GRASP

GRASP Software Overview

ATARRI

5th March, 2025

Funded by
the European Union

David Fuertes and all GRASP team

Introduction

GRASP Activities

GRASP Software Overview

Aerosols: Particles in the atmosphere

Sources:

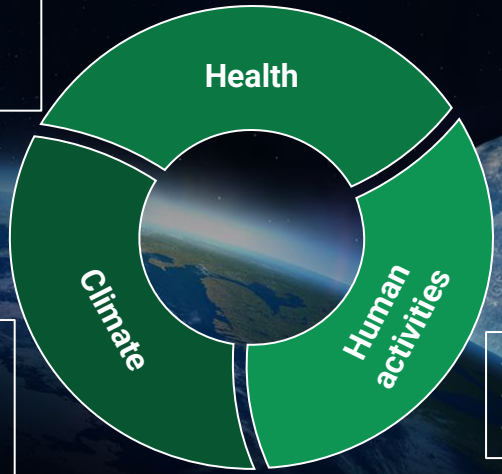
Air quality applications:

- Regions, cities, municipalities
- Pollutant industries
- Mines

Space agencies:

- ESA - EUMETSAT
- Copernicus
- NASA - NOAA

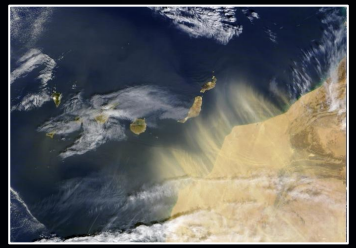
Impact:



- Atmospheric correction
- Precision agriculture
- Solar Radiation

Volcanic ash

Dust



Forest fires

Urban smoke

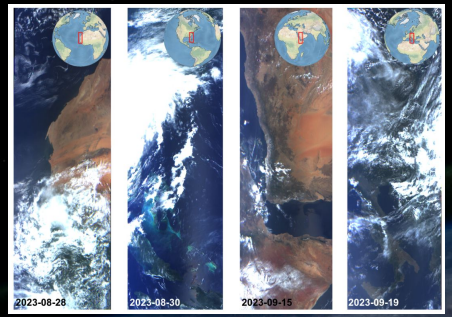
Sea salt

Activities :

Deep expertise in all domains of remote sensing for EO applications.

SATELLITE OPERATOR

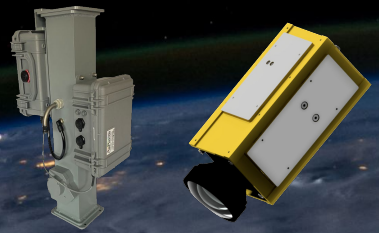
Deployment of 10 cubesat constellation aiming to monitor Air Pollution to be integrated by polluting industries, mines and smart-cities.



HARDWARE

Conception and development of:

- our own ground-based in-situ instrumentation.
- our own space payloads.



SOFTWARE

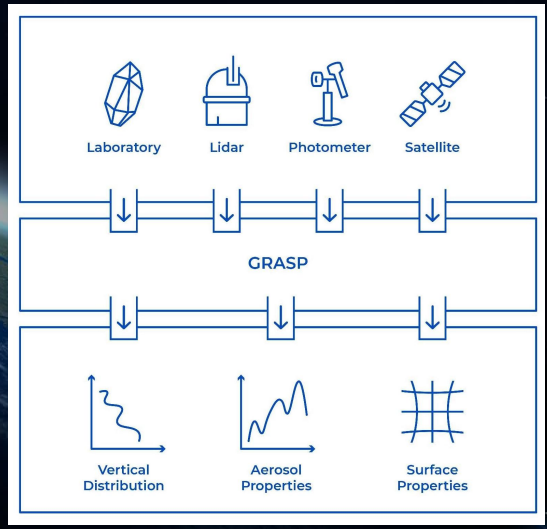
- Retrieval of atmosphere and surface properties (aerosol expertise).
- Retrieval algorithms. Operational products for several public missions:

