
KAPT material properties
BUC = 1.0E-15 [ohm-1.m-1]
MAD = 1000.0 [kg/m3]
PEE = 0.2 [keV]
DMT = 1.0E-4 [m]
IPE = 140.0 [keV]
SEEYa = 1.0E14
SRE = 1.0E15 [ohm]
PEY = 4.5E-6 [A/m2]
RPR2 = 300.0 [Angstrom]
MAP = 10000.0 [V]
RPR1 = 70.0 [Angstrom]
ATN = 5.0 [-]
MSEY = 1.9 [-]
dustRadiusDF = [[D@4b17cb66, [D@7
SEEYK = 0.01
MPD = 2000.0 [V]
RCP = 1.0 [-]
SEY = 0.455 [-]
RPN1 = 0.6 [-]
dustMassDensity = 3000.0 [kg/m3]
RCC = 1.0E-13 [ohm-1.m-1]
RPN2 = 1.75 [-]
RDC = 3.0 [-]
SEEYAlpha = 1.0E8
Mesh Model - Spacecraft default

Id -1 Name dustRadiusDF
Type SeriesOfDouble
Localisation Undefined
Description

Row count 26
Column count 2

	X series	Series1
	[]	[]
1	1.91E-9	5.11E-5
2	3.81E-9	2.05E-4
3	7.63E-9	8.18E-4
4	1.53E-8	0.00327
5	3.05E-8	0.0131
6	6.1E-8	0.0523
7	1.22E-7	0.209
8	2.44E-7	0.835
9	4.88E-7	3.31
10	9.77E-7	12.8
11	1.95E-6	44.3
12	3.91E-6	70.6
13	7.81E-6	84.7
14	1.56E-5	100.0
15	3.13E-5	73.5

Figure 16 - How to edit the dust size distribution function

The numerical model normalizes the volume distribution function. It is scaled by the dust density, with a scaling factor obtained by computing the volume occupied by a dust sample (through Monte Carlo integration techniques).

The weighing algorithm chosen consider non-uniform weight of dust super particles: $w = w_0 r^n$. It permits to sample more dust with a small radius (see section 4.3.3.1.1). The output of this algorithm is the initial condition for dust sampling.

3.5.1.3. Conductivity

The conductivity of lunar dust varies a lot with the samples returned back from Apollo missions. From [LSB] p 531, The bulk conductivity of Apollo 15 soil (15301,38) and Apollo 16 rock (65015,6) depend on the temperature. In addition, [LSB] reports that UV illumination increases the conductivity of soils of a factor 10^6, comparable to that produced by a temperature of 800°C. As a result, we recommend to use resistive dust layers (BUC = 1e-18 $\text{ohm}^{-1}.\text{m}^{-1}$ and SRE = 1e18 ohm) for shaded surface and conductive sunlit dust layers (BUC = -1 and SRE = -1).

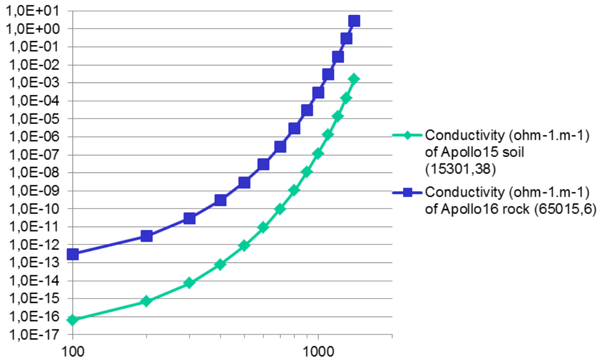
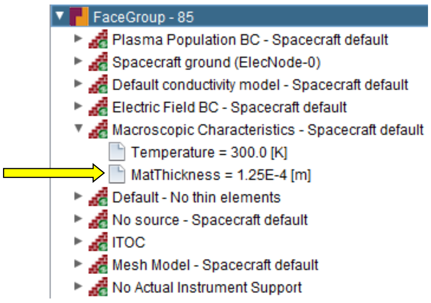


Figure 17 - Bulk conductivity ($\text{ohm}^{-1}.\text{m}^{-1}$) as a function of temperature (K)

3.5.1.4. Thickness

The dust material thickness is generally used to compute the capacitive and conductive coupling with the lunar ground. However, this notion is rather uncertain on the Moon since we do not know exactly the thickness of the layer. Moreover, in the current status of SPIS models, dielectric are considered as thin layers. *NB:* the thickness is defined by the *matThickness* parameter and not by *DMT*



FaceGroup - 85
- Plasma Population BC - Spacecraft default
- Spacecraft ground (ElecNode-0)
- Default conductivity model - Spacecraft default
- Electric Field BC - Spacecraft default
- Macroscopic Characteristics - Spacecraft default
  - Temperature = 300.0 [K]
  - MatThickness = 1.25E-4 [m]
- Default - No thin elements
- No source - Spacecraft default
- ITOC
- Mesh Model - Spacecraft default
- No Actual Instrument Support

Figure 18 - How to define the material thickness (the value of 125 µm is not to relevant to our lunar surface problem)

<u>WARNING</u>: the material thickness is used in the computation of the dielectric surface elements which are treated as planar capacitors (C = $\varepsilon_0\varepsilon_R$ surface / thickness). It is also used for the computation of the electrical conductivity. Thus the user is invited to set the material thickness in order to have a realistic conductivity and to modify the relative dielectric constant ε_R (RDC property in the material properties) to set a numerical capacitance for dielectric materials (for conductive materials an integrated capacitance CSat is defined as a Global Parameter).

It is recommended to pay a special attention to the setting of the numerical capacitance, which may be different from the physical one if only the steady-state solution (or quasi steady-state when the expected evolution of the potential due to environmental variations is much longer than the charging time) is of interest.

The time integration algorithm limits the surface potential variation in a single time step to a fraction of plasma energy. Thus a larger capacitance allows for a longer time step which may be interesting when the dust dynamics is integrated. Larger capacitances smooth the potential variation due to numerical noise and allow damping unphysical instabilities, in particular when the electron flux scaler is used.

The user is invited to perform preliminary short, simplified test runs to determine the optimal numerical capacitance before performing long runs.

3.5.1.5. Secondary emission

The lunar material current density at normal sunlight incidence angle is 4,5 µA/m^2 at 1 AU for lunar dust.

Concerning the secondary emission from particle impact, it is assumed to be low at the surface, but once the dusts are emitted in the volume, the secondary emission model of Chow et al.,1993 is implemented for the emission induced by electron impact.

This model depends on three parameters:

- The inverse of the energy required to excite a single secondary electron given by the material parameter **dustSEEYK**,
- The inverse of the absorption length for secondary electrons given by the material parameter **dustSEEYalpha**,
- TheWhiddington constant for the rate loss with distance given by the material parameter **dustSEEYa**.

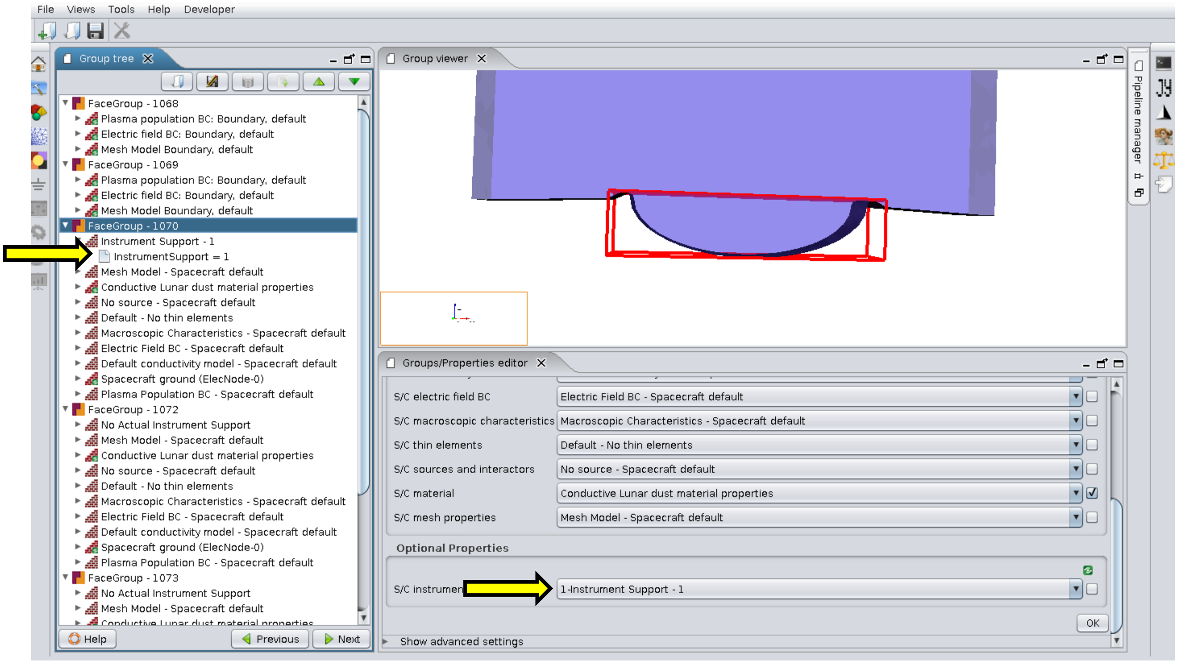
We recommend using the default material parameters [REF!!!]:

- *dustSEEYalpha* = 10^{8}m^{-1}
- *dustSEEYa*= 10^{14}V^{2}m^{-1}
- *dustSEEYK* = 0.01 eV^{-1}

3.5.2. Spacecraft surfaces

3.5.2.1. Instrument surfaces

At **group editor** level, it is sometimes necessary to pre-define the support of dust instruments by selecting the optional properties combo box shown in Figure 19. This is used by some instruments detailed in section 5.



- FaceGroup - 1068
  - Plasma population BC: Boundary, default
  - Electric field BC: Boundary, default
  - Mesh Model Boundary, default
- FaceGroup - 1069
  - Plasma population BC: Boundary, default
  - Electric field BC: Boundary, default
  - Mesh Model Boundary, default
- FaceGroup - 1070
  - Instrument Support - 1
    - InstrumentSupport = 1
  - Mesh Model - Spacecraft default
  - Conductive Lunar dust material properties
  - No source - Spacecraft default
  - Default - No thin elements
  - Macroscopic Characteristics - Spacecraft default
  - Electric Field BC - Spacecraft default
  - Default conductivity model - Spacecraft default
  - Spacecraft ground (ElecNode-0)
  - Plasma Population BC - Spacecraft default
- FaceGroup - 1072
  - No Actual Instrument Support
  - Mesh Model - Spacecraft default
  - Conductive Lunar dust material properties
  - No source - Spacecraft default
  - Default - No thin elements
  - Macroscopic Characteristics - Spacecraft default
  - Electric Field BC - Spacecraft default
  - Default conductivity model - Spacecraft default
  - Spacecraft ground (ElecNode-0)
  - Plasma Population BC - Spacecraft default
- FaceGroup - 1073
  - No Actual Instrument Support
  - Mesh Model - Spacecraft default
  - Conductive Lunar dust material properties

S/C electric field BC	Electric Field BC - Spacecraft default	☐
S/C macroscopic characteristics	Macroscopic Characteristics - Spacecraft default	☐
S/C thin elements	Default - No thin elements	☐
S/C sources and interactors	No source - Spacecraft default	☐
S/C material	Conductive Lunar dust material properties	☑
S/C mesh properties	Mesh Model - Spacecraft default	☐

Optional Properties

S/C instrumen	1-Instrument Support - 1	☐

Figure 19: Instrument setting at group editor level. In this example, the crater is considered as a support of instrumentation of index 1

3.5.2.2. Triboelectric parameters

The triboelectric effect can be taken in consideration for determining the charge of the dust lifting from spacecraft surfaces. It is determined by the surface work function W which is defined by the user at the group editor level.A work function value of -1 deactivates the triboelectric effect for this material (this is typically the case for the dusty surfaces themselves).

[W-2007] recommends using a value of 5.6 eV for oxidized metals.

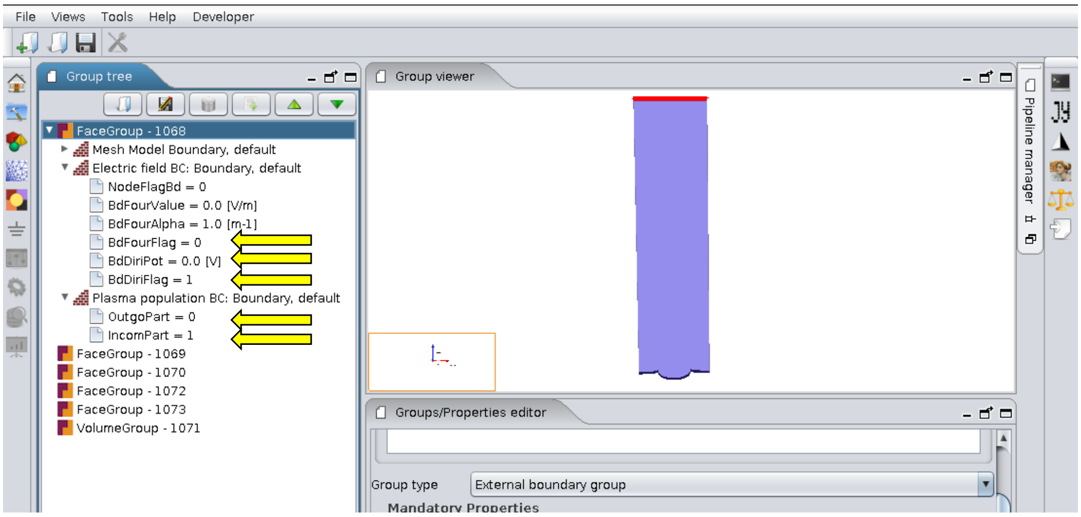
3.5.2.3. Abrasion parameters

In order for the abrasion index computation (see section 4.3.4.5) to be performed, the user must enter the hardness of the material (“HARD”), severity of the abrasion on the component (“SEVER”), and the risk for the mission of the failure of the component due to abrasion (“RISK”). These values are to be defined for each group (if some groups does not have these values defined, the abrasion is not computed for them).

3.6. Definition of the boundary conditions

3.6.1. Open boundary

The open boundary should be set as indicated in Figure 20. It defines how to impose a null potential and classical injection condition for ambient plasma. The choice of a null potential is justified by the large extent of the box with respect to the Debye length. We assume that we are out of the plasma sheath.



FaceGroup - 1068
Mesh Model Boundary, default
Electric field BC: Boundary, default
NodeFlagBd = 0
BdFourValue = 0.0 [V/m]
BdFourAlpha = 1.0 [m-1]
BdFourFlag = 0
BdDiriPot = 0.0 [V]
BdDiriFlag = 1
Plasma population BC: Boundary, default
OutgoPart = 0
IncomPart = 1
FaceGroup - 1069
FaceGroup - 1070
FaceGroup - 1072
FaceGroup - 1073
VolumeGroup - 1071

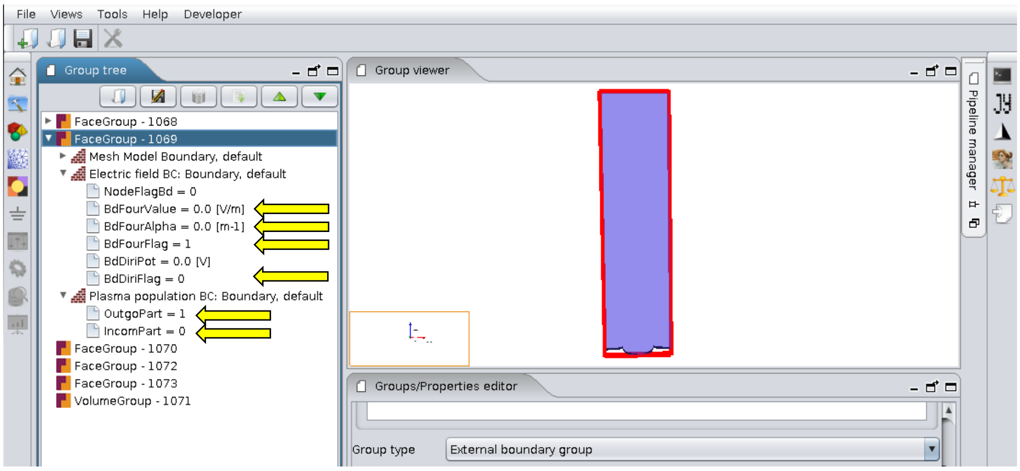
Group type External boundary group

Mandatory Properties

Figure 20 - How to set the top external boundary in the SPIS group editor

3.6.2. Reflective vs periodic conditions

Reflective boundaries can be obtained on the boundary surfaces by setting the particle boundary condition **OutgoPart = 1** and **IncomPart = 0**.

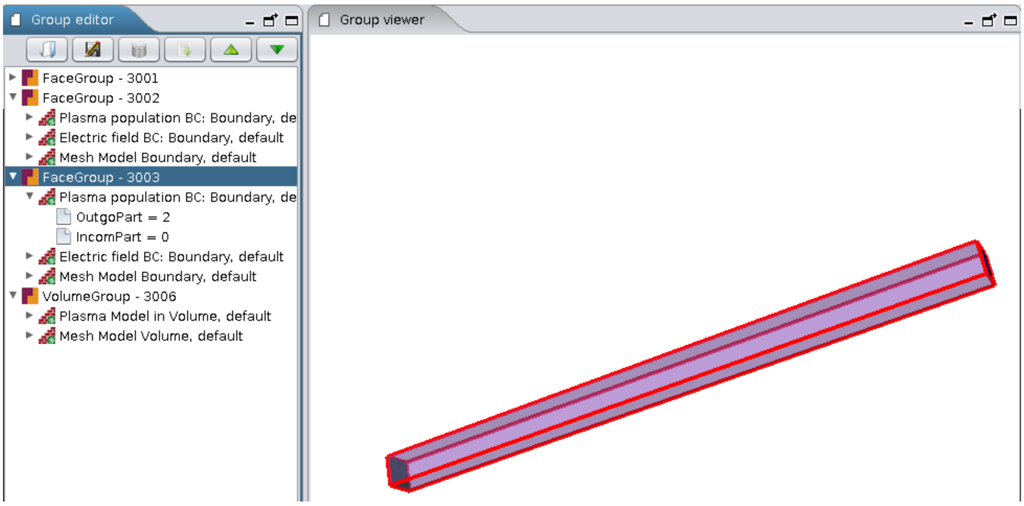


FaceGroup - 1068
FaceGroup - 1069
Mesh Model Boundary, default
Electric field BC: Boundary, default
NodeFlagBd = 0
BdFourValue = 0.0 [V/m]
BdFourAlpha = 0.0 [m-1]
BdFourFlag = 1
BdDiriPot = 0.0 [V]
BdDiriFlag = 0
Plasma population BC: Boundary, default
OutgoPart = 1
IncomPart = 0
FaceGroup - 1070
FaceGroup - 1072
FaceGroup - 1073
VolumeGroup - 1071

Group type External boundary group

Figure 21 - How to set the side external boundaries in the SPIS group editor – reflective conditions

Periodic boundary conditions for the particle fluxes can be set-up in the Group editor UI by setting the particle boundary condition **OutgoPart = 2** and **IncomPart = 0** for the surface groups defined as boundary surfaces.



