**OHB System AG**

J. Laube, B. Zitouni, R. Kast, M. Peukert

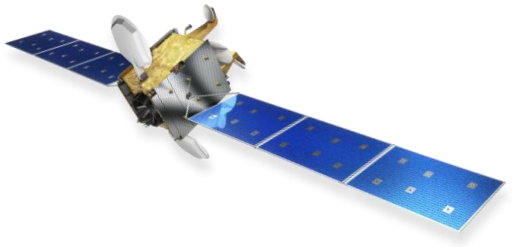
# openPlume for EP and CP plume modelling: tool development status

# OHB System AG: presentation of our company

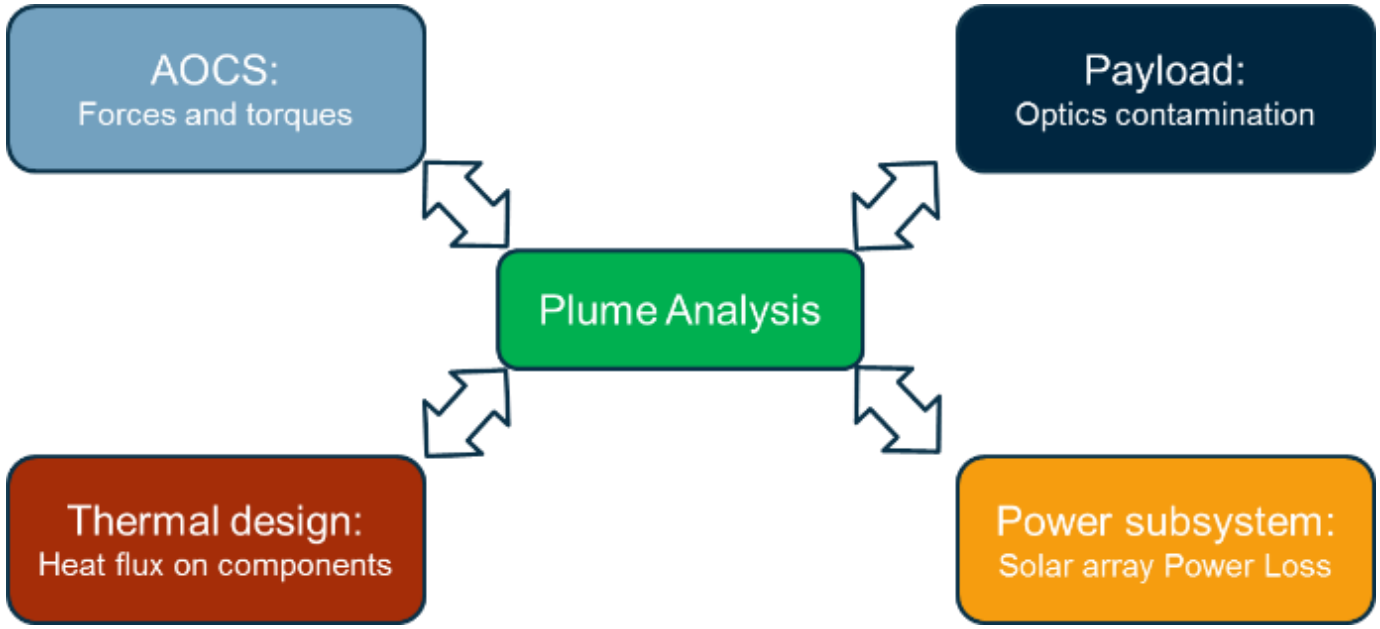
- OHB is a family-run company with the character of a medium-sized business
- From modest beginnings to an internationally recognized and consistent player

- We developed:
  - Galileo navigation satellites constellation (CP)
  - Telecommunication satellite, platform SmallGeo (CP and EP)
  - Observation and Exploration (mainly CP)

- Our Telecommunication spacecrafts based on Electric Propulsion are:
  - SmallGeo: HET SPT100 for Station Keeping
  - Electra: HET thrusters for Orbit Raising and Station Keeping
  - Heinrich Hertz: HEMPT thrusters for Station Keeping



# Importance of the Plume Analysis



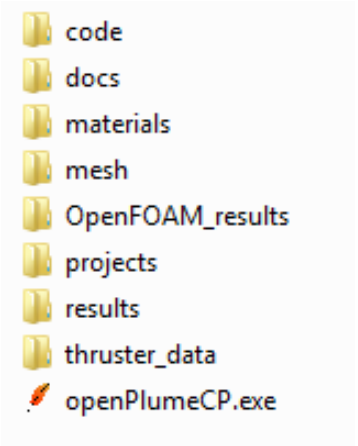
# openPlume project: concept and philosophy

- openPlume provides Fast and simplified calculations using the ray tracing approach
- Gases/Ions are emitted from the thruster and extrapolated to the far field thanks to the $1/r^2$ rule (or any extrapolation method)
- The spacecraft is modelled with 2D structured or unstructured mesh
- The core of the application is python scripts that links the pre-processing to the post-processing.
- Two types of users are imagined:
  - The coders, who have access to the code
  - The analysists, who perform plume analysis

# Items delivery: folders structure

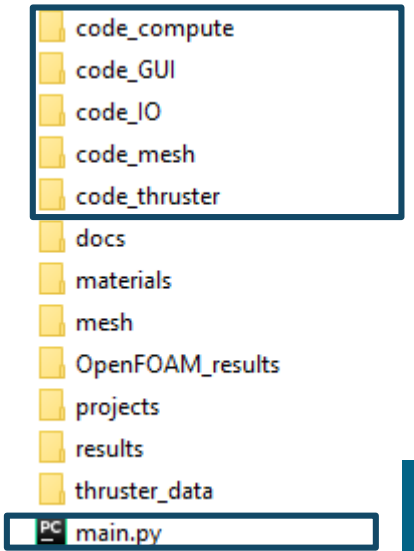
Each of openPlumeCP.zip and openPlumeEP.zip contains an « executable » folder and « scripts » folder

Executable folder




Launch the GUI


Scripts folder




- Python scripts are stored in these additional folders
- They can be easily modified
- main.py launch the whole program, each scripts could be launched in standalone.