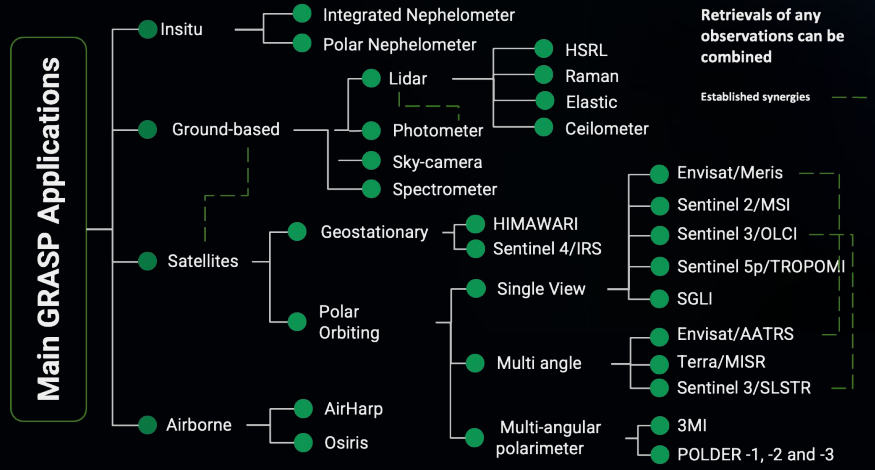
- Sentinel-4
- 3MI
- CO2M

GRASP stands for

Generalized Retrieval of Atmosphere and Surface Properties

GRASP is one of the most advanced algorithms for aerosol, gas and surface properties from any remote sensing source and the combination of them. This inversion algorithm is currently used by main spatial agencies worldwide and was a subject of over 60 scientific publications in the last decade.