FaceGroup - 3001
FaceGroup - 3002
  Plasma population BC: Boundary, de
  Electric field BC: Boundary, default
  Mesh Model Boundary, default
FaceGroup - 3003
  Plasma population BC: Boundary, de
    OutgoPart = 2
    IncomPart = 0
  Electric field BC: Boundary, default
  Mesh Model Boundary, default
VolumeGroup - 3006
  Plasma Model in Volume, default
  Mesh Model Volume, default

Figure 22:Set up of the periodic conditions in the UI

For non-rectangular simulation domains the use of the periodicity may be possible but questionable from both mathematical and physical point-of-views, leading to a possible leakage of particles out of the simulation domain and to unphysical results. For some geometries (e.g. half cylinder), the use of periodic condition leads to such a high leakage of particles that the simulation may be unable to start, because the photon population set to compute the shadowing is lost during the set up.

3.7. Definition of the electrical circuit

The electrical circuit between the electrical nodes can be defined as in previous versions of SPIS, by specifying the impedance of a potential bias between nodes in the UI. In particular, this permits one to specify the electrical coupling between the soil and the spacecraft.

4. SETTING UP THE SIMULATION PARAMETERS

4.1. Sun illumination and shadowing control

The illumination by the Sun and the related shadowing is taken into account by SPIS Num to model the solar flux and the photon pressure, see section 4.3.5.1 for more information.

A new feature has been developed in the global parameter to visualize the sun shadowing effect on the spacecraft. The Sun orientation button displays in 3D a view input geometry (i.e. spacecraft and ground) and the Sun, modeled by yellow sphere.

On the bottom of the frame, a new Shadowing button allows user to process Sun shadowing on spacecraft and the surrounding ground.

The shadowing effect is computed using the Penelope’s ray-tracer, considering as source a surface orthogonal to the Sun direction and centered on the yellow sphere. The target is the sub-set of the surface mesh defined by the spacecraft-like groups. Rays impacts are tabulated for each reached surface element of the target and, at the end, normalized by the number of ray emitted by square meter weighted by the area of the surface, as formulated in the following equation:

$$E_{\mathrm{Sun}} = \frac{N_{\mathrm{impacts}}}{N_{\mathrm{emitted\,shoots}}} \cdot \frac{S_{\mathrm{source}}}{S_{\mathrm{target}}}$$

where:

E_{Sun} is the relative exposition to the Sun
N_{impacts} is the number of shoots reaching the target element of surface
$N_{\mathrm{emitted\ shoots}}$ is the number of shoots emitted by the source element of the total surface
S_{source} is the total surface of the emitting surface element
S_{target} is the surface of the target surface element

Consider a target surface parallel to the source surface. If the number of emitted rays shot by the ray tracer is not enough big, the statistics is bad and the normalized value can be higher than one. If the number of emitted rays is enough big, the discretization of the energy emitted by the source will be good and the higher normalized value is one.

Moreover, as illustrated in the Figure 23, the error is distributed symmetrically around the value one for target surfaces parallel to the emitted source surface.

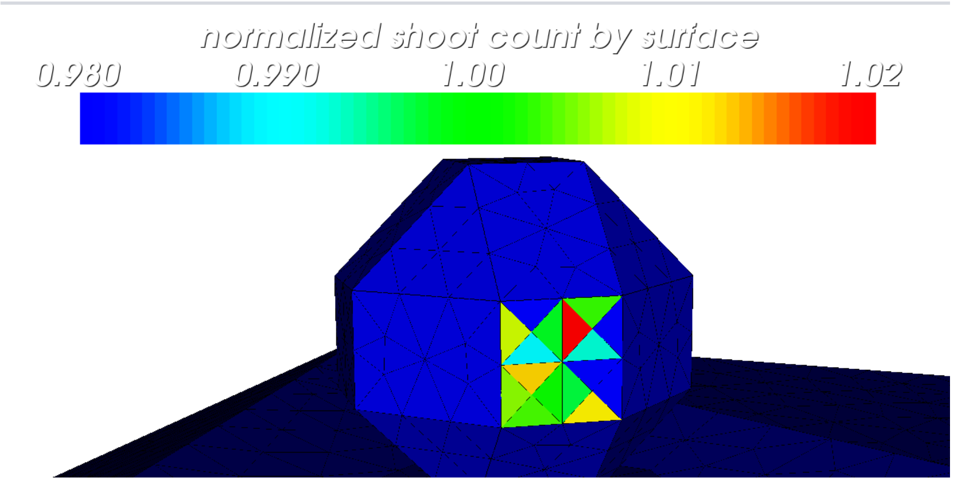


Figure 23: symmetric distribution of the error around the target value one

Some tests have been performed to confirm the precision of the ray-tracing algorithm. First, the error follows the function $\frac{1}{\sqrt{n}}$ like illustrated in the *Figure 24* in accordance with what is expected [Numerical Recipe]. Secondly, as validation, the Figure 25 shows that the number of rays received on a target surface is proportional to the cosine of the angle of the normal of this target surface with the emitting surface normal in accordance with what is expected.

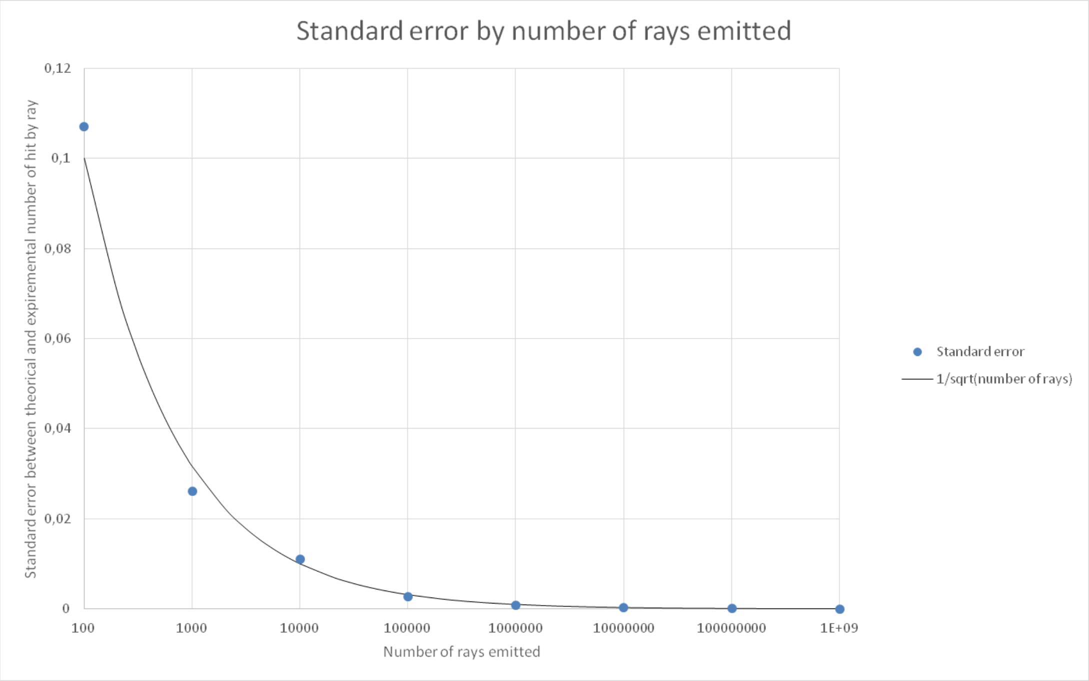


Figure 24: Standard error by number of rays emitted

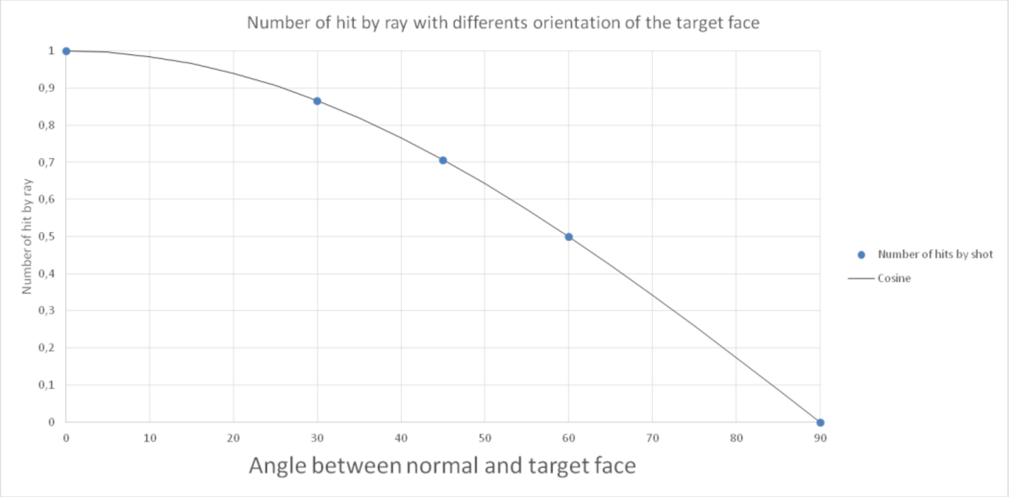


Figure 25: Number of hit by ray with different orientation of the target face

The number of rays used by the ray tracer to model the shadowing effect is displayed on the bottom of the frame. The user can set this number in a dedicated panel when the "*Shadowing*" button is clicked.

To finish, it is possible to launch more rays to improve the ray tracing statistics by clicking on the "*Increase rays*" button.

The Figure 26 presents the 3D result of shadowing. Be careful, this result is used only to control the shadowing but this information is not converted yet to SPIS Num. SPIS Num uses its own internal ray tracer.

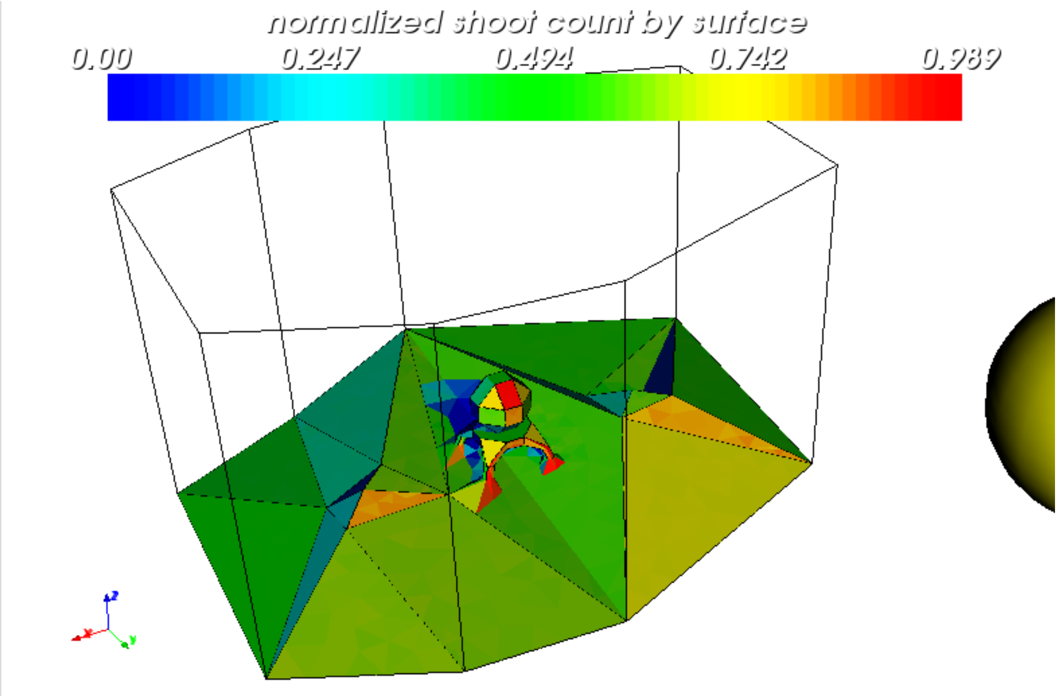


Figure 26: sun shadowing control

4.2. Using predefined parameter sets

4.2.1. Solar wind

A default parameter set as been defined for the simulation of the moon surface in the solar wind (from [MRAD]).These parameters are saved in the SPIS-DUST_SW_parameters.xml file.

Environment parameters:

n_i[cm^{-3}]	10
n_e[cm^{-3}]	10
v_i[$km.s^{-1}$]	430
v_e[$km.s^{-1}$]	430
T_i[eV]	10
T_e[eV]	10
T_{ph}[eV]	2

The time steps for the integration of the plasma and surface evolution have been set up in order to match the stability conditions and to allow the dust motion in a reasonable running time. However, to ensure the simulation stability, the capacitance of the surface must be carefully tuned in accordance with the particular simulation geometry. The global parameters corresponding to the new SPIS-DUST functionalities have been included.

4.2.2. Magnetotail lobes

A default parameter set as been defined for the simulation of the moon surface in the magnetotail lobes (from [MRAD]). These parameters are saved in the SPIS-DUST_MS_parameters.xml file.

Environment parameters:

n_i[cm^{-3}]	0.05
n_e[cm^{-3}]	0.05
T_i[eV]	500
T_e[eV]	50
T_{ph}[eV]	2

The time steps for the integration of the plasma and surface evolution have been set up in order to match the stability conditions and to allow the dust motion in a reasonable running time. To ensure the simulation stability, the capacitance of the surface must be carefully tuned in accordance with the simulation geometry.

4.2.3. Plasma sheet

A default parameter set as been defined for the simulation of the moon surface in the plasma sheet (from [MRAD]). These parameters are saved in the SPIS-DUST_PS_parameters.xml file.

Environment parameters:

n_i[cm^{-3}]	0.5
n_e[cm^{-3}]	0.5
v_i[$km.s^{-1}$]	430
v_e[$km.s^{-1}$]	430
T_i[eV]	5000
T_e[eV]	600
T_{ph}[eV]	2

The time steps for the integration of the plasma and surface evolution have been set up in order to match the stability conditions and to allow the dust motion in a reasonable running time. However, to ensure the simulation stability, the capacitance of the surface must be carefully tuned in accordance with the particular simulation geometry.

4.3. Using new parameters

4.3.1. Solar wind species

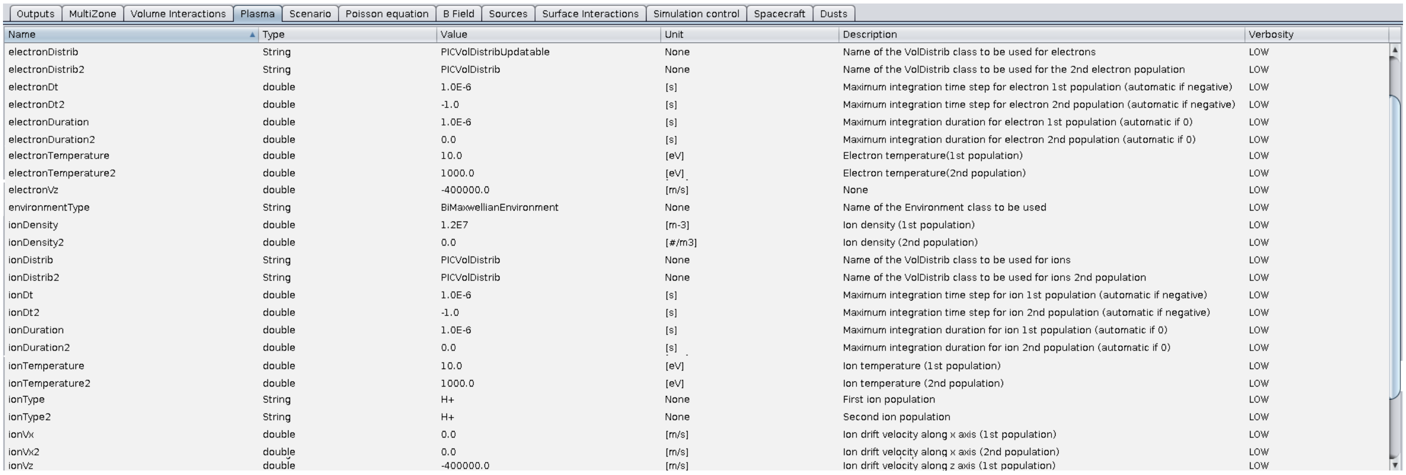
Because of the potential barrier above the surface and because of the expected bulk velocities of the solar wind species, the solar wind species should be defined as PIC distributions.

To do so, the type of environment should be **environmentType** = **BiMaxwellianEnvironment**, and the **electronDistrib, electronDistrib2, ionDistrib and ionDistrib2** global parameters should be **PICVolDistrib**, or**PICVolDistribUpdatable** in order to modify the solar wind species moment during the simulation.

Population densities, temperature and vector velocities can be set by the ***pop*Density, *pop*Temperature, *pop*Vx, *pop*Vy, *pop*Vz** global parameters.

It is recommended to check that the particles densities are in agreement between them (quasi-neutrality where it applies) and with a mesh size sensibly smaller than the Debye length.

The particle time steps should be set such as the plasma oscillation can be properly described (particularly true for the electrons), and such as large charge separations do not appear when the simulation box is filled at the start of the simulation.



Name	Type	Value	Unit	Description	Verbosity
electronDistrib	String	PICVolDistribUpdatable	None	Name of the VolDistrib class to be used for electrons	LOW
electronDistrib2	String	PICVolDistrib	None	Name of the VolDistrib class to be used for the 2nd electron population	LOW
electronDt	double	1.0E-6	[s]	Maximum integration time step for electron 1st population (automatic if negative)	LOW
electronDt2	double	-1.0	[s]	Maximum integration time step for electron 2nd population (automatic if negative)	LOW
electronDuration	double	1.0E-6	[s]	Maximum integration duration for electron 1st population (automatic if 0)	LOW
electronDuration2	double	0.0	[s]	Maximum integration duration for electron 2nd population (automatic if 0)	LOW
electronTemperature	double	10.0	[eV]	Electron temperature(1st population)	LOW
electronTemperature2	double	1000.0	[eV]	Electron temperature(2nd population)	LOW
electronVz	double	-400000.0	[m/s]	None	LOW
environmentType	String	BiMaxwellianEnvironment	None	Name of the Environment class to be used	LOW
ionDensity	double	1.2E7	[m-3]	Ion density (1st population)	LOW
ionDensity2	double	0.0	[#/m3]	Ion density (2nd population)	LOW
ionDistrib	String	PICVolDistrib	None	Name of the VolDistrib class to be used for ions	LOW
ionDistrib2	String	PICVolDistrib	None	Name of the VolDistrib class to be used for ions 2nd population	LOW
ionDt	double	1.0E-6	[s]	Maximum integration time step for ion 1st population (automatic if negative)	LOW
ionDt2	double	-1.0	[s]	Maximum integration time step for ion 2nd population (automatic if negative)	LOW
ionDuration	double	1.0E-6	[s]	Maximum integration duration for ion 1st population (automatic if 0)	LOW
ionDuration2	double	0.0	[s]	Maximum integration duration for ion 2nd population (automatic if 0)	LOW
ionTemperature	double	10.0	[eV]	Ion temperature (1st population)	LOW
ionTemperature2	double	1000.0	[eV]	Ion temperature (2nd population)	LOW
ionType	String	H+	None	First ion population	LOW
ionType2	String	H+	None	Second ion population	LOW
ionVx	double	0.0	[m/s]	Ion drift velocity along x axis (1st population)	LOW
ionVx2	double	0.0	[m/s]	Ion drift velocity along x axis (2nd population)	LOW
ionVz	double	-400000.0	[m/s]	Ion drift velocity along z axis (1st population)	LOW

Figure 27: Set up of the plasma environment in the UI.

4.3.2. Photo-electrons from the ground

4.3.2.1. Maxwellian distribution

The photoelectrons from the ground have by default a Maxwellian distribution. Its temperature can be set by changing the **photoElectronTemperature** global parameter.