**OHB System AG**

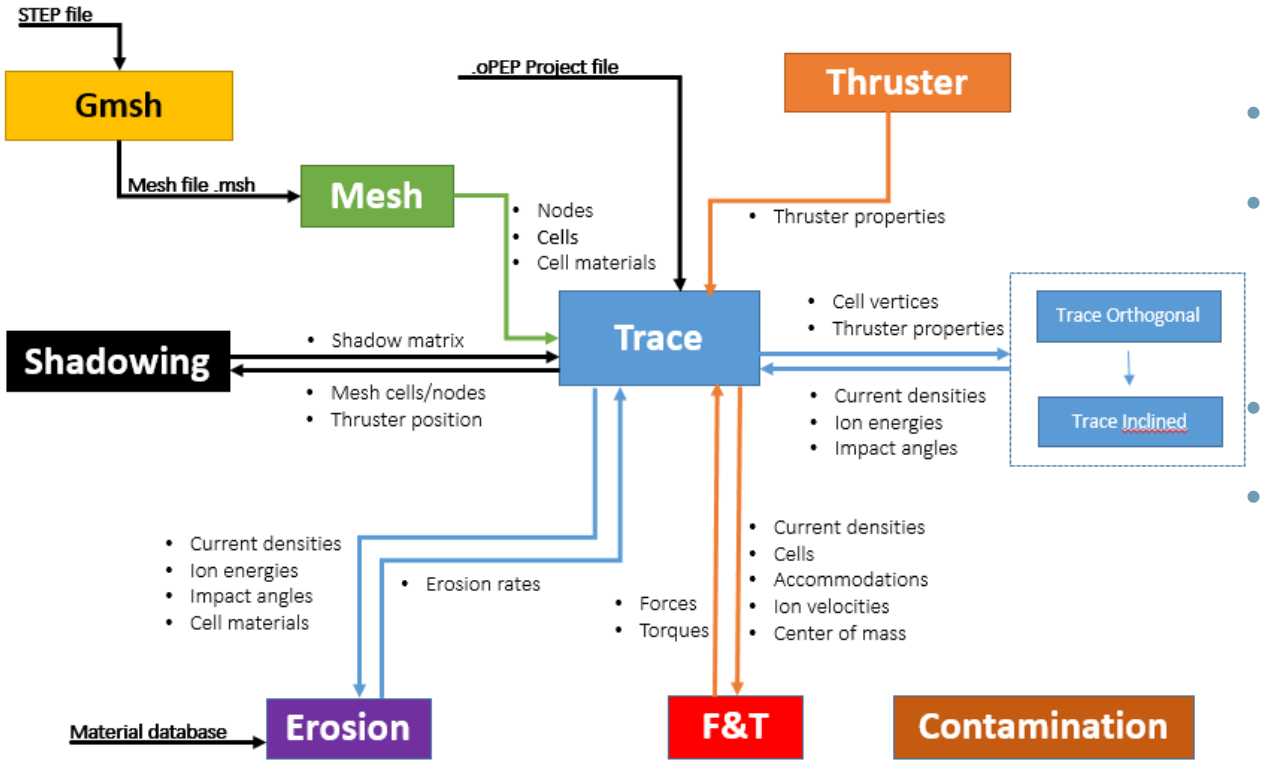
J.Laube, B. Zitouni, R. Kast, M.Peukert

# openPlume for EP plume modelling



# Ray-tracing tool: algorithm and code

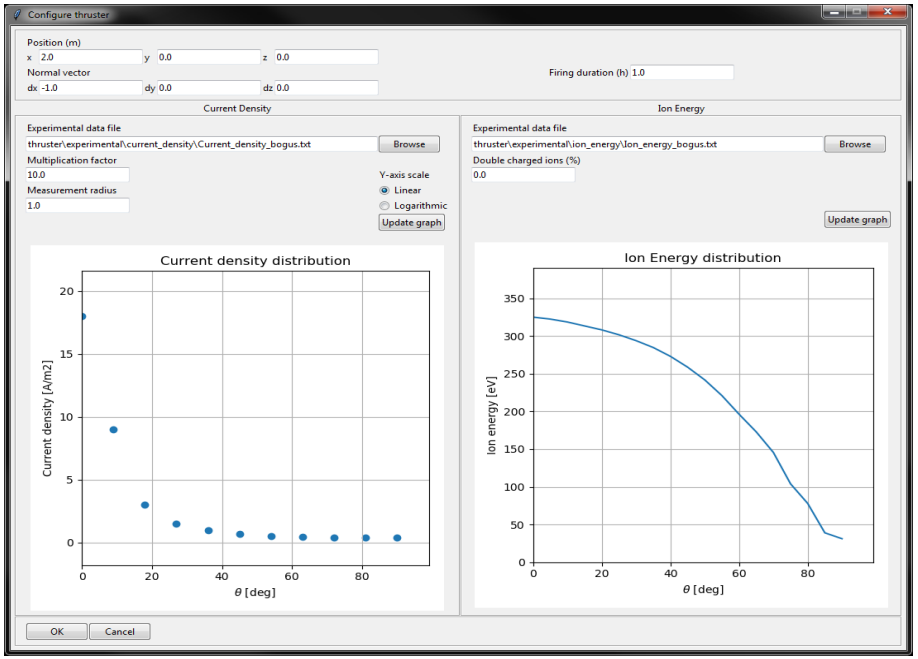
- openPlumeEP is governed by the trace module that makes the connection to all the other modules