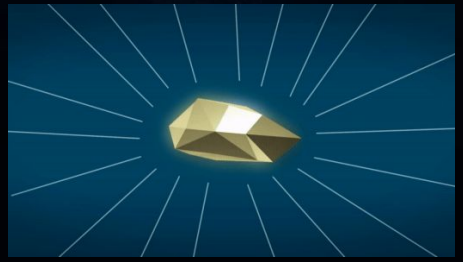




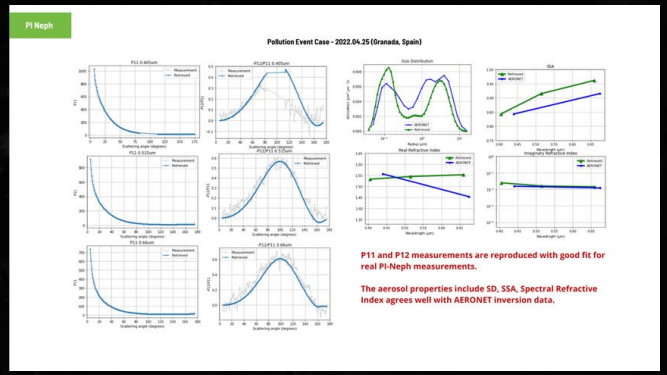
esa

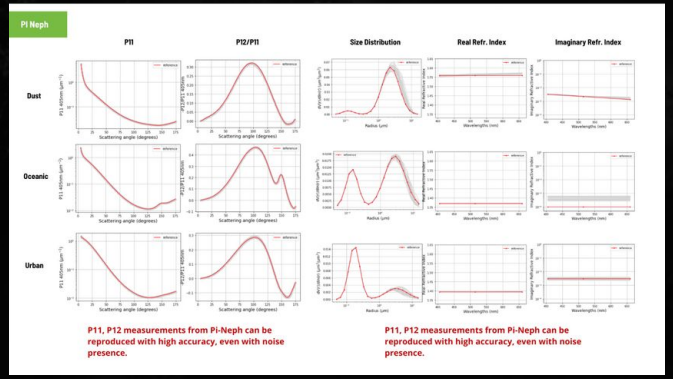
EUMETSAT

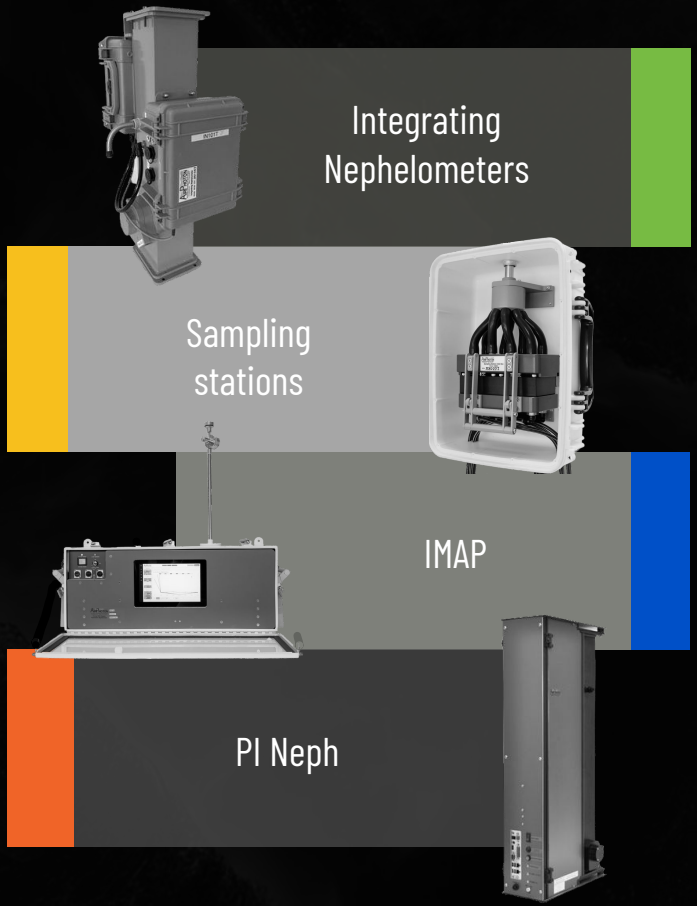
NASA



GRASP/Airphoton instruments







GAPMAP Constellation: Payload description

GAPMAP-0 on board of ADLER-2

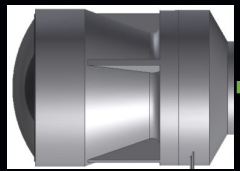
Launched April 2023; Commissioned August 2023

3D render of GAPMAP-0 on board of Adler-2 mission launched on April 2023

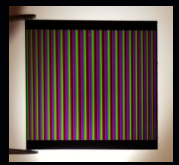
	GAPMAP Constellation
Wavelengths	440, 550, 670, 870nm
Calibration	Full onboard radiometric and polarimetric calibration; solar + moon; ground targets; satellite comparisons.
Nadir pixel resolution	0.58km
Nadir Binned resolution	2.32 km
Cross track Swath	77 deg (1070km@650km orbit)
Along track swath	105.2 deg
Hyper-Angular Sampling	Up to 60 angles
Angular sampling	Hyperangular
Satellite volume	<1.5U

Based on HARP CubeSat (awarded by AIAA)