The number of photo-electrons can be controlled using the **photoElectronDensification** global parameter.

4.3.2.2. User defined analytical distribution

Non-Maxwellian distributions of the photo-electrons can be used with the automated solar wind electron flux scaler (section 4.3.6). These distributions can be set-up by setting the parameters of a distribution function f(v)dv^3 which takes the form of a modified Pearson IV distribution:

$$f(v) = K\, v^{a_1} \left(1 + \frac{1}{a_3}\left(\frac{mv}{2kT}\right)^{2+a_2}\right)^{-a_3} \exp\left(-a_1 \operatorname{atan}\left(\frac{mv}{2kT}\right)\right)$$

where K is the normalization factor which is numerically computed by SPIS, T is the photo electron temperature when this function defines a Maxwellian distribution (a_1, a_2, a_4 = 0 and $a_3 = \infty$). SPIS replaces this distribution by a usual Maxwell distribution when a_1, a_2, a_4 = 0 and a_3> 20 for best performances and precision.

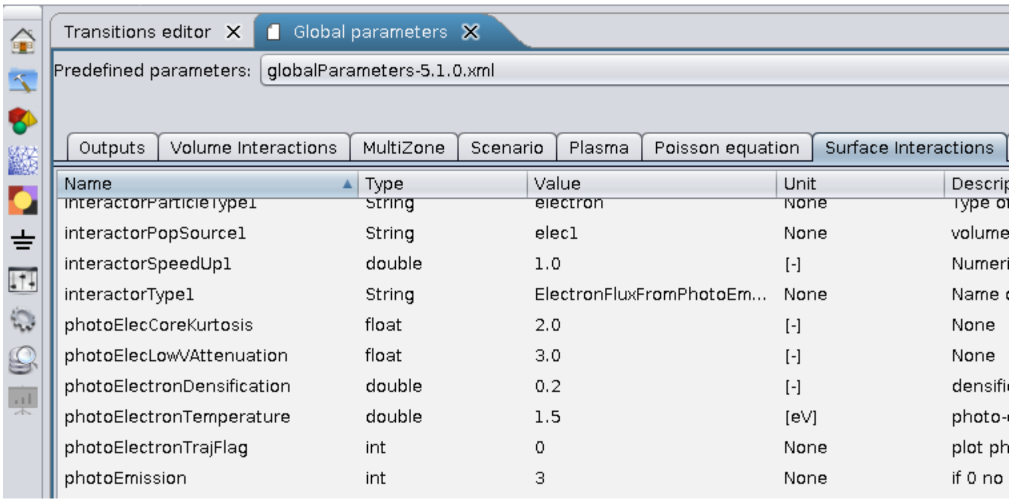
These parameters can be defined as global parameters in the UI:

a_1 is defined by the **photoElecLowVAttenuation** parameter.

a_2 is defined by the **photoElecCoreKurtosis** parameter (the larger it is, the more flat is the core of the distribution, but the tail becomes inexistent).

a_3 is defined by the **photoElecTailKurtosis** parameter (the smaller it is, the more extended is the tail of the distribution)

a_4 is defined by the **photoElecSkewness** parameter, as it derives from the skewness parameter of the Pearson IV distribution.



Predefined parameters: globalParameters-5.1.0.xml

Name	Type	Value	Unit	Descrip
interactorParticleType1	String	electron	None	Type of
interactorPopSource1	String	elec1	None	volume
interactorSpeedUp1	double	1.0	[-]	Numeri
interactorType1	String	ElectronFluxFromPhotoEm...	None	Name c
photoElecCoreKurtosis	float	2.0	[-]	None
photoElecLowVAttenuation	float	3.0	[-]	None
photoElectronDensification	double	0.2	[-]	densifi
photoElectronTemperature	double	1.5	[eV]	photo-
photoElectronTrajFlag	int	0	None	plot ph
photoEmission	int	3	None	if 0 no

Figure 28: Set-up of aFeuerbacher distribution in the UI.

4.3.3. Including dusts

In order for dust to be included in the simulation the **dustEmissionFlag** global parameter must be set to **1** (default value) and a dust source must exist.

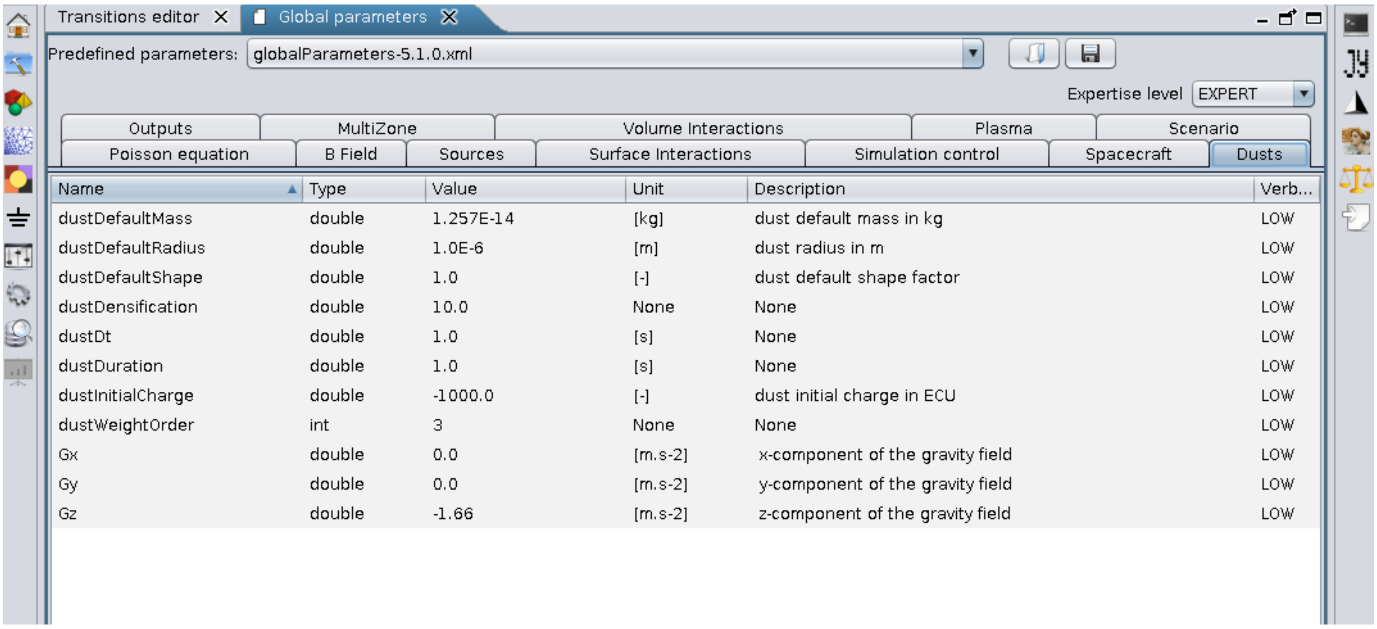
4.3.3.1. Default dust emission

Default dust emission occurs if **dustAutoSource** is set to **1 (default value)** or **2**.

In this case, all the dusty surfaces (surfaces for which the dust properties are set, see section 3.5.1) emit dust according to a force balance between all forces acting of the dusts. The dust emission from the surface is controlled by a surface interactor called *DustySurfaceInteractor*.

4.3.3.1.1. *Control of the non-perturbed distribution on the lunar surface*

The distribution function of the dust on the lunar surface is controlled by the material properties defined on the Local Parameter editor. The first function of the interaction is to convert the distribution function of dust in radius, mass and shape from a continuous function into a particle list representing a sample of particles on the surface. By default, a sample of 100 macroparticles per surface element is defined but this parameter can be controlled by the global parameter called **DustDensification** with is a multiplicative factor of the default number.



Predefined parameters: globalParameters-5.1.0.xml

Expertise level EXPERT

Name	Type	Value	Unit	Description	Verb...
dustDefaultMass	double	1.257E-14	[kg]	dust default mass in kg	LOW
dustDefaultRadius	double	1.0E-6	[m]	dust radius in m	LOW
dustDefaultShape	double	1.0	[-]	dust default shape factor	LOW
dustDensification	double	10.0	None	None	LOW
dustDt	double	1.0	[s]	None	LOW
dustDuration	double	1.0	[s]	None	LOW
dustInitialCharge	double	-1000.0	[-]	dust initial charge in ECU	LOW
dustWeightOrder	int	3	None	None	LOW
Gx	double	0.0	[m.s-2]	x-component of the gravity field	LOW
Gy	double	0.0	[m.s-2]	y-component of the gravity field	LOW
Gz	double	-1.66	[m.s-2]	z-component of the gravity field	LOW

Figure 29:View of the dusts global parameters in the UI.

The sample of particle can be randomly sampled in anisoprobability way (by default). This means that all the macroparticles have the same statistical weight. Thus the most probable macroparticles corresponds to the most numerous dusts in the distribution function. In most of the case, this is not optimal as the numerous dusts that are able to fly are not the most numerous ones on the surface. Thus numerically, it is much more optimal to sample more small dust macroparticles (with radius less than 1µm) than the heaviest ones because the lasts ones are stick on the lunar surface due to the gravity. We introduce a global parameter **dustWeightingOrder** which permits a weighting of the macroparticle sampling probability. If this parameter is set to 0, the sampling is isoprobable. If this parameter is greater than 1, the smaller particles are more represented (i.e. more macroparticles but with a smaller statistical weight to preserve the distribution function). It is recommended to set this parameter to 2or 3 to have a good representation of the dusts between 100nm and 10nm.

4.3.3.1.2. *Dusts density on the surface*

The possibility is let to the user to let this parameter to the user in term of surface density (i.e. number of dust per square centimetres) in the **dustsDensity** global parameter. Otherwise, the code computes a density considering that the distribution function on the surface is the same as in volume (provided by data from the literature) and that the surface density corresponds to:

$$N_s = \frac{\sum_N w}{\sum_N \pi r^2 w}$$

It permits to have an effective charged surface area greater than the real surface area.

4.3.3.1.3. The charge of the dusts

The global charge of a surface element is computed using the Gauss Law with the knowledge of the electric field on the surface. It permits to have a total charge in coherence with the photoelectron sheath or whatever the sheath creation mechanism. In the law, the area used is the area of the surface element and E the electric field on each surface element.

$Q_S = \varepsilon_0 E_S S$

Between two iteration steps (time step), the total charge variation is computed and the charge difference is let between all the dusts on the surface (proportional to the cross section surface but insuring the total charge is conserved). This total charge is divided between the macroparticles with respect to the dust radius,

$$\Delta Q_D \sim r_D^a \times \Delta Q_S$$

ensuring the conservation of the total charge by:

$$\sum_N w\, \Delta Q_D = \Delta Q_S$$

A linear dependence of the charge of the dust with the radius is chosen (a=1) as the potential of the dust on the surface is considered to be the potential of the surface and we assume a spherical-like capacitance for the dusts: $Q_D = CV \sim r_D V$

The charge is not assumed to be equally divided between the physical dust grains composing a macroparticle. The model implemented in SPIS is inspired of the tip model. As the ground is very irregular, there will be tips. The geometry of the tip is summarized by the enhancement factor β: a tip of factor β is β times taller than large and is at a distance β times its width from another tip. In this case, the charges concentrate at the top of the tip, leading to an electric field enhanced by a factor β. If a dust is on the top of the tip, the charge will be partially transferred to the dust which will gain a charge enhanced by a factor and which will also feel the enhanced electric field.

In the model implemented in SPIS, all β are equiprobable, but because of its geometrical definition, tips with a factor β cover a fraction β^{-2} of the surface. In term of macroparticles, it means that on the ground the macroparticle charge and weight are modified so that:

$$Q'_D \rightarrow \beta Q_D \; and \; w' \rightarrow \frac{w}{\beta^2}$$

The dust is assumed to be accelerated by the forces applied to it (see next section) with its enhanced charge until it reaches an altitude corresponding to its size. Then, the charge is recomputed assuming a spherical capacitive coupling of the dust with the plasma (Q=CV). This charge can eventually be modified by the triboelectric effect (see section 4.3.4.4). The emission velocity of the dust is that it gained while accelerated with its enhanced charge. To be effectively emitted, the dust must be able to reach an altitude of 1cm with its velocity at emission and the recomputed charge it has after the ejection.

When a dust is emitted, it is assumed that the freshly uncovered dust has the same properties than the emitted dust (radius,…) and that it charges by conduction between dusts in a time RC=ε_0/σ_D, where σ_D is the conductivity in the dust layer (set to 1E-9 Ω-1.m-1). The charging time is taken into account in SPIS by including its effect in the weight of the emitted dusts:

$$w_{emit} = \frac{w\sigma_D \delta t}{\beta^2 \varepsilon_0}$$

Although it is recommended to keep the default values, the above model can be modified by changing the value of thea-index (global parameter "*dustChargingLawPower*", e.g. a=2 correspond to a charge division proportional to the dust cross-section) and the dust layer conductivity (Global parameter "*dustLayerConductivity*").

4.3.3.1.4. Forces on the dust surface

The forces on the dusts on the surface include:

- the electrostatic force,
- the gravity
- the cohesive force
- the seismic forces

The electrostatic force is computed from the projection of the electric field on the barycenter of the surface elements, the charge of the dusts and the field amplification factor due to local amplification structures.

$$F_E = Q'_D \beta E.n$$

The projected electric field is thus uniform on the surface elements. It is used to compute the electric field but also the total charge of the dusts layer (using the Gauss law).

Other forces are discussed in section 4.3.5.
When the sum of force is considered to be greater than 0, the dust is ejected from the surface.

The number of injected macroparticle of dusts in the simulation is automatically controlled by the interactor following the densification factor set in the input parameters.

4.3.3.2. User defined dust emission

The user defined flux can be set by setting the **dustAutoSource** global parameter to **0**.
Then the user can define the emitted dust velocity distribution, the dust charge distribution and the radius distribution.

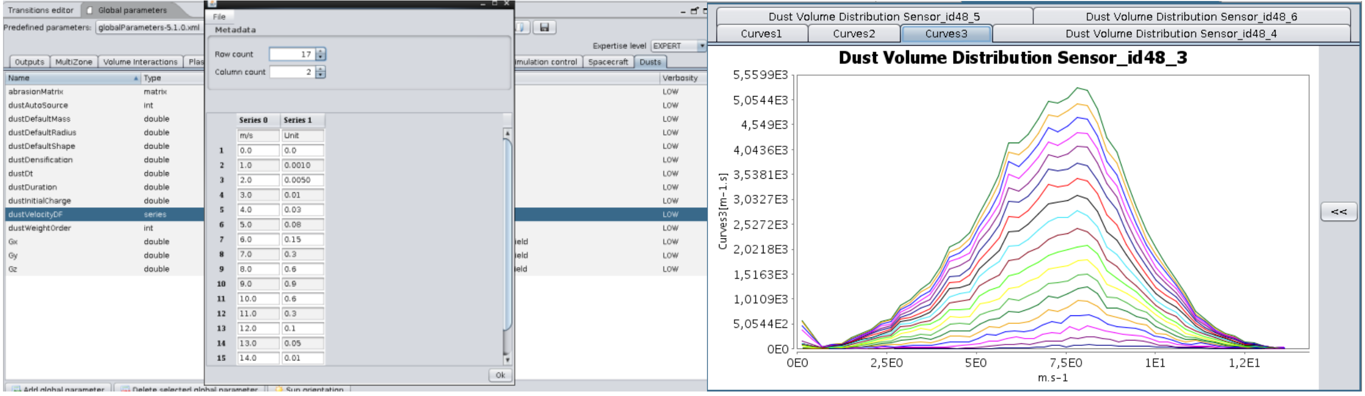
All of these distributions can either be Dirac distributions or tabulated distributions. Alternatively, the radius distribution is by default that of the surface (defined in the group editor) if any.

A tabulated velocity distribution can be set using the global parameter **dustVelocityDF**, if this parameter is not set, a Dirac distribution can be set using the global parameter **dustInitialVelocity**.

A tabulated charge distribution can be set using the global parameter **dustChargeDF**, if this parameter is not set, a Dirac distribution can be set using the global parameter **dustInitialCharge**.

Where no radius distribution exists for the surface, a tabulated radius distribution can be set using the global parameter **dustRadiusDF**, if this parameter is not set, a Dirac distribution can be set using the global parameter **dustDefaultRadius**.

Distributions are defined as "series" global parameters, with the first column being the velocity and the second the distribution amplitude.



Predefined parameters: globalParameters-5.1.0.xml

Expertise level EXPERT

Name	Type	Verbosity
abrasionMatrix	matrix	LOW
dustAutoSource	int	LOW
dustDefaultMass	double	LOW
dustDefaultRadius	double	LOW
dustDefaultShape	double	LOW
dustDensification	double	LOW
dustDt	double	LOW
dustDuration	double	LOW
dustInitialCharge	double	LOW
dustVelocityDF	series	LOW
dustWeightOrder	int	LOW
Gx	double	LOW
Gy	double	LOW
Gz	double	LOW

Metadata

Row count 17

Column count 2

	Series 0	Series 1
	m/s	Unit
1	0.0	0.0
2	1.0	0.0010
3	2.0	0.0050
4	3.0	0.01
5	4.0	0.03
6	5.0	0.08
7	6.0	0.15
8	7.0	0.3
9	8.0	0.6
10	9.0	0.9
11	10.0	0.6
12	11.0	0.3
13	12.0	0.1
14	13.0	0.05
15	14.0	0.01

Figure 30: Example of user defined dust velocity distribution, set up in the UI [left] and observed in the simulation domain [right]. The observation is not exactly the injected distribution, as the detected dust already underwent deceleration due to gravity.

The emission is turned on for all surfaces by default, but can be restricted to a given surface by using **dustSourceId** (in which case the corresponding **sourceId** must be set in the group editor for the surfaces of interest).

The default and user defined interactions can be used together (e.g. to simulate a default surface + a particular ejection spot) by setting the **dustAutoSource** global parameter to 2.