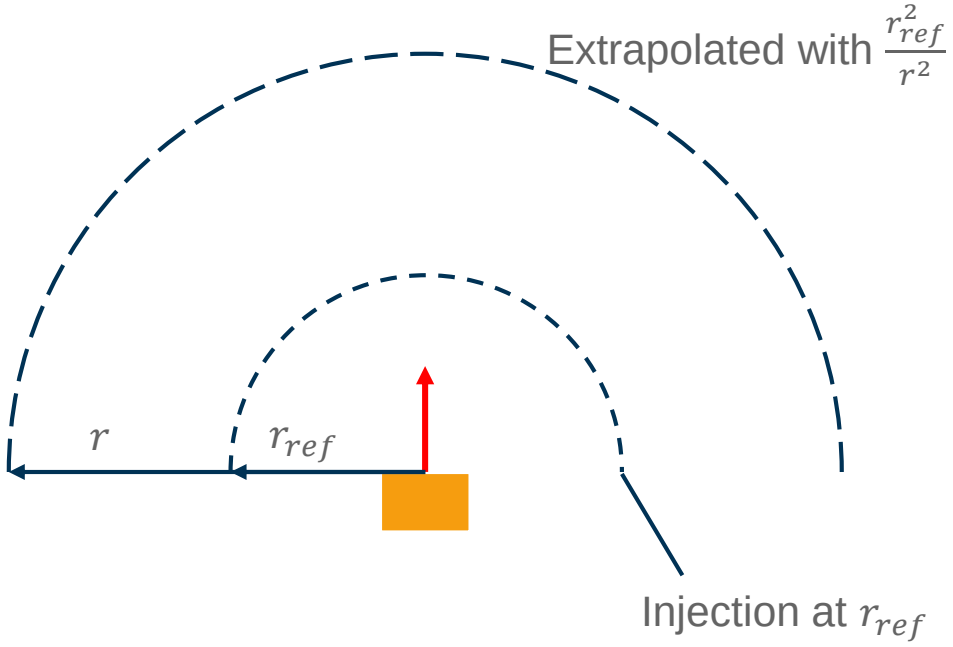



- Mesh loaded from gmsh
- Thruster properties are loaded from experimental data or PIC simulations
- Shadowing calculations
- Extrapolation to the far field

# Thrusters modelling concept

- Thruster modelled as a point source
- Ions are transported to far field with current decreasing following the 1/r² rule
- Possibility to load experimental data, connect to PIC code (PicPluS implemented, SPIS pending, VSTRAP?) and custom fit.



# Thrusters modelling concept

- Multiple charged ions with two methods
  - Mean Method: custom rates for Ions2+, Ions3+ and Ions4+
  - Depending on angle: rate of multiple charged ions depending on angle

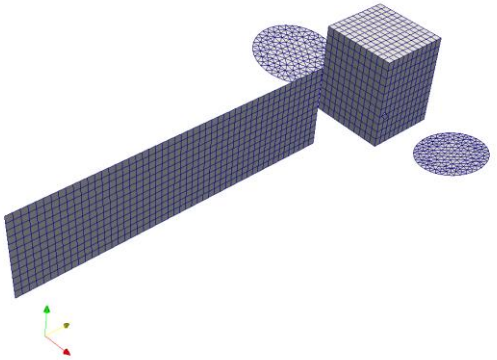

```
#Theta [deg] Energy [eV] 2+ 3+ ...
0.2      244.9 0.35 0.01
5.2      238.9 0.33 0.02
10.2     238.2 0.3 ...
15.1     234.0 0.29
20.1     229.4 0.275
25.0     225.5 0.28
30.0     220.7 0.31
34.9     213.8 0.37
39.9     205.8 0.4
44.8     198.6 0.46
```

# Geometrical modelling: meshing, shadowing and ray tracing

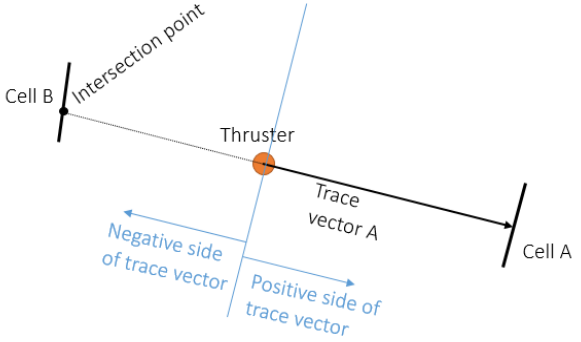
- Before performing the ray-tracing, meshed geometrical model is needed and thus. shadowing is calculated.