Wide FOV lens



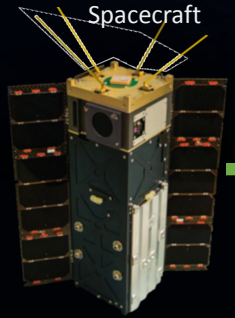
GAPMAP Specialized filter



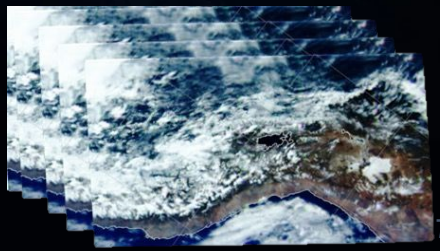
Camera Electronics

Payload computer

Spacecraft

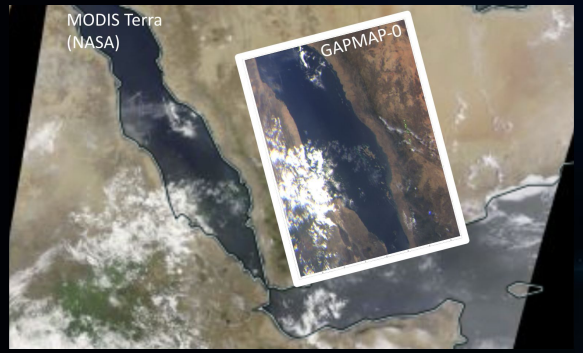


Measurements

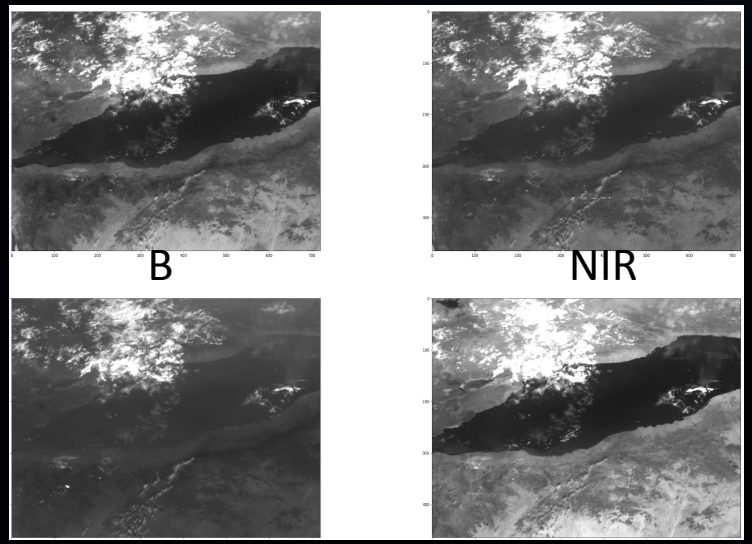


GAPMAP: First measurements from Space

RGB



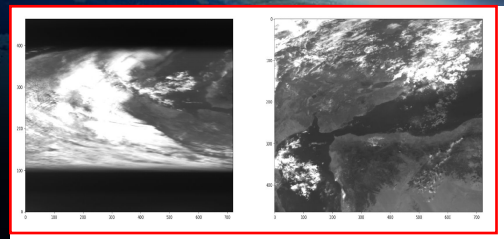
R G



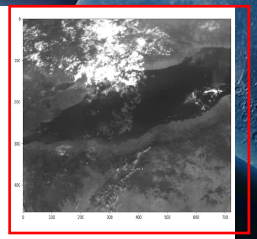
GAPMAP takes 60 measurements per pixel

= 4 wavelengths (440, 550, 670, 870nm) x 3 polarized states [I Q U] x 5 viewing-angles

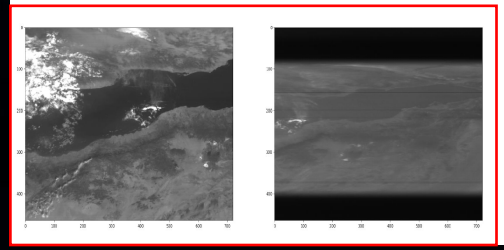
Forward view

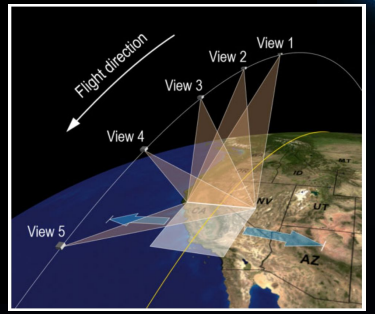


NADIR view



Backward view



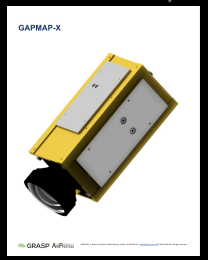


Liu et al., 2017

Déploiement de la constellation

2022 2023 2024 2025 2026 2027 2028 2029

Développements Pré-lancement → Proof-of-concept GAPMAP-0 → Développement de la chaîne de services d'Observation de la Terre → Constellation complète Service opérationnel



GAPMAP-0

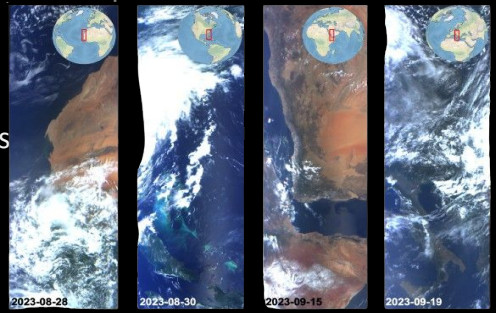
Operated by Spire mission ADLER-2

GAPMAP-1

Operated by U-Space

GAPMAP-n

Planned but funds not secured yet



3. GRASP EARTH: PRODUCT CHAIN AND CUSTOMER SEGMENTS

Hardware manufacturing

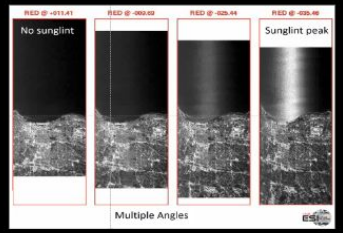
- Satellite sensors
- Ground, airborne, in situ instruments

Data acquisition

- Launching satellites
- Field and airborne campaigns
- Ground Networks

LEVEL 1

- Calibrated, geo-located data products
- Value: Engineering value
- User: R&D and institutional partners

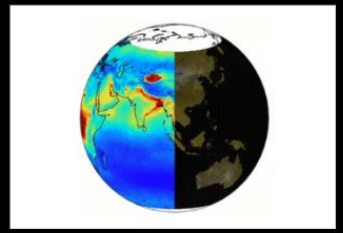


Data processing

- Data aggregation
- Algorithms
- Interpretation
- Radiative transfer

LEVEL 2&3

- Aerosol, surface, gases, clouds,.. products
- Value: Scientific value
- User: Large emitters

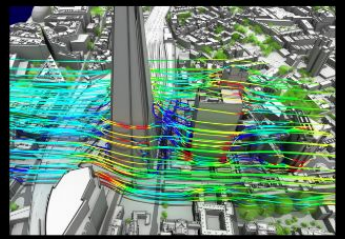


Advanced processing

- Joint databases
- Smart combinations
- Interpretation
- Assimilation to transport models

LEVEL 4

- 3D maps and forecasts of pollution, amongst others.
- Value: Broad public value
- User: Policy makers, smart cities



GRASP Software Overview

What is GRASP?