4.3.4. Letting dust interact

4.3.4.1. Electron collection

The dust charging in volume is controlled by a volume interactor in SPIS. Three versions of the interactor exist following if the user wants to model the plasma collection in a fluid way (considering OML law) or a Monte Carlo way (computation of a collection probability considering an interaction cross section). The 3 versions are:

- DustChargingInteractor
- DustChargingInteractorMCC
- DustChargingInteractorMCC1

(A forth version, named DustBacktrackChargingMCC1 is reserved for dust backtracking)

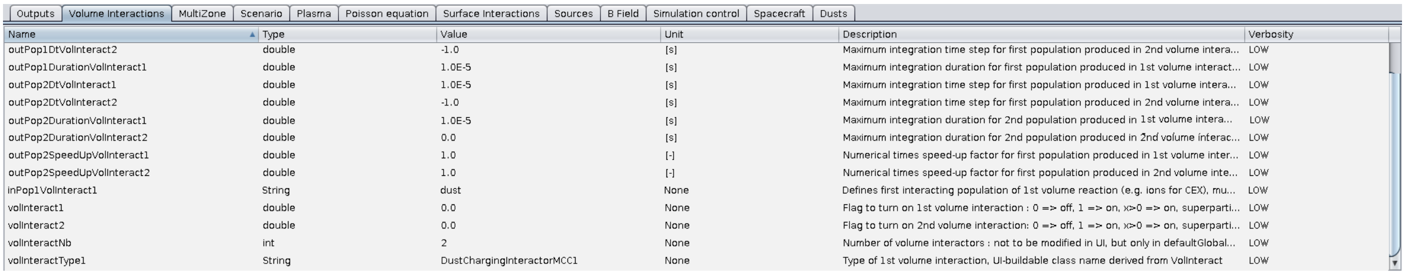
All the interactions are based on the calculation of the potential of the dusts on the plasma with the relation (considering a capacitive coupling):

$$\phi_d(t) = \frac{Q_d(t)}{C} = \frac{Q_d(t)}{4\pi\varepsilon_0 r_d\left(1 + r_d / \lambda_D\right)}.$$

From this potential, the *DustChargingInteractor* computes the current collected with the OML law whatever the distribution function. The *DustChargingInteractorMCC* and *DustChargingInteractorMCC1* compute the current collection from a *PICVolDistrib* in a Monte Carlo way and in an OML way for the other distributions. The total charge of the dust is updated at each time step following the current balance:

$$\frac{dQ_D}{dt} = J$$

To set up the dust charging interactor, one of the above interactors must be associated to a volume interactor in the global parameters. Assuming for the example that we have only one volume interactor, the global parameters must be:
volInteractNb = 1
volInteract1 = 1
volInteractType1 =DustChargingInteractor, DustChargingInteractorMCC or DustChargingInteractorMCC1
inPop1VolInteract1 = dust



Name	Type	Value	Unit	Description	Verbosity
outPop1DtVolInteract2	double	-1.0	[s]	Maximum integration time step for first population produced in 2nd volume intera...	LOW
outPop1DurationVolInteract1	double	1.0E-5	[s]	Maximum integration duration for first population produced in 1st volume interact...	LOW
outPop2DtVolInteract1	double	1.0E-5	[s]	Maximum integration time step for first population produced in 1st volume intera...	LOW
outPop2DtVolInteract2	double	-1.0	[s]	Maximum integration time step for first population produced in 2nd volume intera...	LOW
outPop2DurationVolInteract1	double	1.0E-5	[s]	Maximum integration duration for 2nd population produced in 1st volume intera...	LOW
outPop2DurationVolInteract2	double	0.0	[s]	Maximum integration duration for 2nd population produced in 2nd volume interac...	LOW
outPop2SpeedUpVolInteract1	double	1.0	[-]	Numerical times speed-up factor for first population produced in 1st volume inter...	LOW
outPop2SpeedUpVolInteract2	double	1.0	[-]	Numerical times speed-up factor for first population produced in 2nd volume inte...	LOW
inPop1VolInteract1	String	dust	None	Defines first interacting population of 1st volume reaction (e.g. ions for CEX), mu...	LOW
volInteract1	double	0.0	None	Flag to turn on 1st volume interaction : 0 => off, 1 => on, x>0 => on, superparti...	LOW
volInteract2	double	0.0	None	Flag to turn on 2nd volume interaction: 0 => off, 1 => on, x>0 => on, superparti...	LOW
volInteractNb	int	2	None	Number of volume interactors : not to be modified in UI, but only in defaultGlobal...	LOW
volInteractType1	String	DustChargingInteractorMCC1	None	Type of 1st volume interaction, UI-buildable class name derived from VolInteract	LOW

Figure 31Example of set up of the dust volume interactor

4.3.4.2. Photo-emission

The current density at normal sunlight incidence angle is 4,5 µA/m^2 at 1 AU for lunar dust. On the surface, the flux is moderated by the sun incidence angle. For dust in the gaseous phase, the photocurrent is maximal when irradiated and null when at the shade of a crater for instance. It is the multiplication of the photoelectron current density by the dust cross section.

On the lunar surface, the energy distribution function of photo electron is Maxwellian by default, with a temperature of 2 eV. It is also possible to use the Feuerbacher-1972 distribution function given by using global parameter **electronSecondaryTemperature** that gives the mean velocity of the relation. :

$$f_{ph}(|v|) = \frac{n_{ph}}{\pi\Gamma\left(\frac{3}{4}\right) v_{ph}^{3}} v^{2} exp\left[-\left(\frac{v}{v_{ph}}\right)^{4}\right]$$

The relation is normalized following the photoemission current provided in the standard material properties. For dust in the gaseous phase this latter distribution function is always used.

For both secondary electron emissions under electron and photon impacts, the net current emitted depends on the dust potential and on the plasma potential.

For a positive φ_d (the potential difference between the dust and the plasma) and a Maxwellian distribution function: $I_{ph} = J_{ph}\pi r^2 \exp\left(-\frac{e\varphi}{k_B T_{ph}}\right)$. This formula is used although it is only approximate for the Feuerbacher distribution function. For a negative dust, $I_{ph} = J_{ph}\pi r^2$.

The real energy of the emitted secondaries is adjusted with respect to the potential difference φ_d in order to conserve the total electron energy.

4.3.4.3. Secondary emission

The model of Chow-1993 presented in [MRAD] is implemented. It takes into account the primary electron energy and the dust radius as inputs.

The model is described by the following equations:

$$l(x,\ \varphi) = \left\{ \left(\frac{D}{2}\right)^2 + \left(\frac{D}{2} - x\right)^2 - 2\left(\frac{D}{2}\right)\left(\frac{D}{2} - x\right) \cdot \cos\left\{ \varphi - \sin^{-1}\left[\left(\frac{2}{D}\right)\left(\frac{D}{2} - x\right) \sin(\varphi)\right]\right\}\right\}^{1/2},$$

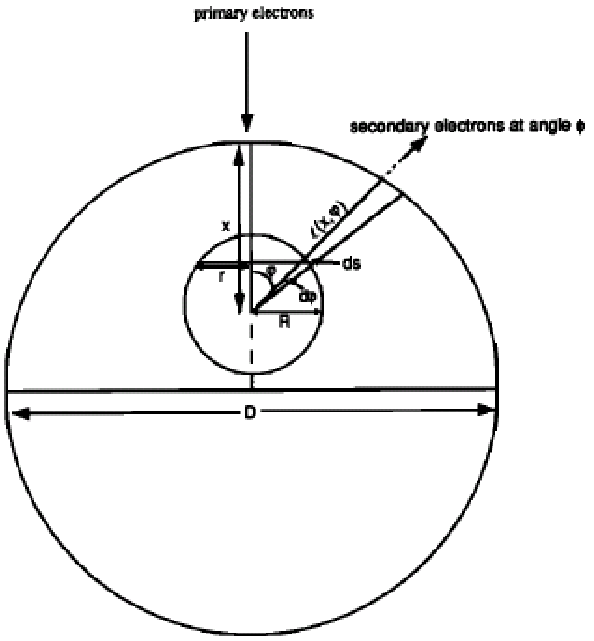
$$di_s = \tfrac{1}{2} K a i_p [(eV_p)^2 - ax]^{-1/2} f(x)\ dx$$

$$f(x) = \int_0^{\pi} \tfrac{1}{2} \sin(\varphi) e^{-\alpha l(x,\ \varphi)}\ d\varphi,$$

$$eV(x) = [(eV_p)^2 - ax]^{1/2},$$

$$x_{\max} = \frac{e^2 V_p^2}{a},$$

$$\delta = \frac{1}{2} \int_0^{\min[D, x_{\max}]} Ka[(eV_p)^2 - ax]^{-1/2} f(x)\ dx.$$



<u>The computation</u> is performed at the beginning of the simulation and can take a few minutes. It consists in generating the 2D tabulation in energy and dust diameter. For each interaction on the dust, the SEEE is interpolated from the tabulated data. The results of the tabulation agree with the literature presented in Figure 32.

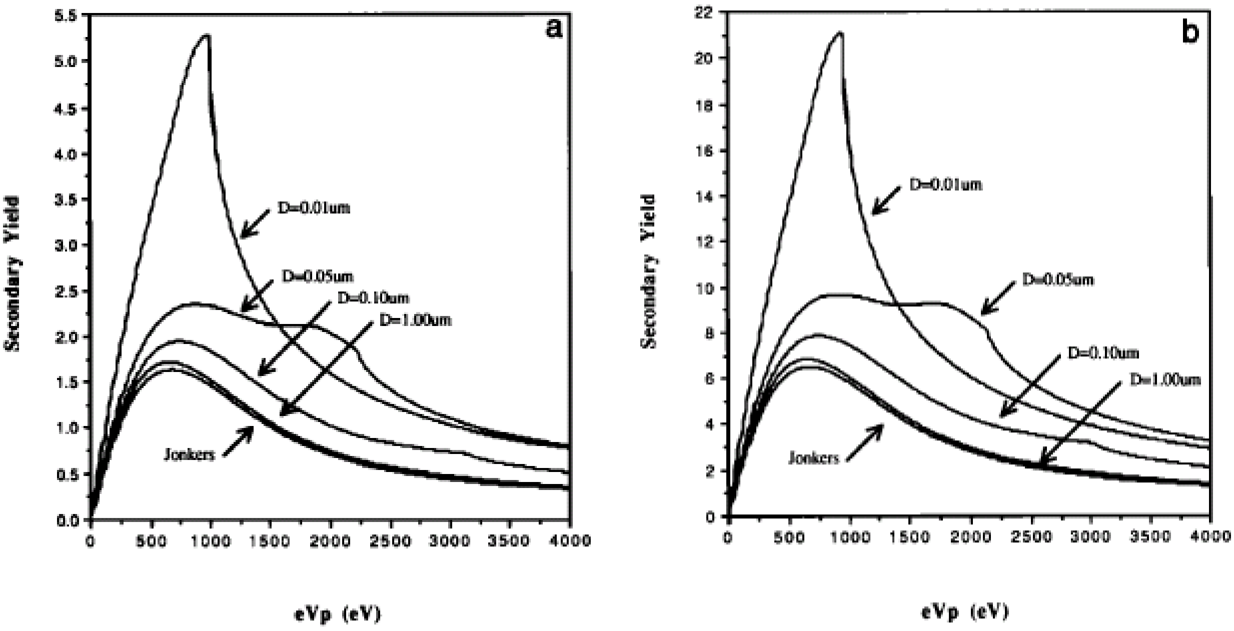


Figure 32 - SEEE yield for insulating (a) and conducting (b) dusts

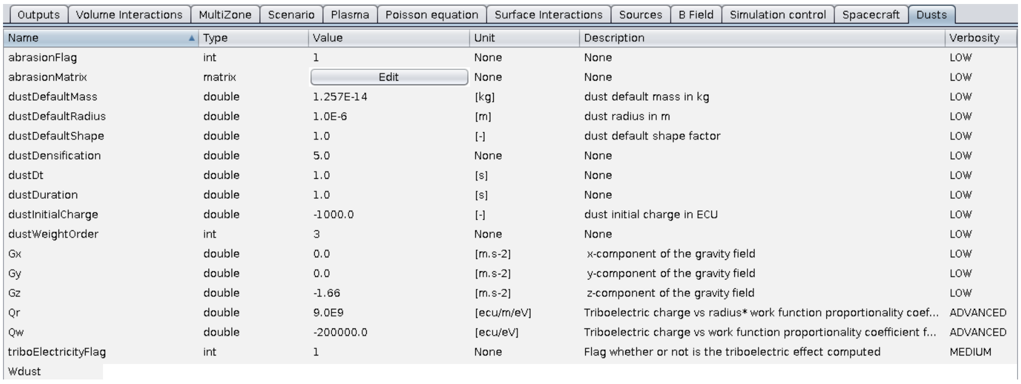
The real energy of the emitted secondaries is adjusted with respect to the potential difference φ_d in order to conserve the total electron energy.

4.3.4.4. Tribo-electricity

The triboelectric effect can be taken in consideration for determining the charge of the dust lifting from spacecraft surfaces. It is determined by four global parameters: **Wdust, Qw, Qr** which determine the charge variation of the dust lifting from a surface:
$\Delta Q=(Qr.r_d+Qw)(W-Wdust)$, with W the surface work function (section 3.5.2.2), and r_d the dust radius.

Such a formula is derived from the work of [S-2002] which showed that for a given dust radius the charge variation is approximately linear with the work function, and that for a given work function the charge variation is approximately linear with the dust radius. The values that match best the [S-2002] results (also shown in [W-2007]) are:

Wdust	=5.8	eV,
Qr	=9E9	ecu/m/eV,
Qw	=-2E5	ecu/eV

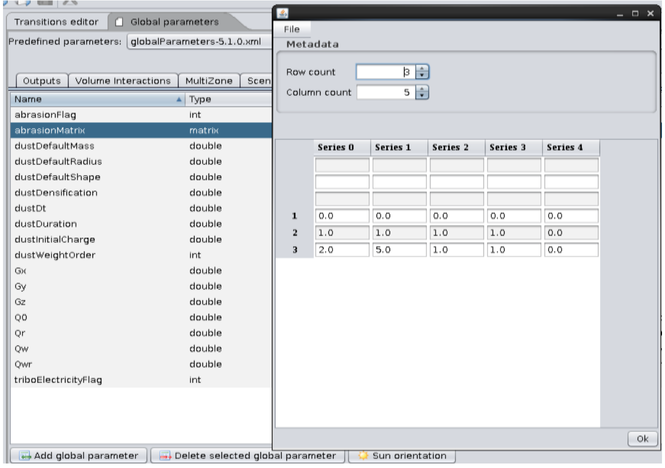


Name	Type	Value	Unit	Description	Verbosity
abrasionFlag	int	1	None	None	LOW
abrasionMatrix	matrix	Edit	None	None	LOW
dustDefaultMass	double	1.257E-14	[kg]	dust default mass in kg	LOW
dustDefaultRadius	double	1.0E-6	[m]	dust radius in m	LOW
dustDefaultShape	double	1.0	[-]	dust default shape factor	LOW
dustDensification	double	5.0	None	None	LOW
dustDt	double	1.0	[s]	None	LOW
dustDuration	double	1.0	[s]	None	LOW
dustInitialCharge	double	-1000.0	[-]	dust initial charge in ECU	LOW
dustWeightOrder	int	3	None	None	LOW
Gx	double	0.0	[m.s-2]	x-component of the gravity field	LOW
Gy	double	0.0	[m.s-2]	y-component of the gravity field	LOW
Gz	double	-1.66	[m.s-2]	z-component of the gravity field	LOW
Qr	double	9.0E9	[ecu/m/eV]	Triboelectric charge vs radius* work function proportionality coef...	ADVANCED
Qw	double	-200000.0	[ecu/eV]	Triboelectric charge vs work function proportionality coefficient f...	ADVANCED
triboElectricityFlag	int	1	None	Flag whether or not is the triboelectric effect computed	MEDIUM
Wdust					

Figure 33: dust global parameters

4.3.4.5. Risk for the mission

SPIS-DUST can determine a risk for the mission at the surface element level. This risk can be visualized on a 2D surface or monitored during the simulation.



Predefined parameters: globalParameters-5.1.0.xml

Name	Type
abrasionFlag	int
abrasionMatrix	matrix
dustDefaultMass	double
dustDefaultRadius	double
dustDefaultShape	double
dustDensification	double
dustDt	double
dustDuration	double
dustInitialCharge	double
dustWeightOrder	int
Gx	double
Gy	double
Gz	double
Q0	double
Qr	double
Qw	double
Qwr	double
triboElectricityFlag	int

Row count 3

Column count 5

	Series 0	Series 1	Series 2	Series 3	Series 4
1	0.0	0.0	0.0	0.0	0.0
2	1.0	1.0	1.0	1.0	0.0
3	2.0	5.0	1.0	1.0	0.0

Figure 34: Set up of the abrasion matrix in the UI

At the global parameters level, the user must define an abrasion matrix of the form: [An | Hn | Sn | Rn | Fn]. The abrasion index for a surface element will be equal to An if the hardness is lower than Hn, the severity larger than Sn, the Risk Larger than Rn and the dust collision frequency larger than Fn.

The first line of the abrasion matrix must be filled with zeros.

4.3.5. Forces acting on the dusts

4.3.5.1. Solar Flux and photon pressure

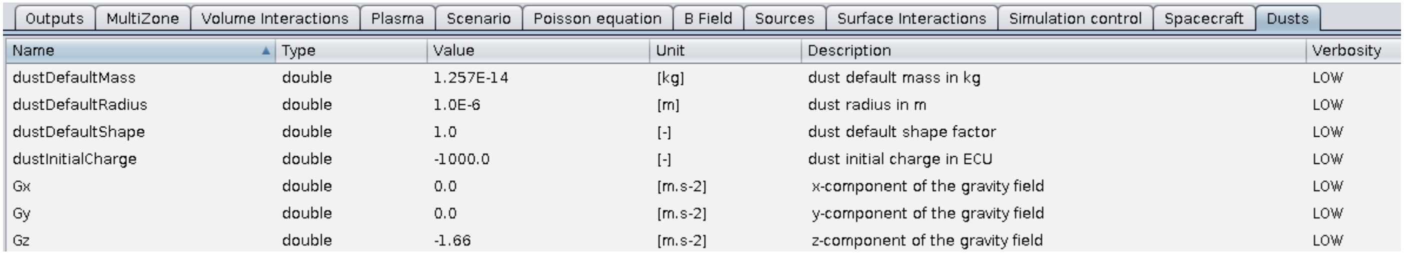
The solar flux does not only act on the photo-emission rate, but also on the photon pressure applied on the dusts. This pressure is computed from the solar flux at the dust particle positions (including the shadowing) and the dust cross-section. For the photon pressure to be taken into account, one of the DustChargingInteractor must be set up.

4.3.5.2. Gravitation

The gravity force is directly the product of the gravity vector projection on surface element and the mass of the dust.

$$F_G = m_D g.n$$

The gravity force is zero where the surfaces are orthogonal to the gravity vector. This can causes some problems in comparison to the reality due to the fact on quite vertical surface, the roughness makes that the gravity is never completely orthogonal to the local surface.



Name ▲	Type	Value	Unit	Description	Verbosity
dustDefaultMass	double	1.257E-14	[kg]	dust default mass in kg	LOW
dustDefaultRadius	double	1.0E-6	[m]	dust radius in m	LOW
dustDefaultShape	double	1.0	[-]	dust default shape factor	LOW
dustInitialCharge	double	-1000.0	[-]	dust initial charge in ECU	LOW
Gx	double	0.0	[m.s-2]	x-component of the gravity field	LOW
Gy	double	0.0	[m.s-2]	y-component of the gravity field	LOW
Gz	double	-1.66	[m.s-2]	z-component of the gravity field	LOW

Figure 35: Set up of the gravitational field in the UI

4.3.5.3. Seismic forces

Seismic forces can be included that will act on the dust on surface. These forces are determined by an amplitude, a frequency and a phase in three directions, one being the averaged normal to the lunar surface. The global parameters controlling these forces are:

SeismicNormalAmplitude, **SeismicNormalFrequency** and **SeismicNormalPhase**, amplitude (in m, m.s-1 or m.s-2), frequency and phase (in radian) of the seismic perturbation normal to the surface.

SeismicTransverseAmplitude1,**SeismicTransverseFrequency1** and **SeismicTransverseFrequency1**,amplitude (in m, m.s-1 or m.s-2), frequency and phase (in radian) of the seismic perturbation parallel to the surface (toward +y if possible).

SeismicTransverseAmplitude2,**SeismicTransverseFrequency2** and **SeismicTransverseFrequency2**,amplitude (in m, m.s-1 or m.s-2), frequency and phase (in radian) of the seismic perturbation parallel to the surface.

4.3.5.4. Cohesive forces

The Van der Waals force is modelled as $F_C = C.S^2.r$ From the literature, the factor C is about 5e-2 kg/s^2 and S is about 0.8 (Perko-2001, Dove-2012). From Hartzell-2013, it seems impossible to launch particles submitted to such a force (1 µm size dusts undergo a 1e-8 N force). As a result, they suggested that the dust cleanliness and the non uniform charge could respectively lead to a strong decrease of the cohesion force, and to a strong increase of the electrostatic force. In the present model, we consider that CS^2 is equal to 1e-6 kg/s^2.TheCS^2value can be set using the **dustCohesionConstant** global parameter (in unit of kg/s^2).