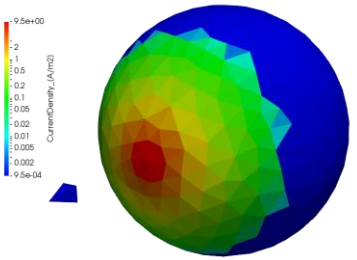
- openPlumeEP uses purely 2D meshes defined on 2D surfaces.
- Gmsh load step files and can mesh them through geo scripts
- Geo files are handy that can be reused for any geometry

```
Merge "mystepfile.stp"
Transfinite Line Transfinite Line
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12} = 10 Using Progression 1;
Transfinite Surface {14, 16, 18, 20,
22, 24};
Physical Surface(21) = {14, 16, 18,
20, 22, 24};
```



- Shadowing calculations are needed to calculate intersection points and thus check is a cell is shadowed by another



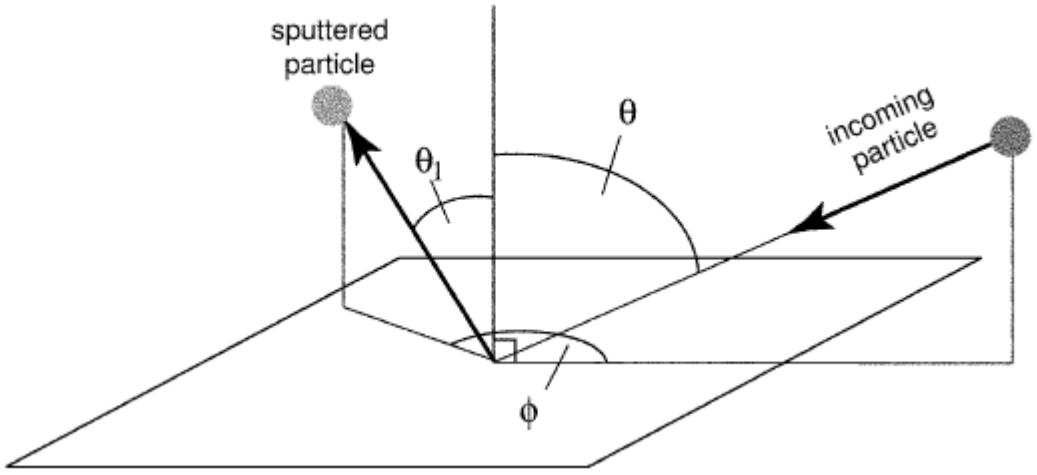



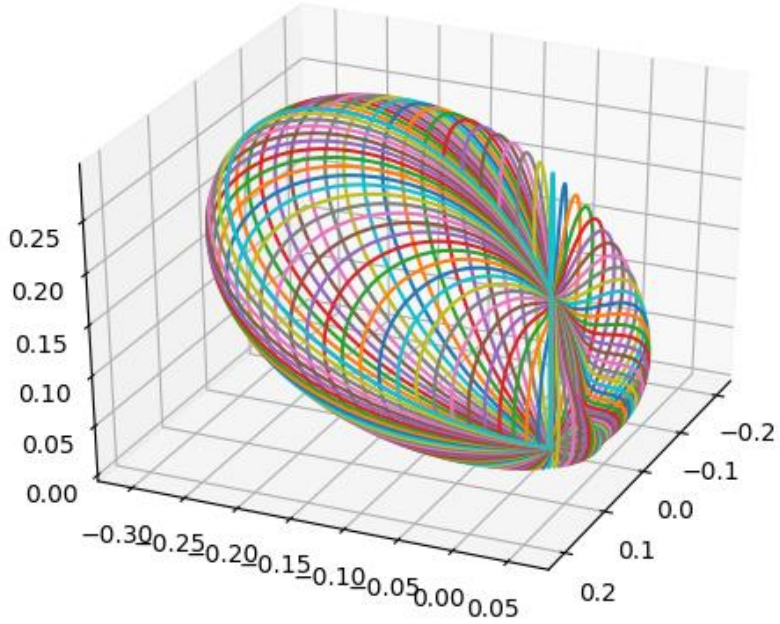
# Contamination modelling

- Computation of the sputtered materials deposition is performed through a distribution function S:

$$contamination = \sum erosion * S \frac{A\ cos\alpha}{r^2}$$

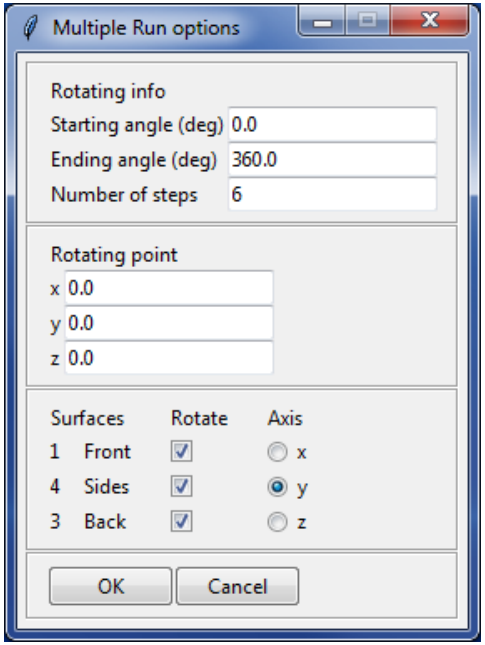
  - Distribution function can be performed with a simple cosine law or quasi-specular