- **Generalize**
  - Based on physical concepts
  - Multi-instrument
- **Retrieval**
  - But not only! A forward model too!
- **Atmosphere**
  - Aerosol optical and microphysical properties
- **Surface**
  - BRDF
- **Properties**

GRASP code has been developed at the University of Lille by **LOA** (Laboratoire d'Optique Atmospherique)

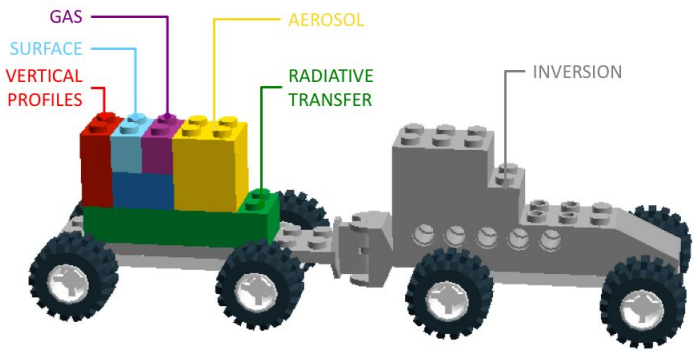
GRASP SAS:

- Promoting
- Consulting
- Maintaining
- Developing

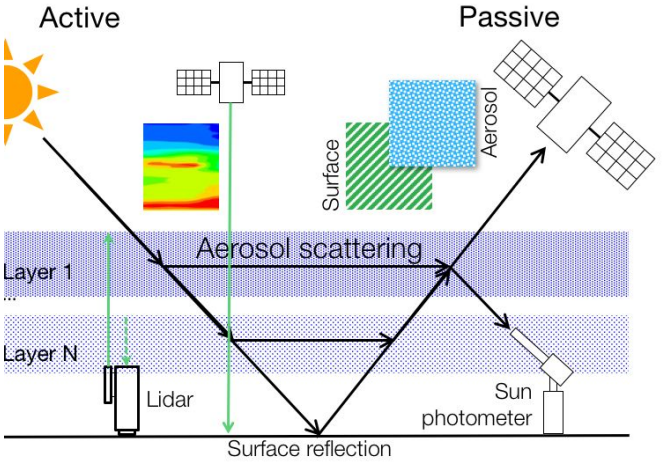
Why GRASP ?

1. **Generalized**: Based on physics concepts instead of instrument dependent concepts

2. **Modular**: The software is highly optimized and designed as independent modules.