4.3.5.5. Magnetic Field

Particle pushers including the magnetic field have been updated and tested for the dust transport (gravity and photon pressure have been added).The inclusion of a dipolar field is possible from the UI, using the following global parameters:

MagDipoleCenterx, MagDipoleCentery, MagDipoleCenterz, the position of the dipole center.

MagMomentx, MagMomenty, MagMomentz, which gives the dipole moment in T/m^3.

4.3.6. *Modelling the sheath*

The simulation domain is typically designed as the celestial body surface facing one open boundary and with closed (either reflecting or periodic) walls on the side. This set-up generate problems of quasi neutrality at the open boundary since the solar wind species are in different regimes (supersonic ions, subsonic electrons) and since there are other sources of electrons (mainly photoelectrons from the ground). The problem is to ensure plasma neutrality at z = 0, since there is an excess of electrons due to photoemission and SEEE. If nothing is done, we obtain the maps of Figure 36.

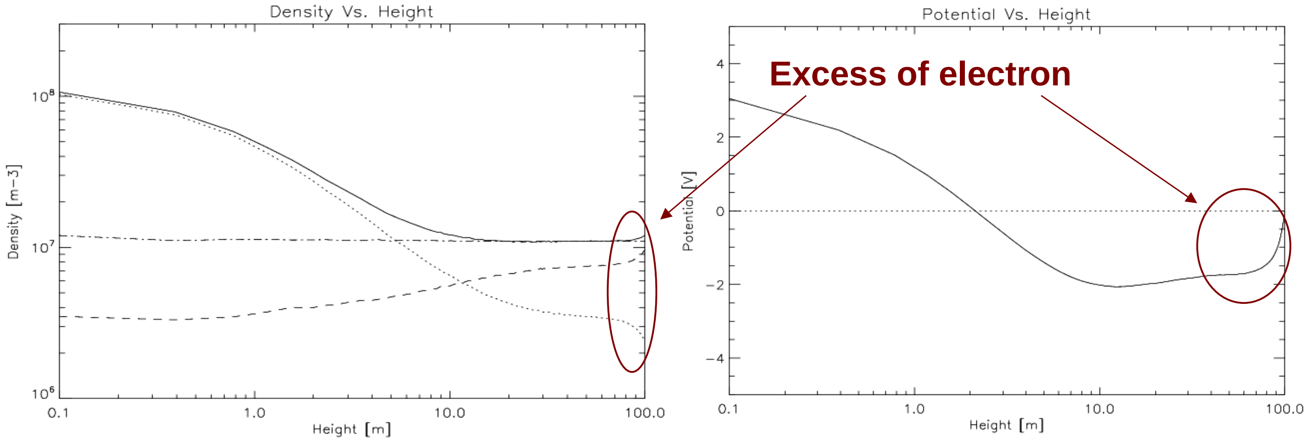


Figure 36 - Effect of the electron excess

The solution proposed is to adapt the condition of ambient electron condition at the top boundary. To do so, the population densities at the open boundary must be evaluated. We estimate the densities from the potential profile across the simulation domain.

The population density at the open boundary is given by the following expressions, where *erf* is the error function, n_{e0} and n_{i0} the plasma density in undisturbed conditions; v_0 the drift velocity, v_{Te} the electron thermal velocity.

Model of the electron density to inject for Maxwellian distributions:

For a planar isopotential surface

Ion density: $n_i = n_{io}$

Electron density: $$n_e = \frac{n_{eo}}{2}\left(1 + erf\left(\frac{v_o}{v_{Te}}\right) + erf\left(\frac{v_o}{v_{Te}}\right) - erf\left(\frac{v_o}{v_{Te}} - \sqrt{\frac{e\Delta V_2}{kT_e}}\right)\right) \quad \text{with} \quad v_{Te} = \sqrt{\frac{2KT_e}{m_e}}$$

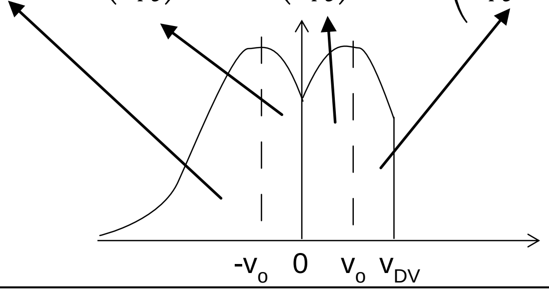


Figure: electron distribution at the open boundary

Photo- *electron current at and after the potential minimum*

Photo electron density:

$$n_{\phi} = \frac{J_{\phi}\exp\left(\frac{-e(\Delta V_1 + \Delta V_2)}{kT_{\phi}}\right)}{e\sqrt{\frac{\frac{2}{\pi}kT_{\phi} + e\Delta V_2}{m_e/2}}}$$

charge × *mean velocity at the open boundary*

Amplitude of the electron distribution function to inject:

$$n_{eo} = 2\left(n_{io} - \frac{J_{\phi}\exp\left(\frac{-e(\Delta V_1 + \Delta V_2)}{kT_{\phi}}\right)}{e\sqrt{\frac{\frac{2}{\pi}kT_{\phi} + e\Delta V_2}{m_e/2}}}\right).\left(1 + 2\mathrm{erf}\left(\frac{v_o}{v_{Te}}\right) - erf\left(\frac{v_o}{v_{Te}} - \sqrt{\frac{e\Delta V_2}{kT_e}}\right)\right)^{-1}$$

For non-planar and dielectric surfaces (non-isopotentials):

$$n_{eo} = 2\left(n_{io} - \frac{\Phi_{\phi}}{e\sqrt{\frac{\frac{2}{\pi}kT_{\phi} + e\Delta V_2}{m_e/2}}}\right).\left(1 + 2\mathrm{erf}\left(\frac{v_o}{v_{Te}}\right) - erf\left(\frac{v_o}{v_{Te}} - \sqrt{\frac{e\Delta V_2}{kT_e}}\right)\right)^{-1}$$

the photo electron flux at the electric potential minimum is :

$$\Phi_{\phi} = \int J_{\phi}(\vec{n}.\vec{\sigma})(\vec{x})\left(1 - \frac{1}{\pi}acos(\vec{n}(\vec{x}).\langle\vec{n}\rangle)\right)\left(1 - \frac{1}{\pi}atan\left(\frac{max(0, -\mathrm{e}(\langle\Delta V_1\rangle - \Delta V_1(\vec{x})))}{kT_{\phi}}\right)\right)\exp\left(\frac{max(0, -\mathrm{e}(\Delta V_1(\vec{x}) - \Delta V_2))}{kT_{\phi}}\right)d\vec{x}$$

n is the normal to the simulation domain cross-section, **σ** the normal to each element of the surface, <ΔV_1> the averaged potential at the surface. There are three terms in addition to the exp(-eV/kT) term representing the decrease of the density due to the potential difference between the surface and the potential minimum (this time computed using the local surface potential). The potential differences in the formula are limited in such a way that the flux considered is never larger than the one effectively emitted by the surface (saturation):

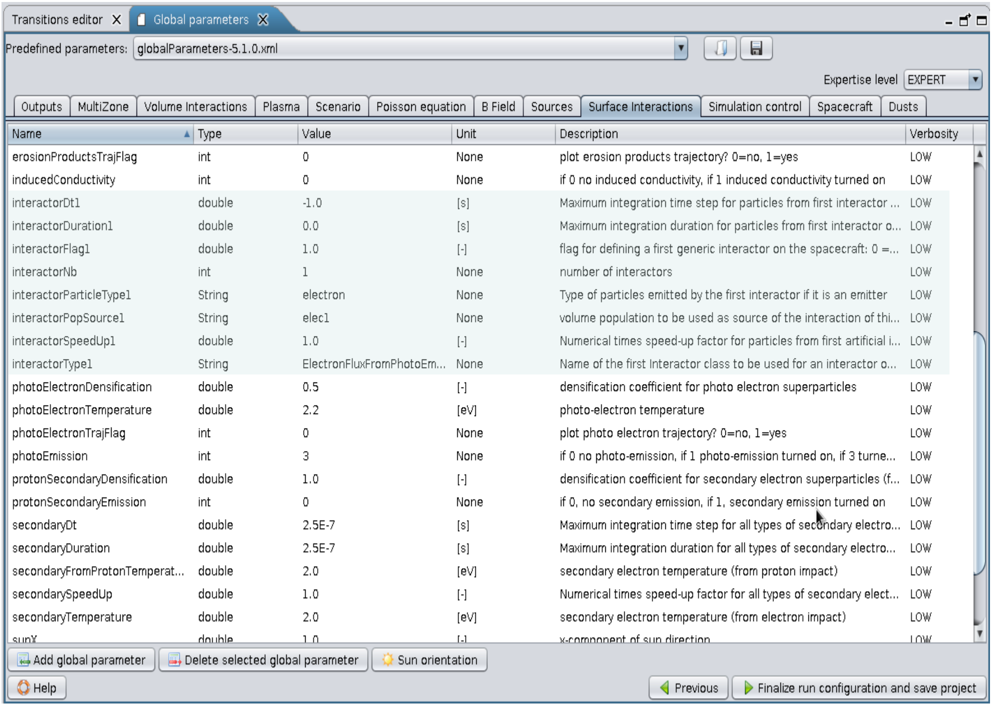
_ First one (scalar product) represents the decrease of the photo electron current in the **n** direction due to the surface element inclination.

_ Second one (with an arccosine) removes the part of the photoelectron distribution that corresponds to negative velocities in regards of the n direction (photoelectron can be emitted toward negatives altitudes in the surface is inclined). Alternatively, this term can be replaced by $\vec{n}(\vec{x}).\langle\vec{n}\rangle$which accounts for the projection of the normal current. This alternative term may lead to a better estimate, but leads to divergence of the simulation in some cases.

_ Third one (with an arctangent) represents the effect of the non-isopotential surface. It stands for the misalignment of the direction of emission of the photo-electrons in regards to the normal to the surface elements due to perpendicular electric fields generated by the potential differences on the surface. It can be obtained by assuming that the plasma potential parallel to the surface goes from the potential surface to the average potential over λ_D, so that E=(<ΔV_1>-ΔV_1)/ λ_D, and that this electric field decreases linearly to 0 over an altitude range of λ_D (above this the potential ~ homogeneous), so over a time λ_D/<v_z> (with <v_z> the mean velocity perpendicular to the surface). Hence, the particle gain a velocity parallel to the surface δv=-e(<ΔV_1>-ΔV_1)/m<v_z>, and the deviation angle is atan(δv/<v_z>) =atan(-e(<ΔV_1>-ΔV_1)/kT). The same result can also easily be obtained based on energetic assumption (kinetic vs potential).

The above formula has been validated over several test cases combining planar or non-planar surfaces, conductive and/or dielectric surfaces and different solar inclinations. For each of these cases, it has been verified that above the sheath, the potential is nearly constant and about 0V, and the densities of each species are constant and close to that predicted. The estimate of the electron density to inject is strongly dependent on the estimate of the photo-electron flux at the potential minimum Φ_ϕ. Any systematic error on the estimate larger than ~1% leads at best to values of the potential above the sheath off by several tenths of Volts and at worst to a divergence of the simulation. Because of this sensitivity, it is necessary to smooth the variation of the injected density to avoid positive feedbacks with the variation of the surface potential. The smoothing factor is set by default to 0.1, i.e. the variation of the density to inject over a time step is divided by 10.

The automated fit of the density to inject can be obtained by using a surface interactor called *“ElectronFluxFromPhotoEmission”*. It can be set in the “Surface Interactions” tab in the UI as shown on Figure 18 in the case where the interactor is Interactor1 (it could be another number, depending whether other interactors are used, and that the *interactorNb* parameter is set accordingly).
_ **interactorFlag#** must be set to **1.0**
_ **interactorType#** must be set to **ElectronFluxFromEmission**
_ **interactorParticleType#** must be set to **electrons**
_ **interactorPopSource#** is the name of the electron population to inject (e.g. **elec1**).
_**convergenceFactorInteractor#** can be used to define how fast the density to inject can vary. Its default value is -1, meaning that the factor auto-adapts to limit the relative variation at each step to 2E-3.
To use the alternative estimate of the photoelectron current, use *“ElectronFluxFromPhotoEmission2”*.



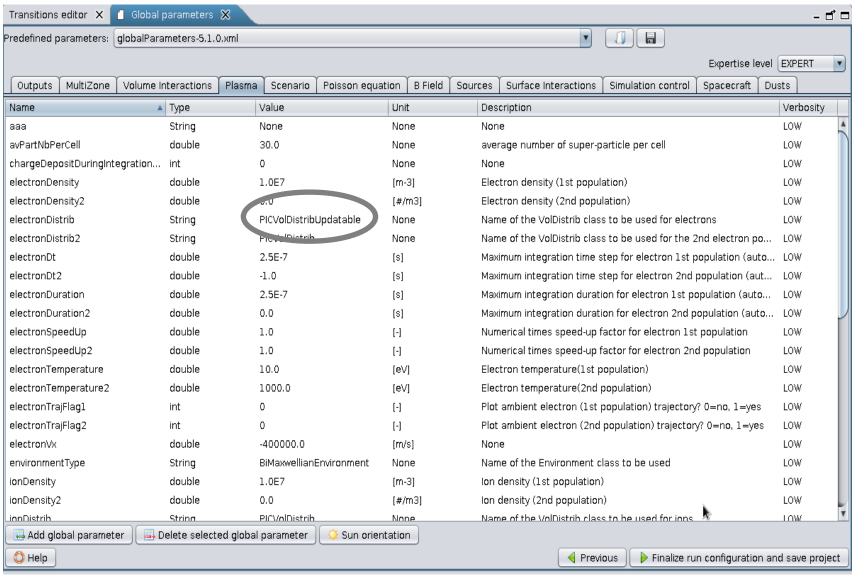
Predefined parameters: globalParameters-5.1.0.xml

Expertise level EXPERT

Name	Type	Value	Unit	Description	Verbosity
erosionProductsTrajFlag	int	0	None	plot erosion products trajectory? 0=no, 1=yes	LOW
inducedConductivity	int	0	None	if 0 no induced conductivity, if 1 induced conductivity turned on	LOW
interactorDt1	double	-1.0	[s]	Maximum integration time step for particles from first interactor ...	LOW
interactorDuration1	double	0.0	[s]	Maximum integration duration for particles from first interactor o...	LOW
interactorFlag1	double	1.0	[-]	flag for defining a first generic interactor on the spacecraft: 0 =...	LOW
interactorNb	int	1	None	number of interactors	LOW
interactorParticleType1	String	electron	None	Type of particles emitted by the first interactor if it is an emitter	LOW
interactorPopSource1	String	elec1	None	volume population to be used as source of the interaction of thi...	LOW
interactorSpeedUp1	double	1.0	[-]	Numerical times speed-up factor for particles from first artificial i...	LOW
interactorType1	String	ElectronFluxFromPhotoEm...	None	Name of the first Interactor class to be used for an interactor o...	LOW
photoElectronDensification	double	0.5	[-]	densification coefficient for photo electron superparticles	LOW
photoElectronTemperature	double	2.2	[eV]	photo-electron temperature	LOW
photoElectronTrajFlag	int	0	None	plot photo electron trajectory? 0=no, 1=yes	LOW
photoEmission	int	3	None	if 0 no photo-emission, if 1 photo-emission turned on, if 3 turne...	LOW
protonSecondaryDensification	double	1.0	[-]	densification coefficient for secondary electron superparticles (f...	LOW
protonSecondaryEmission	int	0	None	if 0, no secondary emission, if 1, secondary emission turned on	LOW
secondaryDt	double	2.5E-7	[s]	Maximum integration time step for all types of secondary electro...	LOW
secondaryDuration	double	2.5E-7	[s]	Maximum integration duration for all types of secondary electro...	LOW
secondaryFromProtonTemperat...	double	2.0	[eV]	secondary electron temperature (from proton impact)	LOW
secondarySpeedUp	double	1.0	[-]	Numerical times speed-up factor for all types of secondary elect...	LOW
secondaryTemperature	double	2.0	[eV]	secondary electron temperature (from electron impact)	LOW
sunX	double	1.0	[-]	x-component of sun direction	LOW

Figure 37: set up of the interactor in the UI

The population of electron to inject must be associated to a “PICVolDistribUpdatable” distribution by changing the **electronDistrib** parameter as shown on Fig. 20.



Predefined parameters: globalParameters-5.1.0.xml

Expertise level EXPERT

Name	Type	Value	Unit	Description	Verbosity
aaa	String	None	None	None	LOW
avPartNbPerCell	double	30.0	None	average number of super-particle per cell	LOW
chargeDepositDuringIntegration...	int	0	None	None	LOW
electronDensity	double	1.0E7	[m-3]	Electron density (1st population)	LOW
electronDensity2	double	0.0	[#/m3]	Electron density (2nd population)	LOW
electronDistrib	String	PICVolDistribUpdatable	None	Name of the VolDistrib class to be used for electrons	LOW
electronDistrib2	String	PICVolDistrib	None	Name of the VolDistrib class to be used for the 2nd electron po...	LOW
electronDt	double	2.5E-7	[s]	Maximum integration time step for electron 1st population (auto...	LOW
electronDt2	double	-1.0	[s]	Maximum integration time step for electron 2nd population (aut...	LOW
electronDuration	double	2.5E-7	[s]	Maximum integration duration for electron 1st population (auto...	LOW
electronDuration2	double	0.0	[s]	Maximum integration duration for electron 2nd population (auto...	LOW
electronSpeedUp	double	1.0	[-]	Numerical times speed-up factor for electron 1st population	LOW
electronSpeedUp2	double	1.0	[-]	Numerical times speed-up factor for electron 2nd population	LOW
electronTemperature	double	10.0	[eV]	Electron temperature(1st population)	LOW
electronTemperature2	double	1000.0	[eV]	Electron temperature(2nd population)	LOW
electronTrajFlag1	int	0	[-]	Plot ambient electron (1st population) trajectory? 0=no, 1=yes	LOW
electronTrajFlag2	int	0	[-]	Plot ambient electron (2nd population) trajectory? 0=no, 1=yes	LOW
electronVx	double	-400000.0	[m/s]	None	LOW
environmentType	String	BiMaxwellianEnvironment	None	Name of the Environment class to be used	LOW
ionDensity	double	1.0E7	[m-3]	Ion density (1st population)	LOW
ionDensity2	double	0.0	[#/m3]	Ion density (2nd population)	LOW
ionDistrib	String	PICVolDistrib	None	Name of the VolDistrib class to be used for ions	LOW

Figure 38: set up of the particle distribution for the automated fit of the electron density to inject

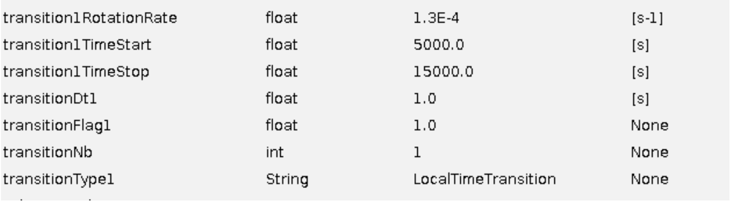
4.3.7. Time-dependent simulation

4.3.7.1. Changing the sun direction

Although SPIS already included several transitions allowing to update some parameters, a new **LocalTimeTransition** was introduced which permits to rotate the sun light and background plasma species direction of arrival (computing the shadowing), and to modify each species moment as a function of time and/or local time (e.g. to use local time variation obtained from a large scale simulation). Each of the modified values is given by:

value(t) = nominal*Value*×(1+*value*IncreaseRate×t) × (1+zenith*Value*Enhancement×cos(θ(t))), with θ the zenith angle and t the time.