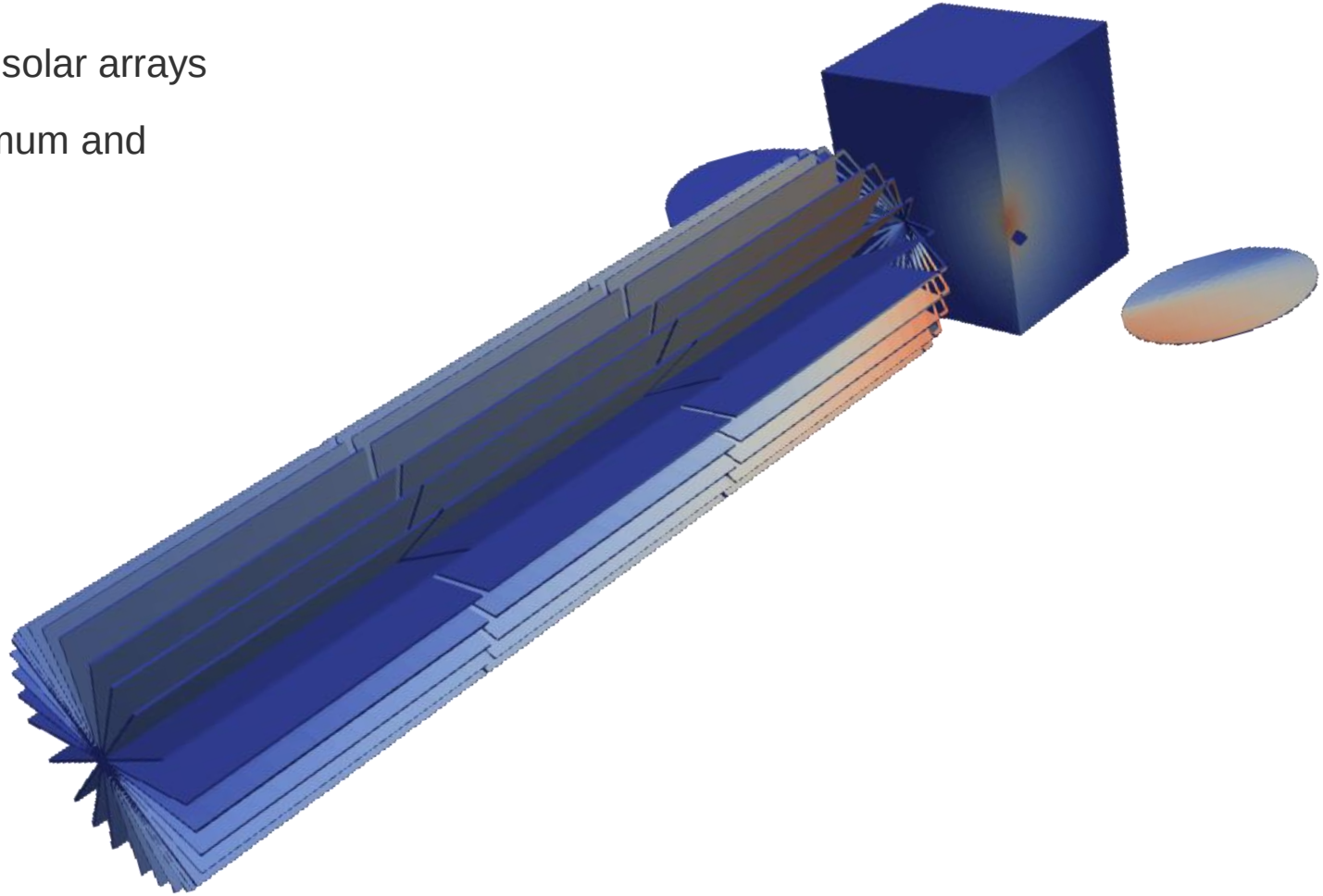


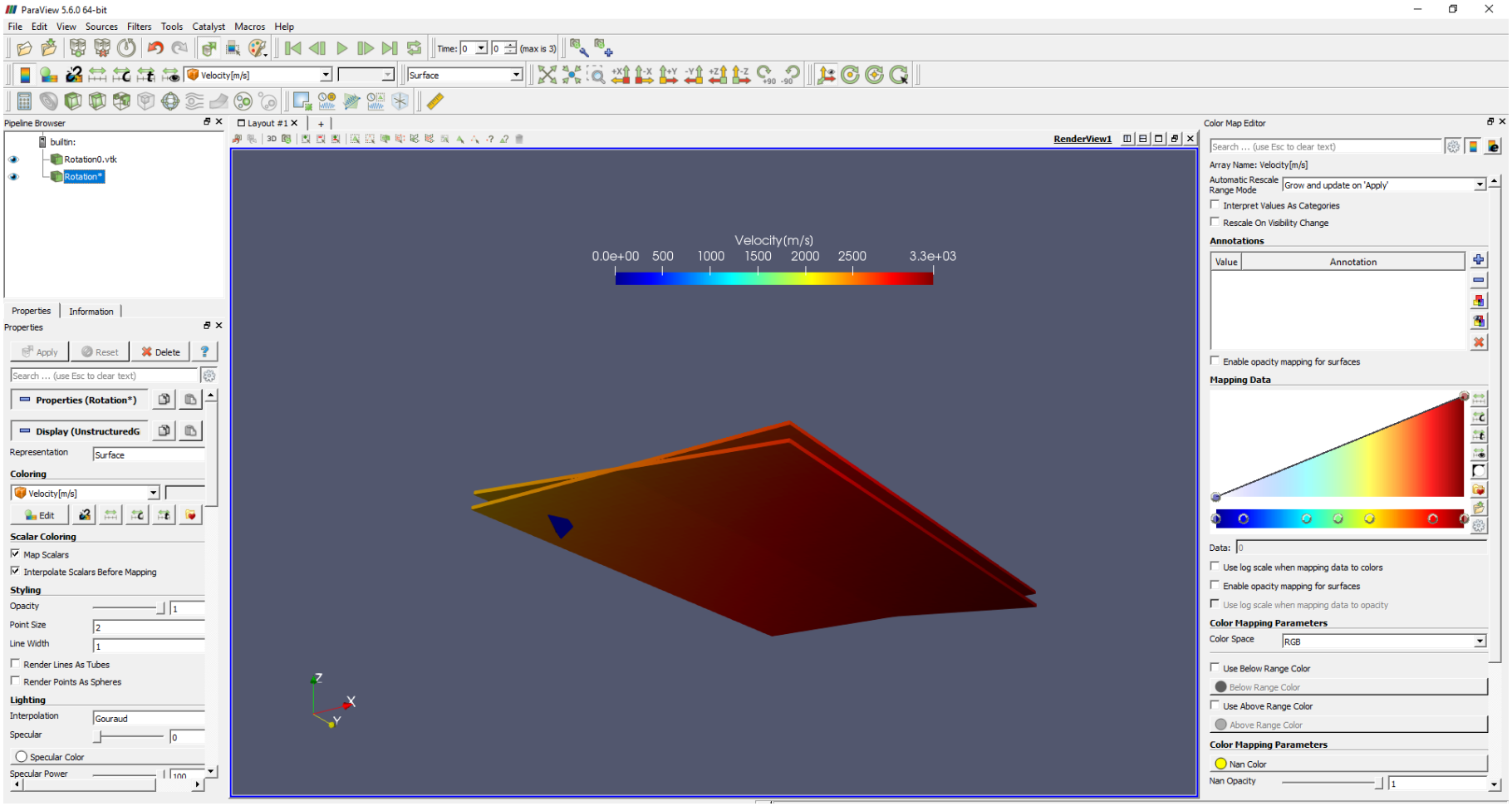
# Rotating modules

- Studied to handle rotating solar arrays
- Generates files with maximum and average values



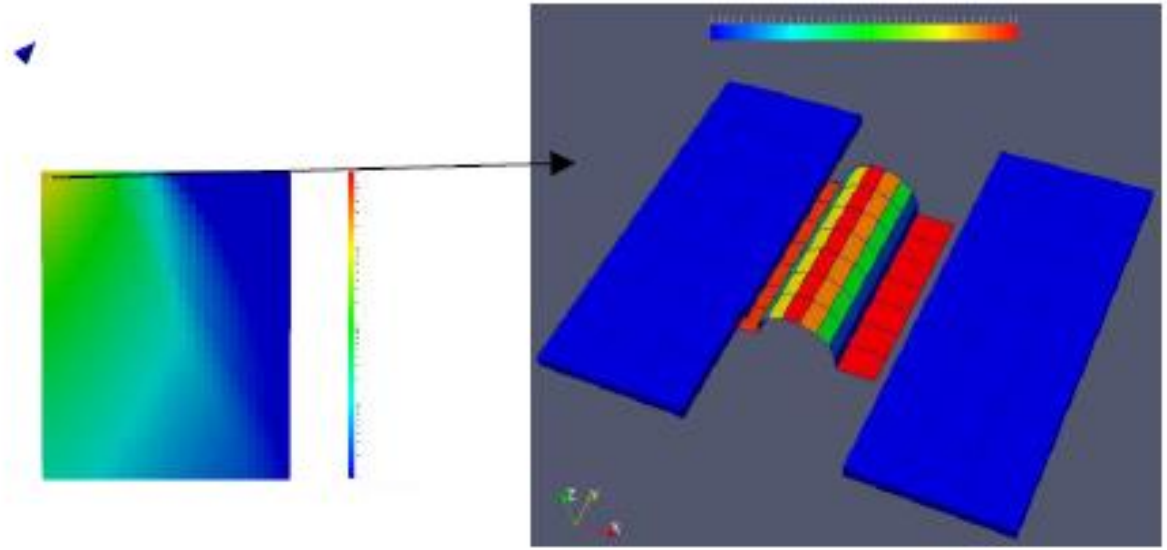



# Post-processing and results in paraview:



# Application case: Solar Array Interconnectors

- A combination of very large and very small geometries can be studied due to the ray-tracing approach.
- Impact of geometry and orientation of the interconnectors can be studied.



# Analytical test cases: Proof of Concept

- **Ray tracing is a robust and simple method.**
- **Impingement on single cells can be checked by solving the equations analytically.**