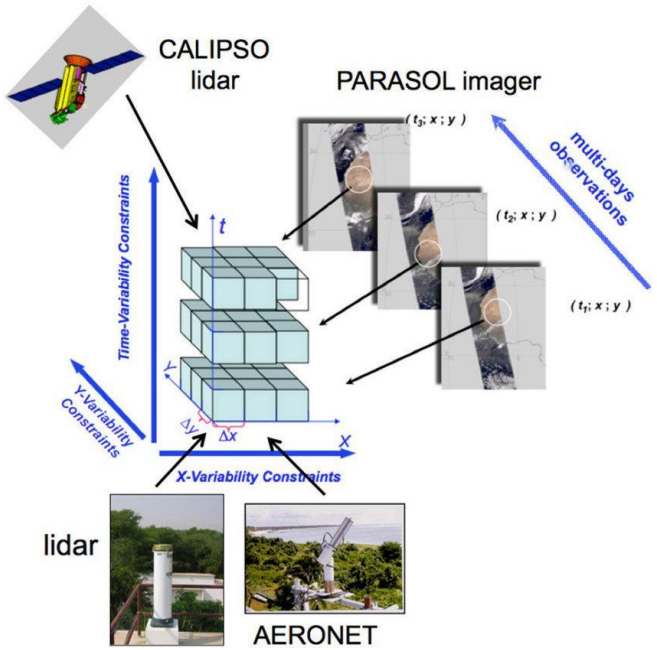


3. **Versatile**: Highly configurable, easily adaptable and extensible



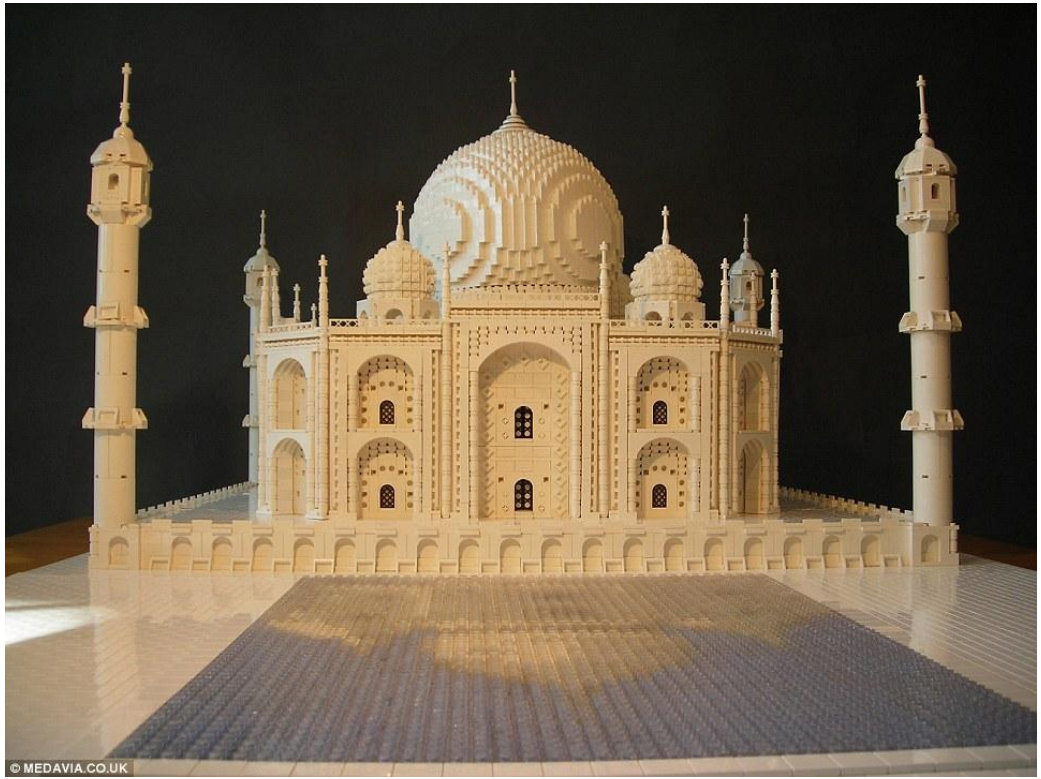
Are more reasons needed?

- Founded on solid mathematical and scientific background
- Innovative:
  - Multi-pixel constraints
  - Multi-instrument
  - Combined retrieval of atmosphere and ground
- Open Source GRASP OPEN
- **It's fun!**



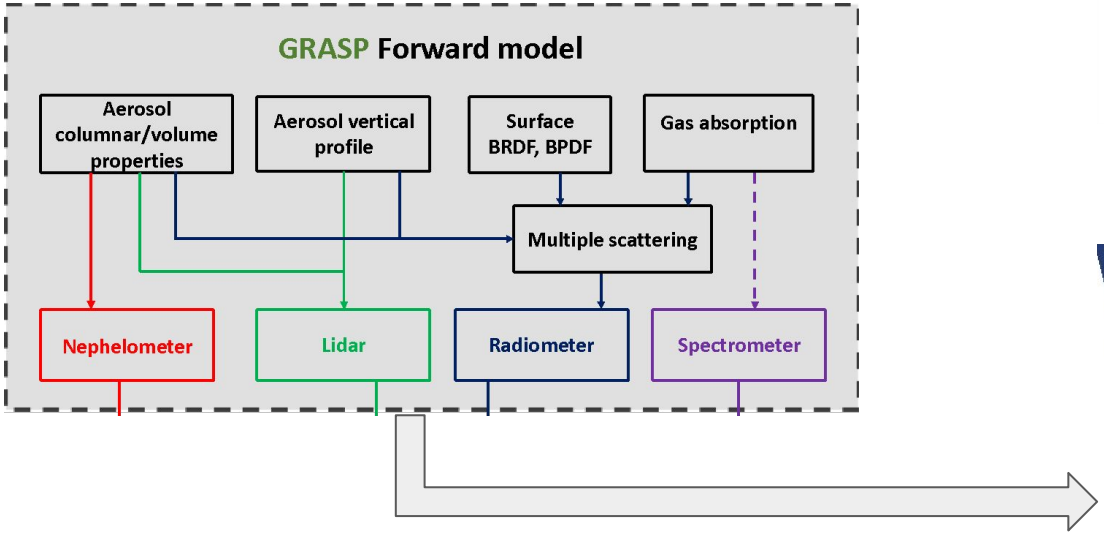
Let’s start!