To do so, **LocalTimeTransition** must be set in **transitionType***N*, (with N the number of the transition). The transition operates between two times defined by the global parameters **transition***N***TimeStart** and **transition***N***TimeStop**, with a rotation rate given by the **transition***N***RotationRate** Global parameter. All other parameters are optional.



transition1RotationRate	float	1.3E-4	[s-1]
transition1TimeStart	float	5000.0	[s]
transition1TimeStop	float	15000.0	[s]
transitionDt1	float	1.0	[s]
transitionFlag1	float	1.0	None
transitionNb	int	1	None
transitionType1	String	LocalTimeTransition	None

Figure 39: Set up of the local time transition in the UI

The direction of the zenith angle and the rotation axis can be set using the global parameters: **zenithDirectionX, zenithDirectionY, zenithDirectionZ** *(by default, normal to the surface)*

rotationAxisX, rotationAxisY, rotationAxisZ *(by default, perpendicular to the zenith, toward +y)*

The photon flux variation can be set using the global parameters:
zenithPhotonFluxEnhancement: enhancement of the photon flux when θ=0°.
photonFluxIncreaseRate: increase of the photon flux versus time

4.3.7.2. Changing the solar wind direction

The particle population whose moment can be changed must be **PICVolDistribUpdatable** populations.

If the environment is set as **BiMaxwellianEnvironment**, the LocalTimeTransition module **automatically selectsthe two electron and two ion populations** and change their velocity direction and moments if the distributions are PICVolDistribUpdatable.

Otherwise, the number of population concerned must be set in **transition*N*PopNumber**, and the name of the population in **transition*N*Pop***M*, where *M* is the population number and N the number of the transition.

4.3.7.3. Changing the solar wind moments versus time

The moments (density, velocity and temperature) of the different species can evolve linearly with time by setting the **densityIncreaseRate, velocityIncreaseRate, temperatureIncreaseRate** global parameters respectively. These parameters are in second-1 as they are multiplicative factors to the initial moments.

The moment variation for particular species can be set by using the global parameters, **pop#DensityIncreaseRate, pop#VelocityIncreaseRate, pop#TemperatureIncreaseRate**, which override the aforementioned global parameters (# being the population number).

4.3.7.4. Changing the solar wind moments versus local time

The moments (density, velocity and temperature) of the different species can evolve with the local time by setting the **zenithDensityEnhancement, zenithVelocityEnhancement, zenithTemperatureEnhancement** global parameters respectively. These parameters are adimensioned as they are multiplicative factors to the initial moments.

The moment variation for particular species can be set by using the global parameters **pop#ZenithDensityEnhancement, pop#ZenithVelocityEnhancement, pop#ZenithTemperatureEnhancement,** which override the aforementioned global parameters (# being the population number).Such parameters may be used to adapt the plasma parameters obtained from a global scale simulation to the SPIS simulation.

5. SETTING UP THE INSTRUMENTATION

5.1. New instrument features

In the simulation step, some new features have been developed about managing instruments.

After select an existing instrument on the instrument list defined in the *Instrument list viewer* panel, it is possible to duplicate the selected instrument by clicking on the new “*duplicate the selected instrument*” button. The duplicated instrument is renamed with a “duplicated” extension name and with exactly the same parameters as the initial instrument. The user can edit it like usual.

The new “*Import instruments from external SPIS project*” button allows user to import some instruments defined in an external SPIS project. The imported instruments are renamed with an “imported” extension name and with exactly the same parameters.

To finish, when users edit an instruments, it is now possible to edit the name of this instrument in a dedicated text field.

5.2. Default monitoring of the dust-surface interaction

5.2.1. 1D live monitors

Several 1D monitors dedicated to dust interaction with surfaces have been developed. These monitors are automatically built if dusts exist in the simulation.

There are four dust-surface interaction monitors:

- Dust current on the surface per electrical nodes

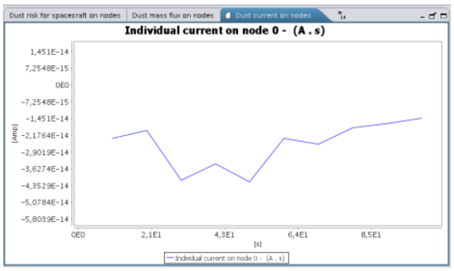


Figure 40: Live monitoring of the dust current

- Dust obscuration of the surface per electrical nodes
- Dust mass flux on the surface per electrical nodes
- Maximum risk for the mission by electrical nodes

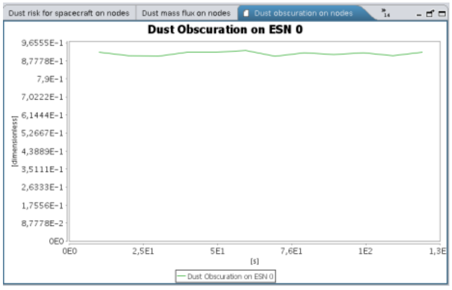


Figure 41: Live monitoring of the obscuration factor

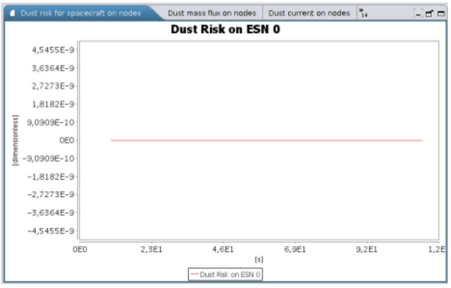


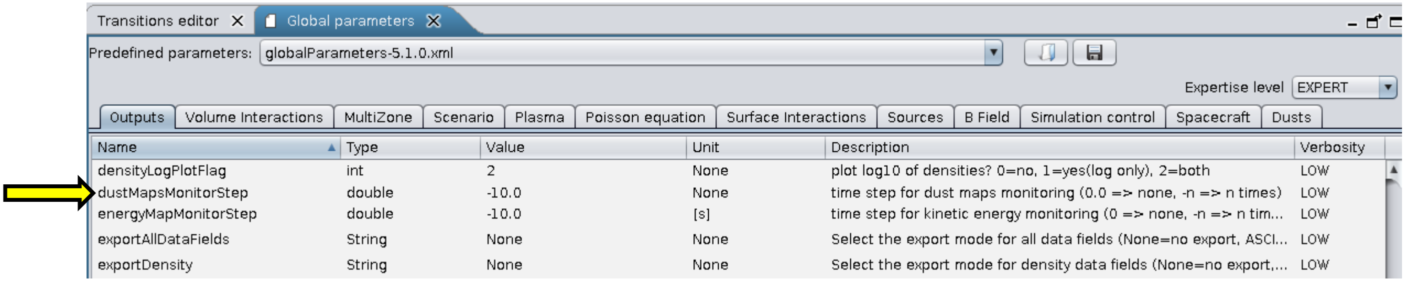
Figure 42: Live monitoring of the abrasion risk

5.2.2. 2D surface maps

SPIS generates several 2D maps of dust moments and of dust dependent quantities: averaged radius, mass charge, layer width on surfaces, abrasion index, obscuration factor…

To activate all dust maps the parameter **dustMapsMonitorStep** shall be defined at Global Editor level with the default value of -10 (indicates that the dust maps are produced every 1/10 of the simulation duration). In the event it does not already exist, please use the "add" button and modify the *newParameter* so-generated.

Some parameters, such as the abrasion related quantities can only be displayed if the associated quantities (global and local parameters) are defined.



Predefined parameters: globalParameters-5.1.0.xml

Expertise level EXPERT

Name	Type	Value	Unit	Description	Verbosity
densityLogPlotFlag	int	2	None	plot log10 of densities? 0=no, 1=yes(log only), 2=both	LOW
dustMapsMonitorStep	double	-10.0	None	time step for dust maps monitoring (0.0 => none, -n => n times)	LOW
energyMapMonitorStep	double	-10.0	[s]	time step for kinetic energy monitoring (0 => none, -n => n tim...	LOW
exportAllDataFields	String	None	None	Select the export mode for all data fields (None=no export, ASCI...	LOW
exportDensity	String	None	None	Select the export mode for density data fields (None=no export,...	LOW

Figure 43: Set up of the dust monitors time step

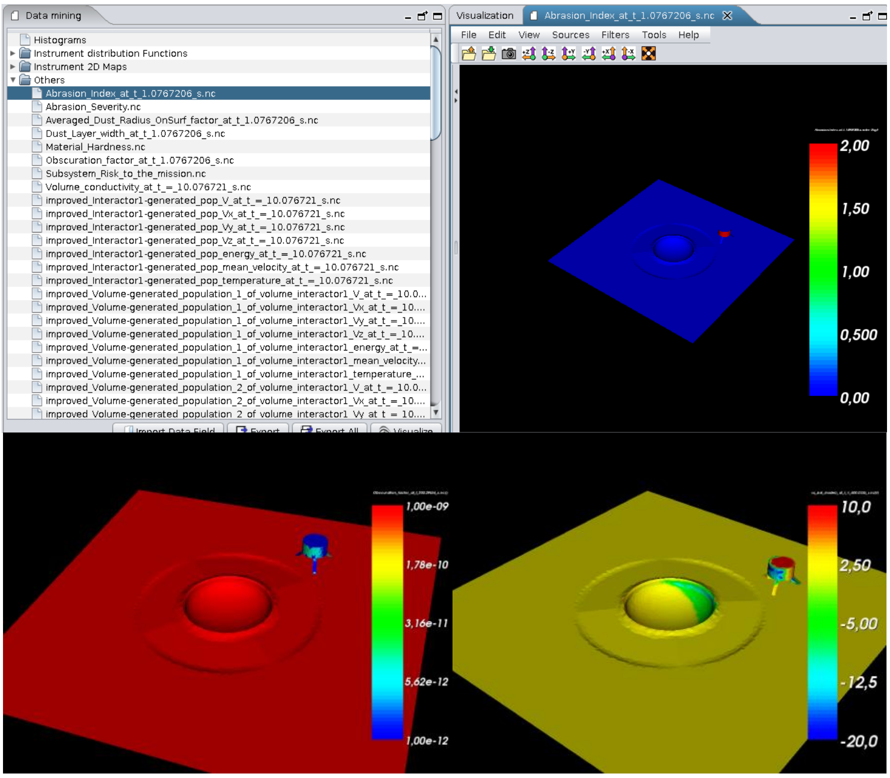

Data mining
Histograms
Instrument distribution Functions
Instrument 2D Maps
Others
Abrasion_Index_at_t_1.0767206_s.nc
Abrasion_Severity.nc
Averaged_Dust_Radius_OnSurf_factor_at_t_1.0767206_s.nc
Dust_Layer_width_at_t_1.0767206_s.nc
Material_Hardness.nc
Obscuration_factor_at_t_1.0767206_s.nc
Subsystem_Risk_to_the_mission.nc
Volume_conductivity_at_t_=_10.076721_s.nc
improved_Interactor1-generated_pop_V_at_t_=_10.076721_s.nc
improved_Interactor1-generated_pop_Vx_at_t_=_10.076721_s.nc
improved_Interactor1-generated_pop_Vy_at_t_=_10.076721_s.nc
improved_Interactor1-generated_pop_Vz_at_t_=_10.076721_s.nc
improved_Interactor1-generated_pop_energy_at_t_=_10.076721_s.nc
improved_Interactor1-generated_pop_mean_velocity_at_t_=_10.076721_s.nc
improved_Interactor1-generated_pop_temperature_at_t_=_10.076721_s.nc
improved_Volume-generated_population_1_of_volume_interactor1_V_at_t_=_10.0...
improved_Volume-generated_population_1_of_volume_interactor1_Vx_at_t_=_10....
improved_Volume-generated_population_1_of_volume_interactor1_Vy_at_t_=_10....
improved_Volume-generated_population_1_of_volume_interactor1_Vz_at_t_=_10....
improved_Volume-generated_population_1_of_volume_interactor1_energy_at_t_=...
improved_Volume-generated_population_1_of_volume_interactor1_mean_velocity...
improved_Volume-generated_population_1_of_volume_interactor1_temperature_...
improved_Volume-generated_population_2_of_volume_interactor1_V_at_t_=_10.0...
improved_Volume-generated_population_2_of_volume_interactor1_Vx_at_t_=_10....
improved_Volume-generated_population_2_of_volume_interactor1_Vy_at_t_=_10....
Import Data Field
Export
Export All
Visualize
Visualization
Abrasion_Index_at_t_1.0767206_s.nc
File Edit View Sources Filters Tools Help
2,00
1,50
1,00
0,500
0,00
1,00e-09
1,78e-10
3,16e-11
5,62e-12
1,00e-12
10,0
2,50
-5,00
-12,5
-20,0

Figure 44: 2D maps of the risk induced by abrasion [top], the obscuration factor [bottom-left] and the potential [bottom-right]

5.2.3. 3D dust moments in the simulation volume

3D dust moments are provided at **Data Mining** level in the "Other" and "Volume densities" folders, including:

- dusts_V: mean velocity magnitude
- dusts_Vx-y-z: mean velocity components
- dusts_energy: mean energy
- dusts_temperature: temperature
- dusts_dustMeanCharge : mean electrical charge expressed in Coulomb unit
- dusts_dustMeanRadius: mean radius expressed in meter unit
- dusts_dustObscur: obscuration level defined as the summed cross section of dusts divided by the local cell cross section (0.01 indicates that luminosity would be decreased by 1% due to dust)
- dusts_numberDensity: density in particle per cubic meter

To activate all dust maps the parameter **dustMapsMonitorStep** shall be defined at Global Editor level with the default value of -10 (indicates that the dust maps are produced every 1/10 of the simulation duration). In the event it does not already exist, please use the "add" button and modify the *newParameter* so-generated.

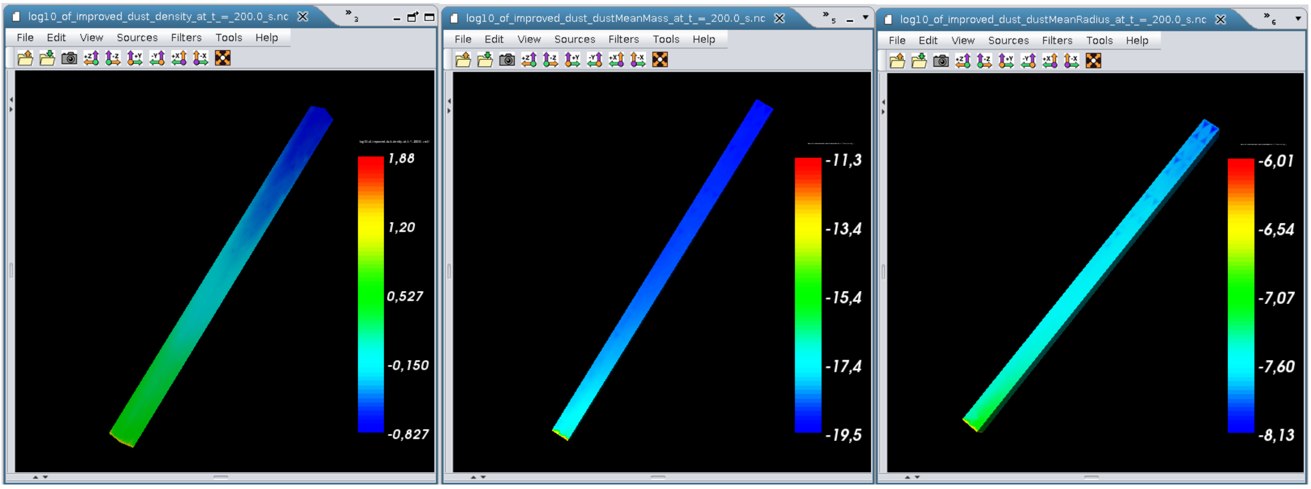

log10_of_improved_dust_density_at_t_=_200.0_s.nc
log10_of_improved_dust_dustMeanMass_at_t_=_200.0_s.nc
log10_of_improved_dust_dustMeanRadius_at_t_=_200.0_s.nc
File Edit View Sources Filters Tools Help
1,88
1,20
0,527
-0,150
-0,827
-11,3
-13,4
-15,4
-17,4
-19,5
-6,01
-6,54
-7,07
-7,60
-8,13

Figure 45: Dust density, mean mass and mean radius in volume (log scales)

When dusts are present in the simulation, the velocity and temperature monitoring is activated for all species, otherwise these quantities are not monitored by default, following the SPIS-SCIENCE/GEO standard. This monitoring can be controlled by using the “*momentsMonitoringFlag*” global parameter (0=no monitoring; 1=monitoring). The monitoring of the vector component of the velocity, electric and magnetic field can be controlled by using the “*vectorMonitoringFlag*”(0=no monitoring; 1=monitoring).

5.2.4. Default dust trajectories

It is possible to set-up directly in the global parameters the tracking of a given number of dusts in the simulation. Contrary to the dust trajectory sensor, these dusts are “real” simulation macroparticles and not test-particles.

To enable the dust trajectory tracing, the global parameter **dustTrajFlag** must be set to 1 (it is by default). There are **particleTrajectoriesNb** dust trajectories that will be traced.

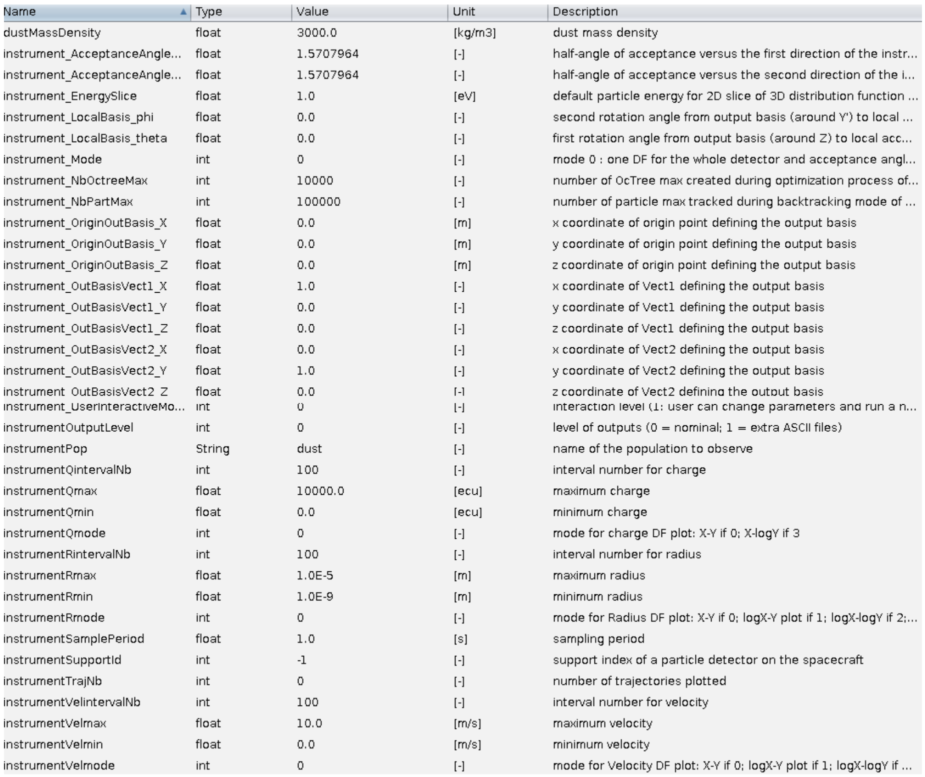
5.3. Dust particle detector

A special type of particle detector, which uses backtracking to determine the distribution of dusts on a particular surface, has been developed. It can be set up in the UI by selecting “Dust Particle Detector” among the dust instruments. The name of the population to be used for dust is **dust** (by default).