- **Simple erosion test cases have been run for proof of concept of the erosion function.**
- **Exact agreement with analytical equations.**

| Cell 1 | | | |
|---|---|---|---|
| Position (X,Y,Z) | 0 m | 0.5 m | 1 m |
| Orientation (X,Y,Z) | 0 | 0 | -1 |
| Sputter rate computed | 3.24E-10 m/s | | |
| Sputter rate analytical | 3.24E-10 m/s | | |
| Deviation | 0.0% | | |

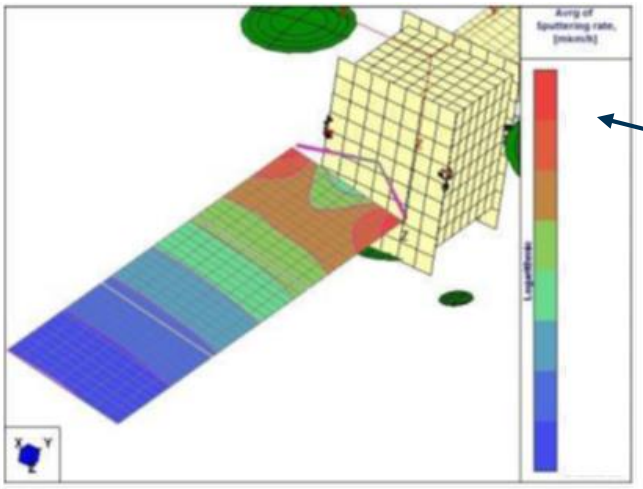
| Cell 2 | | | |
|---|---|---|---|
| Position (X,Y,Z) | 0 m | 1 m | 0.5 m |
| Orientation (X,Y,Z) | 0 | -1 | 0 |
| Sputter rate computed | 1.42E-11 m/s | | |
| Sputter rate analytical | 1.42E-11 m/s | | |
| Deviation | 0.0% | | |

| Thruster | | | |
|---|---|---|---|
| Position (X,Y,Z) | 0 m | 0 m | 0 m |
| Orientation (X,Y,Z) | 0 | 0 | 1 |