Our goal!

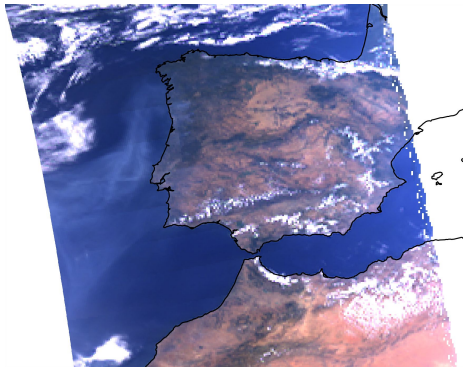


But so far... we are here:

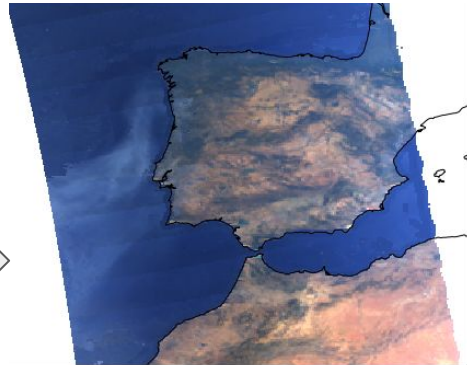
The Forward Model



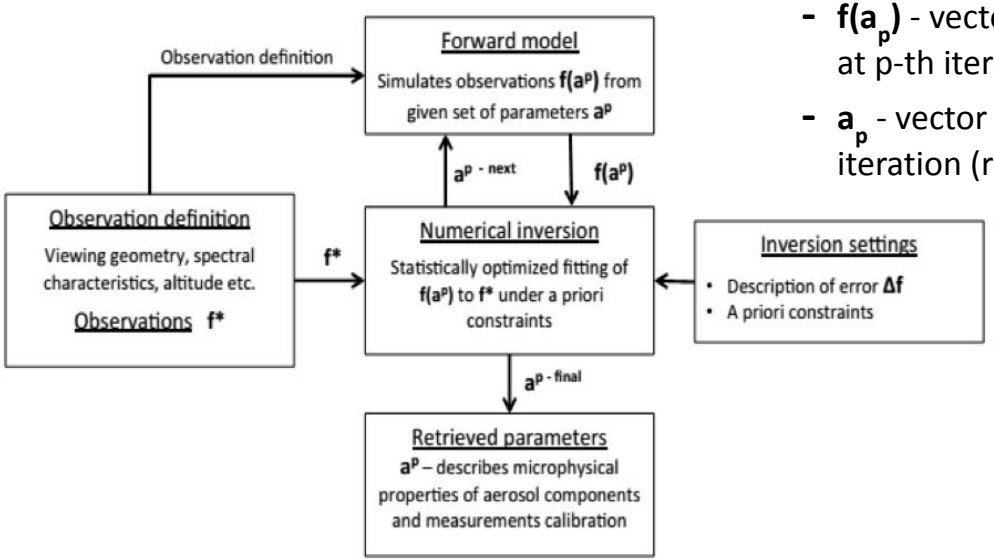
Real POLDER image
(31st Aug 2013)