The surface on which the dust particles are collected is to be specified in the **instrumentSupportId** field. This ID must be attached to a surface in the group editor.

Apart from the abovementioned parameter, the instrument should be usable with the default parameters. However, depending on the type of surface collecting the dusts, it may be needed to modify the orientation of the output basis.

The outputs from this instruments are saved in netCDF file format by default, but ACSII outputs, including a Gmsh representation of the 3D distribution (radius*velocity*charge) can be obtained by setting the parameter **instrumentOutputLevel** to 1.



Name	Type	Value	Unit	Description
dustMassDensity	float	3000.0	[kg/m3]	dust mass density
instrument_AcceptanceAngle...	float	1.5707964	[-]	half-angle of acceptance versus the first direction of the instr...
instrument_AcceptanceAngle...	float	1.5707964	[-]	half-angle of acceptance versus the second direction of the i...
instrument_EnergySlice	float	1.0	[eV]	default particle energy for 2D slice of 3D distribution function ...
instrument_LocalBasis_phi	float	0.0	[-]	second rotation angle from output basis (around Y') to local ...
instrument_LocalBasis_theta	float	0.0	[-]	first rotation angle from output basis (around Z) to local acc...
instrument_Mode	int	0	[-]	mode 0 : one DF for the whole detector and acceptance angl...
instrument_NbOctreeMax	int	10000	[-]	number of OcTree max created during optimization process of...
instrument_NbPartMax	int	100000	[-]	number of particle max tracked during backtracking mode of ...
instrument_OriginOutBasis_X	float	0.0	[m]	x coordinate of origin point defining the output basis
instrument_OriginOutBasis_Y	float	0.0	[m]	y coordinate of origin point defining the output basis
instrument_OriginOutBasis_Z	float	0.0	[m]	z coordinate of origin point defining the output basis
instrument_OutBasisVect1_X	float	1.0	[-]	x coordinate of Vect1 defining the output basis
instrument_OutBasisVect1_Y	float	0.0	[-]	y coordinate of Vect1 defining the output basis
instrument_OutBasisVect1_Z	float	0.0	[-]	z coordinate of Vect1 defining the output basis
instrument_OutBasisVect2_X	float	0.0	[-]	x coordinate of Vect2 defining the output basis
instrument_OutBasisVect2_Y	float	1.0	[-]	y coordinate of Vect2 defining the output basis
instrument_OutBasisVect2_Z	float	0.0	[-]	z coordinate of Vect2 defining the output basis
instrument_UserInteractiveMo...	int	0	[-]	interaction level (1: user can change parameters and run a n...
instrumentOutputLevel	int	0	[-]	level of outputs (0 = nominal; 1 = extra ASCII files)
instrumentPop	String	dust	[-]	name of the population to observe
instrumentQintervalNb	int	100	[-]	interval number for charge
instrumentQmax	float	10000.0	[ecu]	maximum charge
instrumentQmin	float	0.0	[ecu]	minimum charge
instrumentQmode	int	0	[-]	mode for charge DF plot: X-Y if 0; X-logY if 3
instrumentRintervalNb	int	100	[-]	interval number for radius
instrumentRmax	float	1.0E-5	[m]	maximum radius
instrumentRmin	float	1.0E-9	[m]	minimum radius
instrumentRmode	int	0	[-]	mode for Radius DF plot: X-Y if 0; logX-Y plot if 1; logX-logY if 2;...
instrumentSamplePeriod	float	1.0	[s]	sampling period
instrumentSupportId	int	-1	[-]	support index of a particle detector on the spacecraft
instrumentTrajNb	int	0	[-]	number of trajectories plotted
instrumentVelintervalNb	int	100	[-]	interval number for velocity
instrumentVelmax	float	10.0	[m/s]	maximum velocity
instrumentVelmin	float	0.0	[m/s]	minimum velocity
instrumentVelmode	int	0	[-]	mode for Velocity DF plot: X-Y if 0; logX-Y plot if 1; logX-logY if ...

Figure 46 Dust particle detector parameters.

The instrument backtracks particles in a smart way (using a 3D O.C. Tree) from the instrument surface to boundary and spacecraft surfaces. If the backtracked particles first hit an open boundary, it counts for zero for the distribution on the instrument surface. If it hits a “spacecraft” (including lunar) surface, a statistical weight is determined that is added to the dust distribution on the instrument surface. This weight depends on the dust distribution on the “spacecraft” surface and is determined differently depending on the type of surface:

_ for surface corresponding to a user defined source of dust (section 4.3.3.2), the weight is the value of the dust distribution for the dust velocity, charge and radius on the spacecraft surface, as given by the backtracking. Note that for the backtracking purpose, Dirac distributions have a 1% standard deviation.

_ for surface corresponding to a default source of dust (section 4.3.3.1), the weight depends on the dust velocity, charge and radius on the spacecraft surface, as given by the backtracking. It is the product of the radius distribution, of a charge distribution (Gaussian centred on $Q=4\pi\varepsilon_0 r_d V$, with a standard deviation = 10%), and of a distribution in velocity derived from a $1/\beta^2$distribution of the field amplification factor, β. From the β of the dust is obtained by equating the electrostatic potential energy of the dust on the surface and the kinetic energy ($\rho r_d^3/2v_d^2$) of the backtracked dust.

_ for surface corresponding to non-dusty surface, the weight is the value of the deposited dust distribution for the dust velocity, charge and radius on the spacecraft surface, as given by the backtracking. The deposited dust distribution is a log-normal distribution, determined by the mean dust radius and the radius logarithm variance on surface which are saved by default.

The outputs of the instruments are 1D velocity, radius and charge distributions in time,

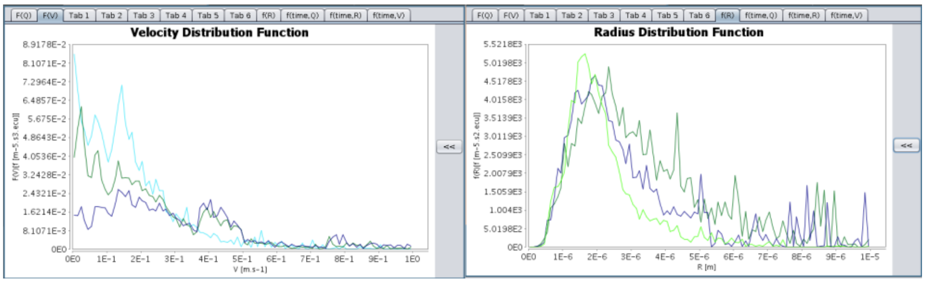


Figure 47:1D radius and velocity distributions versus time displayed as 1D curves.

2D plots of the same,

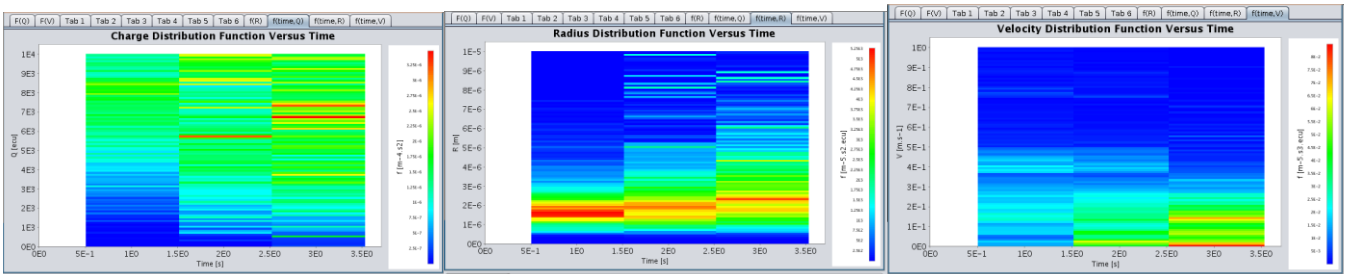

Charge Distribution Function Versus Time
1E4
9E3
8E3
7E3
6E3
5E3
4E3
3E3
2E3
1E3
0E0
Q [ecu]
0E0 5E-1 1E0 1.5E0 2E0 2.5E0 3E0 3.5E0
Time [s]
Radius Distribution Function Versus Time
1E-5
9E-6
8E-6
7E-6
6E-6
5E-6
4E-6
3E-6
2E-6
1E-6
0E0
R [m]
0E0 5E-1 1E0 1.5E0 2E0 2.5E0 3E0 3.5E0
Time [s]
Velocity Distribution Function Versus Time
1E0
9E-1
8E-1
7E-1
6E-1
5E-1
4E-1
3E-1
2E-1
1E-1
0E0
V [m.s-1]
0E0 5E-1 1E0 1.5E0 2E0 2.5E0 3E0 3.5E0
Time [s]

Figure 48: 1D charge, radius and velocity distributions versus time displayed as 2D plots.

and 2D distributions.

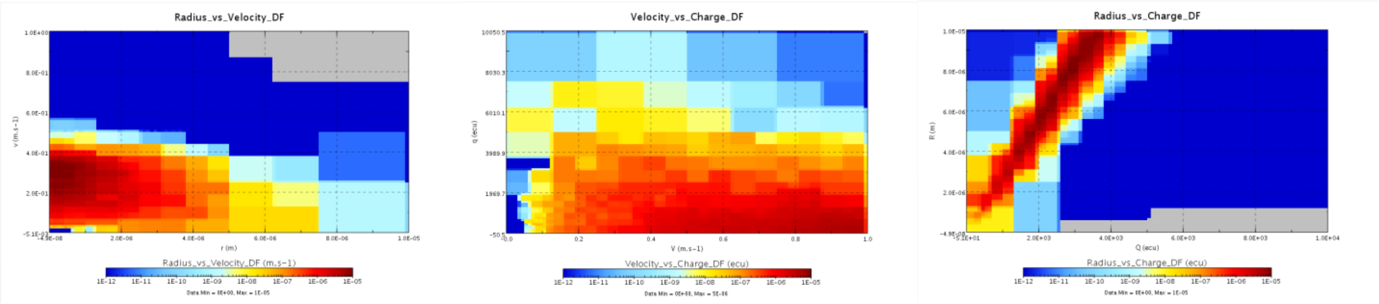


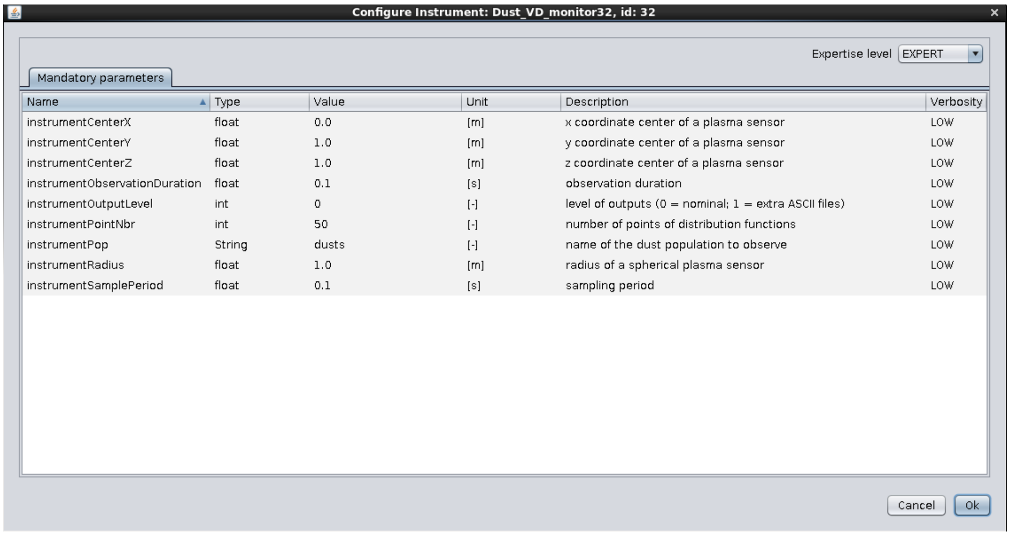
Figure 49: Dust backtracked distributions obtained with the Dust particle detector (radius vs velocity; velocity vs charge; radius vs charge).Visualized in a third party software from the netCDF outputs.

5.4. Dust volume detector

The second set of dust instruments is composed of *DustVolumeDistributionMonitor* objects. This monitor provides regular outputs concerning the dust volume distributions as a function of time:

- 1D distributions: radius distribution function, charge distribution function, energy distribution function
- Average data: mean radius, charge and energy
- Rough dust particle list in ASCII and netCDF format files, written in the NumKernel/Outputs repository

The procedure to define such a sensor is defined in Figure 50. The name of the population to be used for dust is **dust**. The center coordinates must be inside the computational volume. The radius defines a sphere around the center coordinates. Only the dust particles inside the sphere are treated. It is possible to use a large radius to get the information over the full simulation box.



Name	Type	Value	Unit	Description	Verbosity
instrumentCenterX	float	0.0	[m]	x coordinate center of a plasma sensor	LOW
instrumentCenterY	float	1.0	[m]	y coordinate center of a plasma sensor	LOW
instrumentCenterZ	float	1.0	[m]	z coordinate center of a plasma sensor	LOW
instrumentObservationDuration	float	0.1	[s]	observation duration	LOW
instrumentOutputLevel	int	0	[-]	level of outputs (0 = nominal; 1 = extra ASCII files)	LOW
instrumentPointNbr	int	50	[-]	number of points of distribution functions	LOW
instrumentPop	String	dusts	[-]	name of the dust population to observe	LOW
instrumentRadius	float	1.0	[m]	radius of a spherical plasma sensor	LOW
instrumentSamplePeriod	float	0.1	[s]	sampling period	LOW

Figure 50 - How to set up a Dust Volume Distribution Sensor

The information displayed in the UI is:

- Velocity distribution

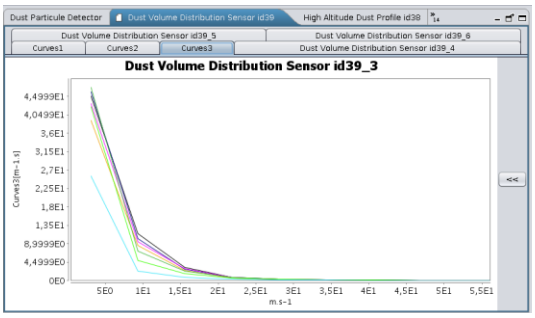

Dust Particule Detector
Dust Volume Distribution Sensor id39
High Altitude Dust Profile id38
14
Dust Volume Distribution Sensor id39_5
Dust Volume Distribution Sensor id39_6
Curves1
Curves2
Curves3
Dust Volume Distribution Sensor id39_4
Dust Volume Distribution Sensor id39_3
4,4999E1
4,0499E1
3,6E1
3,15E1
2,7E1
2,25E1
1,8E1
1,35E1
8,9999E0
4,4999E0
0E0
Curves3[m-1.s]
5E0
1E1
1,5E1
2E1
2,5E1
3E1
3,5E1
4E1
4,5E1
5E1
5,5E1
m.s-1
<<

Figure 51: Live monitoring of the distribution of dust velocities

- Radius distribution

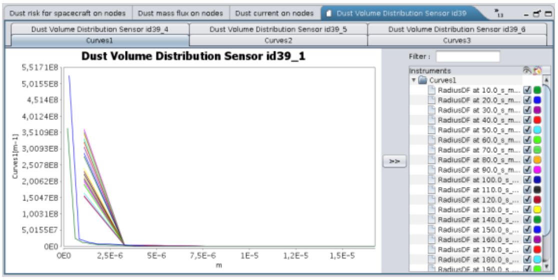

Dust risk for spacecraft on nodes
Dust mass flux on nodes
Dust current on nodes
Dust Volume Distribution Sensor id39
13
Dust Volume Distribution Sensor id39_4
Dust Volume Distribution Sensor id39_5
Dust Volume Distribution Sensor id39_6
Curves1
Curves2
Curves3
Dust Volume Distribution Sensor id39_1
5,5171E8
5,0155E8
4,514E8
4,0124E8
3,5109E8
3,0093E8
2,5078E8
2,0062E8
1,5047E8
1,0031E8
5,0155E7
0E0
Curves1[m-1]
0E0
2,5E-6
5E-6
7,5E-6
1E-5
1,2E-5
1,5E-5
m
>>
Filter :
Instruments
Curves1
RadiusDF at 10.0_s_m... ☑
RadiusDF at 20.0_s_m... ☑
RadiusDF at 30.0_s_m... ☑
RadiusDF at 40.0_s_m... ☑
RadiusDF at 50.0_s_m... ☑
RadiusDF at 60.0_s_m... ☑
RadiusDF at 70.0_s_m... ☑
RadiusDF at 80.0_s_m... ☑
RadiusDF at 90.0_s_m... ☑
RadiusDF at 100.0_s_... ☑
RadiusDF at 110.0_s_... ☑
RadiusDF at 120.0_s_... ☑
RadiusDF at 130.0_s_... ☑
RadiusDF at 140.0_s_... ☑
RadiusDF at 150.0_s_... ☑
RadiusDF at 160.0_s_... ☑
RadiusDF at 170.0_s_... ☑
RadiusDF at 180.0_s_... ☑
RadiusDF at 190.0_s_... ☑

Figure 52: Live monitoring of the distribution of dust radii

- Charge distribution

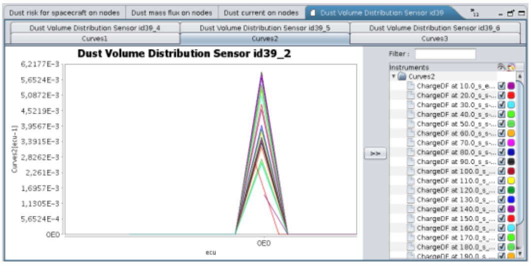

Dust risk for spacecraft on nodes
Dust mass flux on nodes
Dust current on nodes
Dust Volume Distribution Sensor id39
13
Dust Volume Distribution Sensor id39_4
Dust Volume Distribution Sensor id39_5
Dust Volume Distribution Sensor id39_6
Curves1
Curves2
Curves3
Dust Volume Distribution Sensor id39_2
6,2177E-3
5,6524E-3
5,0872E-3
4,5219E-3
3,9567E-3
3,3915E-3
2,8262E-3
2,261E-3
1,6957E-3
1,1305E-3
5,6524E-4
0E0
Curves2[ecu-1]
0E0
ecu
>>
Filter :
Instruments
Curves2
ChargeDF at 10.0_s_e... ☑
ChargeDF at 20.0_s_s-... ☑
ChargeDF at 30.0_s_s-... ☑
ChargeDF at 40.0_s_s-... ☑
ChargeDF at 50.0_s_s-... ☑
ChargeDF at 60.0_s_s-... ☑
ChargeDF at 70.0_s_s-... ☑
ChargeDF at 80.0_s_s-... ☑
ChargeDF at 90.0_s_s-... ☑
ChargeDF at 100.0_s_... ☑
ChargeDF at 110.0_s_... ☑
ChargeDF at 120.0_s_... ☑
ChargeDF at 130.0_s_... ☑
ChargeDF at 140.0_s_... ☑
ChargeDF at 150.0_s_... ☑
ChargeDF at 160.0_s_... ☑
ChargeDF at 170.0_s_... ☑
ChargeDF at 180.0_s_... ☑
ChargeDF at 190.0_s_... ☑