# Validation against existing ray-tracing code

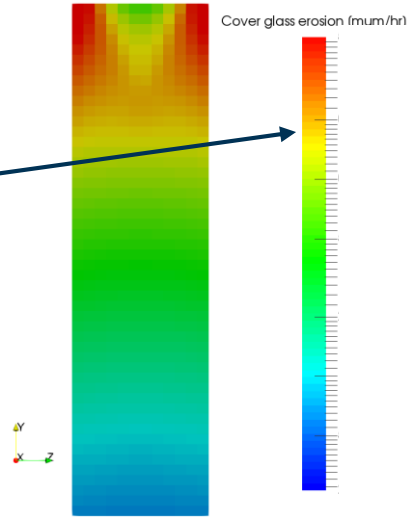
- **openPlume was compared to an existing ray-tracing software**
- **The two tools yield equal results from equal input-data**

Reference case glass erosion:



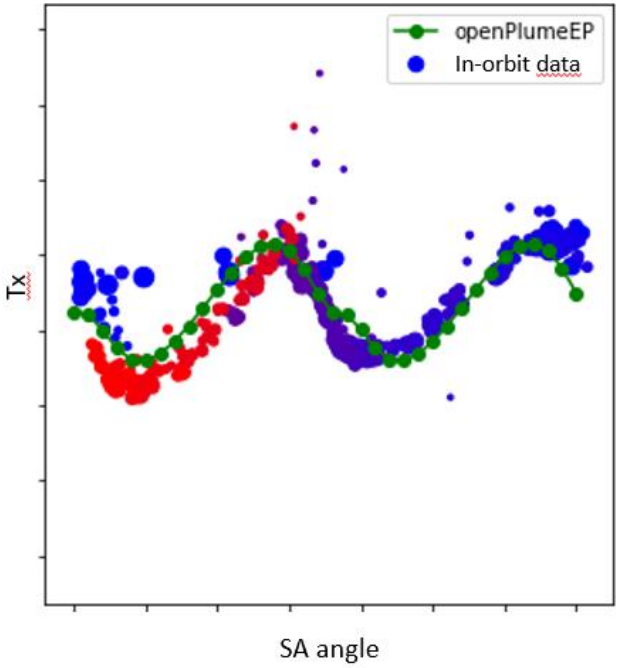
Identical scales

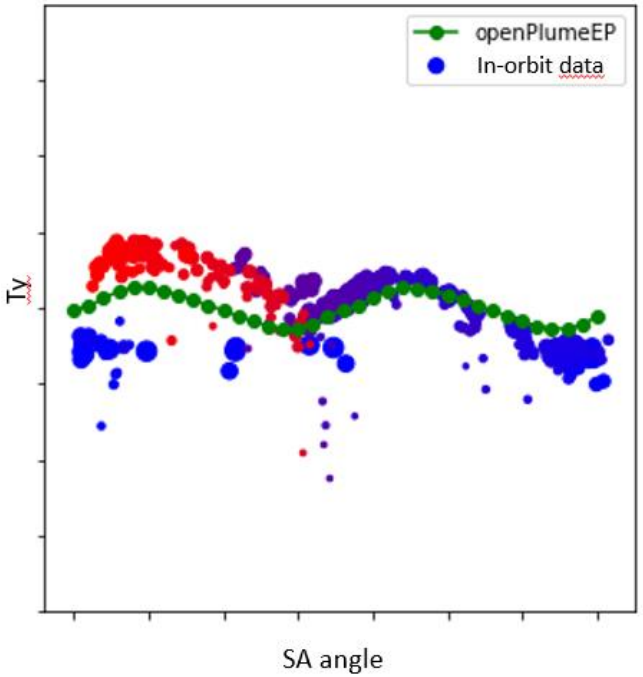
openPlumeEP glass erosion:

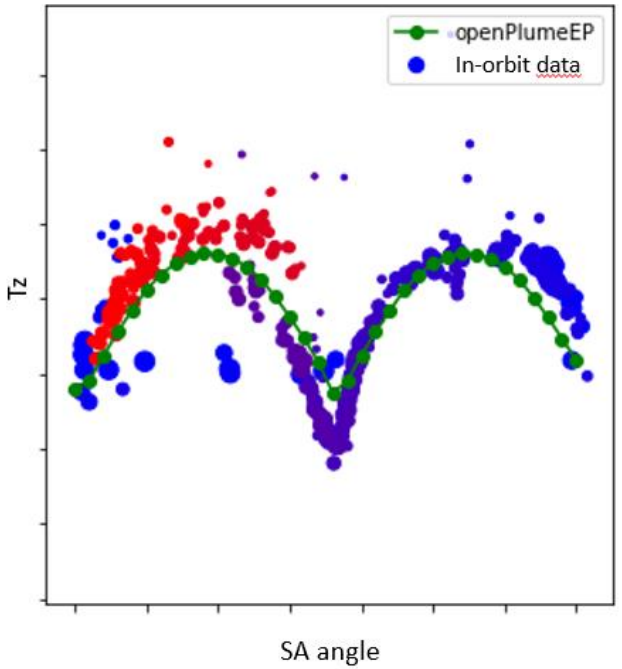


# Validation against in-orbit data

- Very good agreement to in-orbit torques telemetry

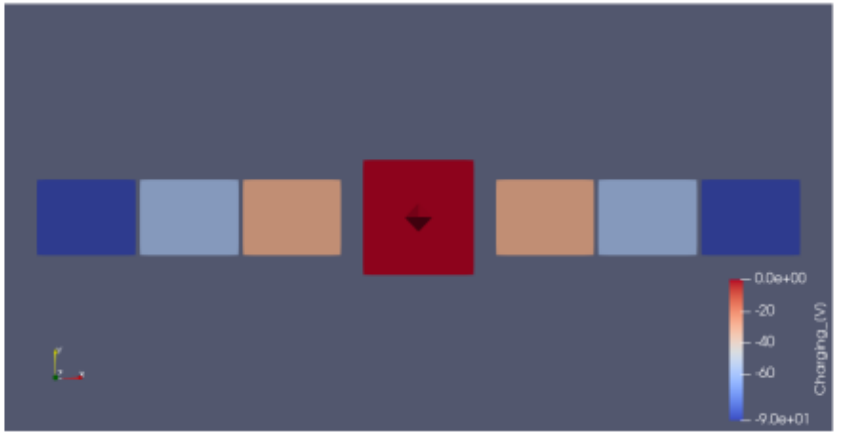


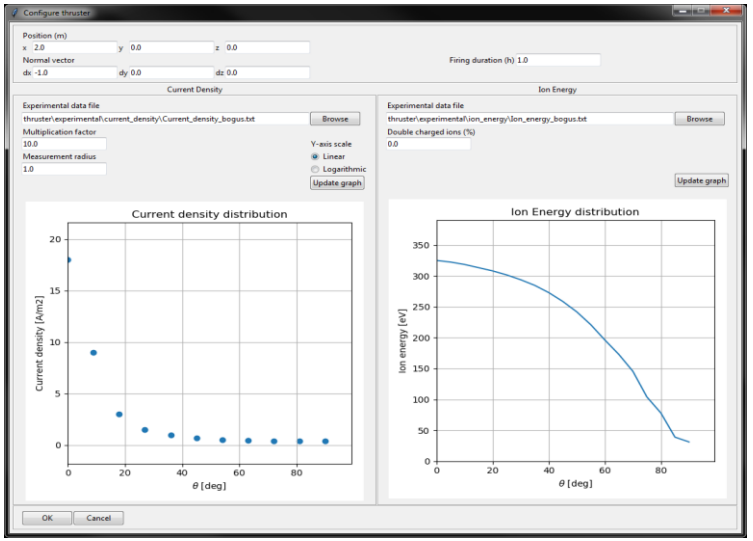



# Future implementations

- Surface charging:
  - Charging effects lead to the Plume deflection/attraction from the charged surfaces.
  - New feature: possibility to load a charging map of the spacecraft surfaces
    - Through surfaces identification
    - By loading a specific file for the cell charging
    - Connection to SPIS?
- Thruster plume input:
  - Loading thruster plume data directly from different PIC simulations output.
    - Loading from SPIS instruments?
    - Automatic generation of instruments files for SPIS?



# Conclusion

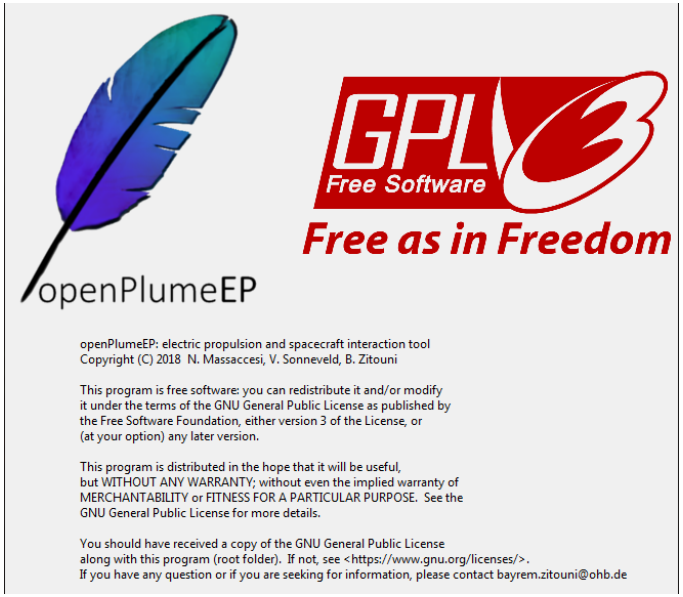
- EP Plume impingement is a constant challenge for spacecraft design

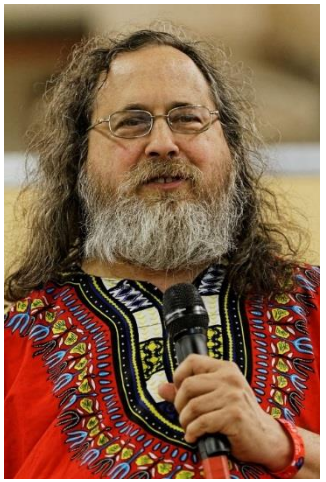
- openPlumeEP has been developed, tested and validated to answer variety of configurations

- Thanks to its flexibility:
  - It has been applied in OHB projects
  - It can be used to study new thrusters
  - It can load any mathematical model with its python modularity

# openPlumeEP: license, training

- GPLv3 license: Free download
- Premium offers for users includes training and support



# Conclusion

- openPlumeCP and EP have been enhanced with several new features
- Possibility to download the tool for free here: **www.openplume.org**
- Training and specific user support can be provided through a premium membership