Figure 53: Live monitoring of the distribution of dust charges

- Averaged radius versus time

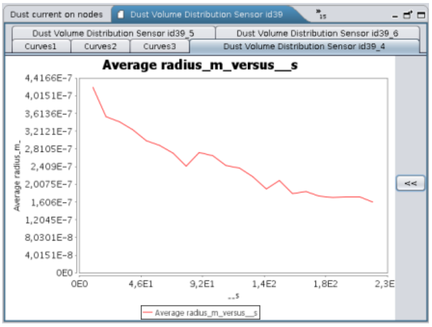

Dust current on nodes
Dust Volume Distribution Sensor id39
15
Dust Volume Distribution Sensor id39_5
Dust Volume Distribution Sensor id39_6
Curves1
Curves2
Curves3
Dust Volume Distribution Sensor id39_4
Average radius_m_versus__s
4,4166E-7
4,0151E-7
3,6136E-7
3,2121E-7
2,8105E-7
2,409E-7
2,0075E-7
1,606E-7
1,2045E-7
8,0301E-8
4,0151E-8
0E0
Average radius_m_
0E0
4,6E1
9,2E1
1,4E2
1,8E2
2,3E
__s
Average radius_m_versus__s
<<

Figure 54: Live monitoring of the averaged radius versus time

- Averaged charge versus time

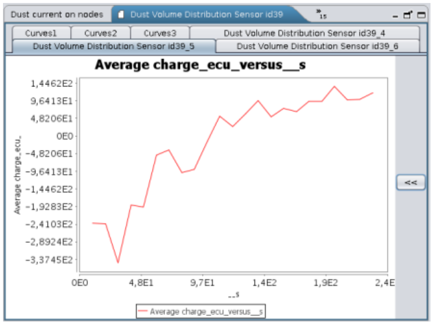


Figure 55: Live monitoring of the averaged charge versus time

- Averaged velocity versus time

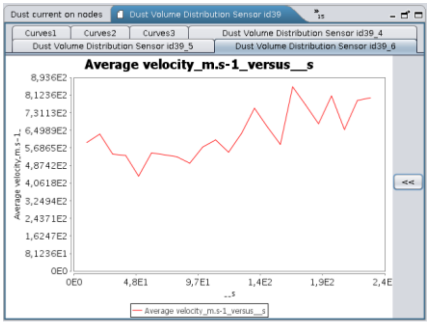
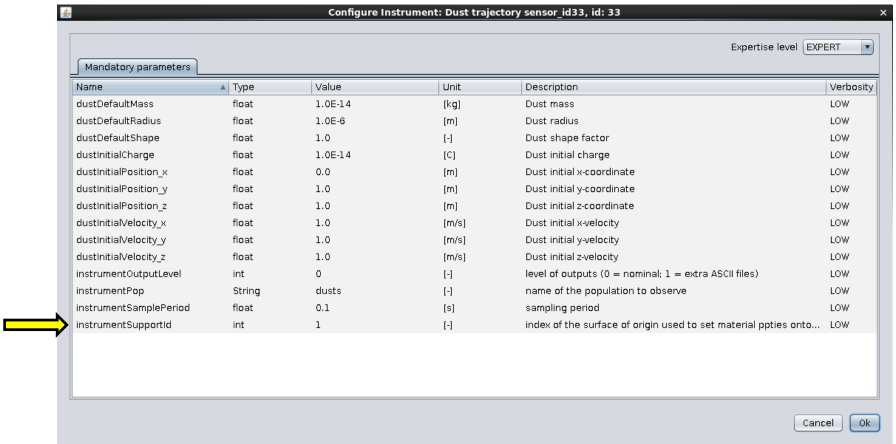


Figure 56: Live monitoring of the averaged velocities versus time

5.5. Dust trajectory sensor

This instrument provides information along the trajectory of a virtual dust. The dust properties are initialized with the material properties of the physical surface group whose *instrumentSupport* is equal to this *instrumentSupportId*, see Figure 57 and if no compliance can be found, the instrument is not built. **It is recommended** to always define an **instrumentSupport** in Figure 19.



Configure Instrument: Dust trajectory sensor_id33, id: 33

Expertise level EXPERT

Mandatory parameters

Name	Type	Value	Unit	Description	Verbosity
dustDefaultMass	float	1.0E-14	[kg]	Dust mass	LOW
dustDefaultRadius	float	1.0E-6	[m]	Dust radius	LOW
dustDefaultShape	float	1.0	[-]	Dust shape factor	LOW
dustInitialCharge	float	1.0E-14	[C]	Dust initial charge	LOW
dustInitialPosition_x	float	0.0	[m]	Dust initial x-coordinate	LOW
dustInitialPosition_y	float	1.0	[m]	Dust initial y-coordinate	LOW
dustInitialPosition_z	float	1.0	[m]	Dust initial z-coordinate	LOW
dustInitialVelocity_x	float	1.0	[m/s]	Dust initial x-velocity	LOW
dustInitialVelocity_y	float	1.0	[m/s]	Dust initial y-velocity	LOW
dustInitialVelocity_z	float	1.0	[m/s]	Dust initial z-velocity	LOW
instrumentOutputLevel	int	0	[-]	level of outputs (0 = nominal; 1 = extra ASCII files)	LOW
instrumentPop	String	dusts	[-]	name of the population to observe	LOW
instrumentSamplePeriod	float	0.1	[s]	sampling period	LOW
instrumentSupportId	int	1	[-]	index of the surface of origin used to set material ppties onto...	LOW

Figure 57 - How to set a dust trajectory sensor

The dust characteristics can be defined by the user (charge, radius, initial position and velocity) and reset during the simulation. The information generated is:

- X-Y-Z components of the particle position

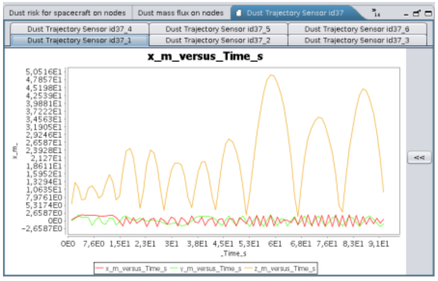


Figure 58: Live monitoring of the position of a dust particle

- X-Y-Z components of the particle velocity

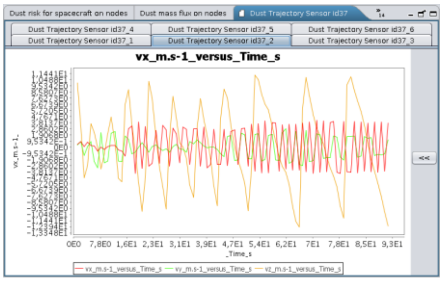


Figure 59: Live monitoring of the velocity of a dust particle

- X-Y-Z components of the electric field at dust position

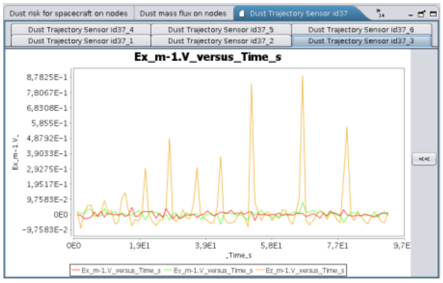


Figure 60: Live monitoring of the electric field felt by a dust particle

- Plasma potential, dust absolute potential, and differential potential with respect to plasma potential

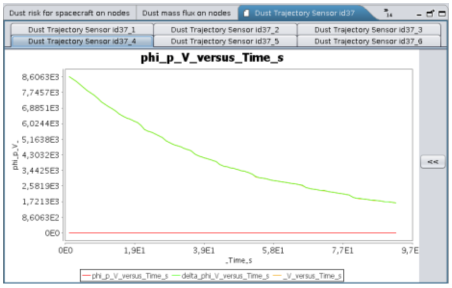

Dust risk for spacecraft on nodes
Dust mass flux on nodes
Dust Trajectory Sensor id37
Dust Trajectory Sensor id37_1
Dust Trajectory Sensor id37_2
Dust Trajectory Sensor id37_3
Dust Trajectory Sensor id37_4
Dust Trajectory Sensor id37_5
Dust Trajectory Sensor id37_6
phi_p_V_versus_Time_s
8,6063E3
7,7457E3
6,8851E3
6,0244E3
5,1638E3
4,3032E3
3,4425E3
2,5819E3
1,7213E3
8,6063E2
0E0
phi_p_V_
0E0
1,9E1
3,9E1
5,8E1
7,7E1
9,7E
_Time_s
<<
phi_p_V_versus_Time_s
delta_phi_V_versus_Time_s
_V_versus_Time_s

Figure 61: Live monitoring of the potential of a dust particle

- Kinetic, electrical potential and total energy of the dust

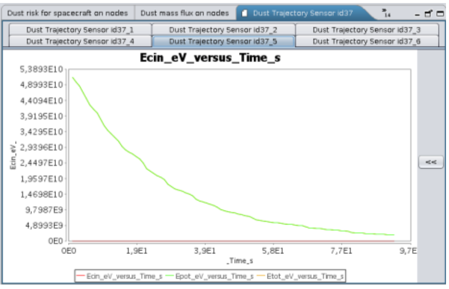


Figure 62: Live monitoring of the kinetic energy of a dust particle

- Charge on the dust

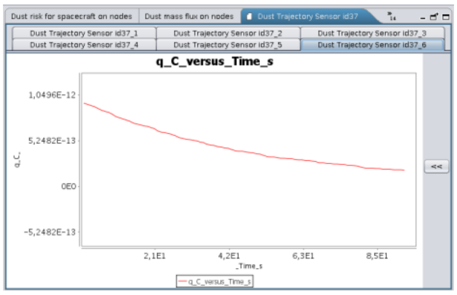
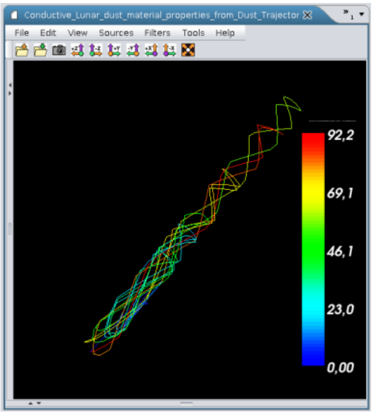

Dust risk for spacecraft on nodes
Dust mass flux on nodes
Dust Trajectory Sensor id37
Dust Trajectory Sensor id37_1
Dust Trajectory Sensor id37_2
Dust Trajectory Sensor id37_3
Dust Trajectory Sensor id37_4
Dust Trajectory Sensor id37_5
Dust Trajectory Sensor id37_6
q_C_versus_Time_s
1,0496E-12
5,2482E-13
0E0
-5,2482E-13
q_C_
2,1E1
4,2E1
6,3E1
8,5E1
_Time_s
<<
q_C_versus_Time_s

Figure 63: Live monitoring of the charge of a dust particle


Conductive_Lunar_dust_material_properties_from_Dust_Trajector
File
Edit
View
Sources
Filters
Tools
Help
92,2
69,1
46,1
23,0
0,00

The instrument also saves the 3D trajectories in 3D that can be seen in the data mining part of the UI.

Figure 64: Trajectory of a dust in the simulation domain

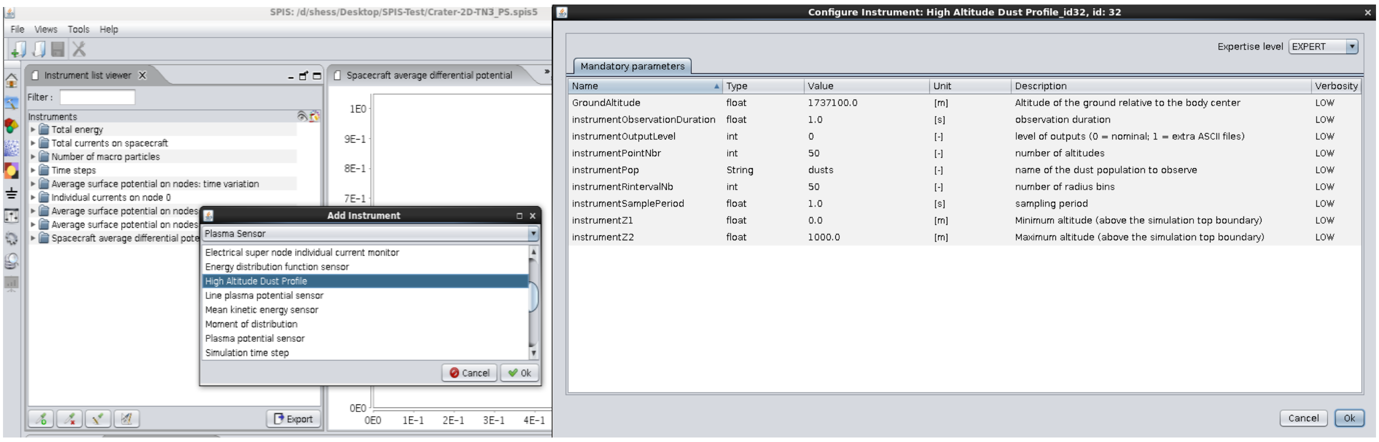
5.6. Dust profiles at high altitude

This instrument performs the calculation of the dust ballistic trajectories, from the top external boundary. It is performed by integrating analytically the dynamics of dust, assuming that the plasma is neutral beyond the top boundary limit. The trajectories of dusts reaching the open boundary of the simulation domain are computed assuming the equation of motion:

$$\frac{d^2r}{dt^2} = \frac{v_\theta^2 - gR}{r + R} + \frac{q_d}{m_d} E_r e^{-r/\lambda_D}$$

where R is the celestial object radius, g and E the gravity acceleration and electric field at the open boundary, respectively.

The celestial object radius R can be set up in the UI by setting the value of the **GroundAltitude** parameter. The minimum and maximum altitudes are fixed by setting the **instrumentZ1** and **instrumentZ2** parameters, respectively. The numbers of bins of the altitude and radius distributions are fixed by setting the **instrumentPointNbr** and the **instrumentRIntervalNbr**, respectively.



Instruments
- Total energy
- Total currents on spacecraft
- Number of macro particles
- Time steps
- Average surface potential on nodes: time variation
- Individual currents on node 0
- Average surface potential on nodes
- Average surface potential on nodes
- Spacecraft average differential pote

Add Instrument: Plasma Sensor
- Electrical super node individual current monitor
- Energy distribution function sensor
- High Altitude Dust Profile
- Line plasma potential sensor
- Mean kinetic energy sensor
- Moment of distribution
- Plasma potential sensor
- Simulation time step

Configure Instrument: High Altitude Dust Profile_id32, id: 32

Name	Type	Value	Unit	Description	Verbosity
GroundAltitude	float	1737100.0	[m]	Altitude of the ground relative to the body center	LOW
instrumentObservationDuration	float	1.0	[s]	observation duration	LOW
instrumentOutputLevel	int	0	[-]	level of outputs (0 = nominal; 1 = extra ASCII files)	LOW
instrumentPointNbr	int	50	[-]	number of altitudes	LOW
instrumentPop	String	dusts	[-]	name of the dust population to observe	LOW
instrumentRintervalNb	int	50	[-]	number of radius bins	LOW
instrumentSamplePeriod	float	1.0	[s]	sampling period	LOW
instrumentZ1	float	0.0	[m]	Minimum altitude (above the simulation top boundary)	LOW
instrumentZ2	float	1000.0	[m]	Maximum altitude (above the simulation top boundary)	LOW

Figure 65: Selection and set-up of the high altitude dust profile instrument in the UI.

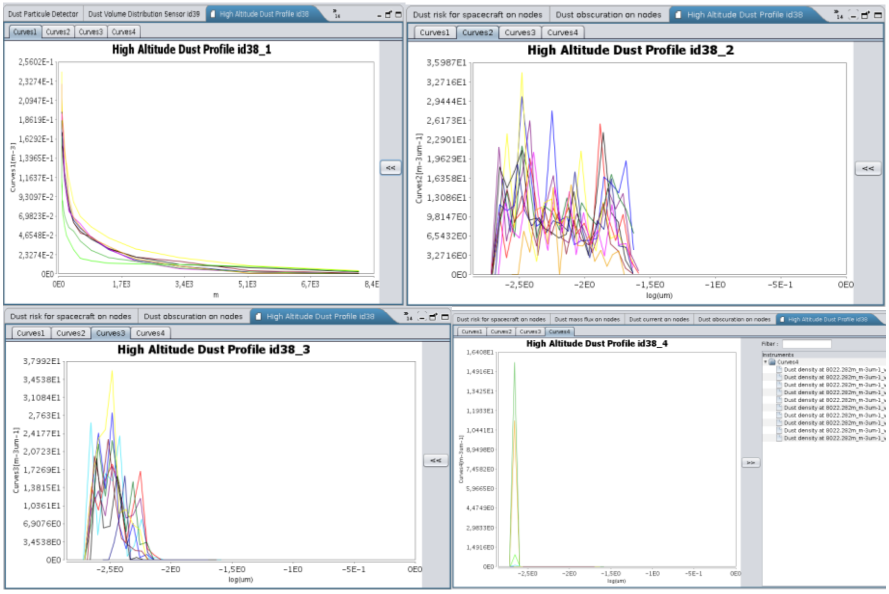

Dust Particule Detector
Dust Volume Distribution Sensor id39
High Altitude Dust Profile id38
Curves1
Curves2
Curves3
Curves4
High Altitude Dust Profile id38_1
High Altitude Dust Profile id38_2
High Altitude Dust Profile id38_3
High Altitude Dust Profile id38_4
Dust risk for spacecraft on nodes
Dust obscuration on nodes
Dust mass flux on nodes
Dust current on nodes
m
log(um)
0E0
1,7E3
3,4E3
5,1E3
6,7E3
8,4E3
-2,5E0
-2E0
-1,5E0
-1E0
-5E-1
Filter :
Instruments

Figure 66[top-left]Live monitoring of the dust density versus altitude for different times,[top-right, bottom-left, bottom-right] dust radius distributions at 80m, 800m and 8000m of altitude above the ground.

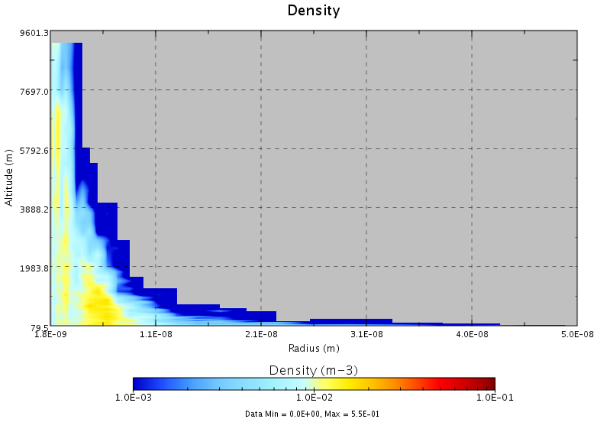


Figure 67: Density of dusts versus altitude and radius obtained with the sensor saved in the netCDF format and plotted in a third-party software

Remarks:

- All results generated by the instruments are stored directly in the .spis5 project in the directory {SPIS5Project}/DefaultStudy/Simulations/Run1/OutputFolder/DataFieldMonitored
- Other results generated by the simulations are stored directly in the .spis5 project in the directory {SPIS5Project}/DefaultStudy/Simulations/Run1/OutputFolder/DataFieldExtracted