Synthetic image

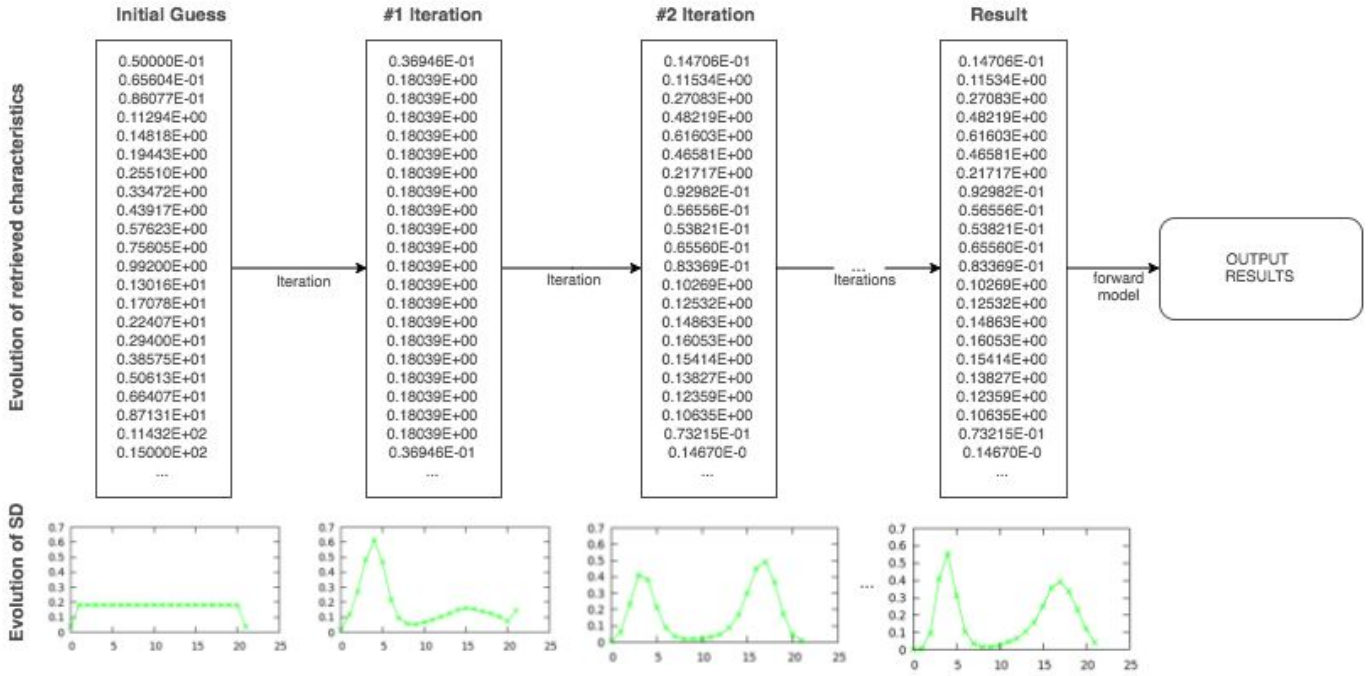


The inversion

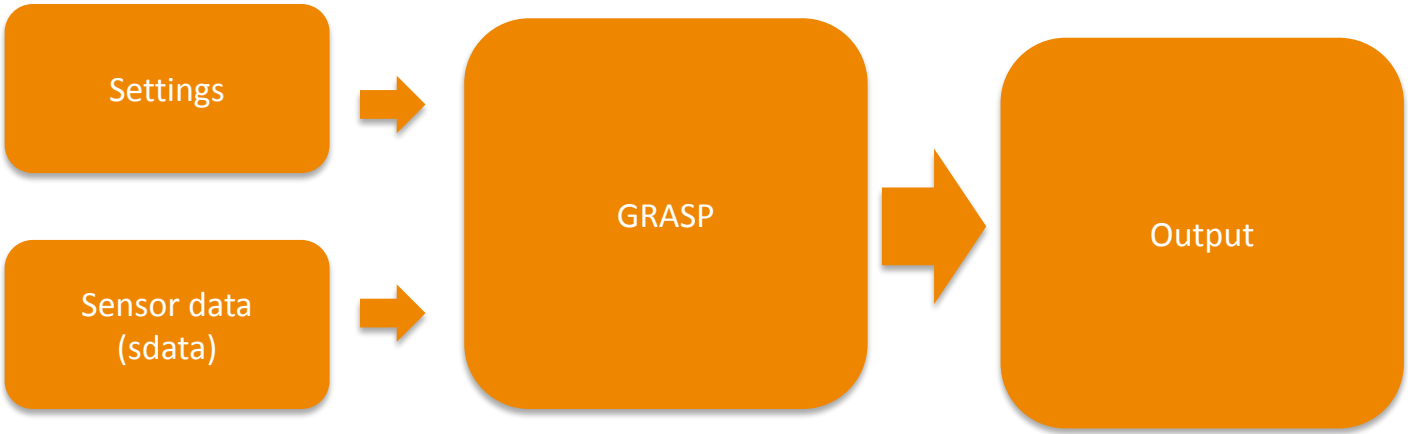


- **f*** - vector of inverted
- **$f(a_p)$** - vector of measurement fit at p-th iterations
- **a_p** - vector of unknowns at p-th iteration (retrieved parameters).

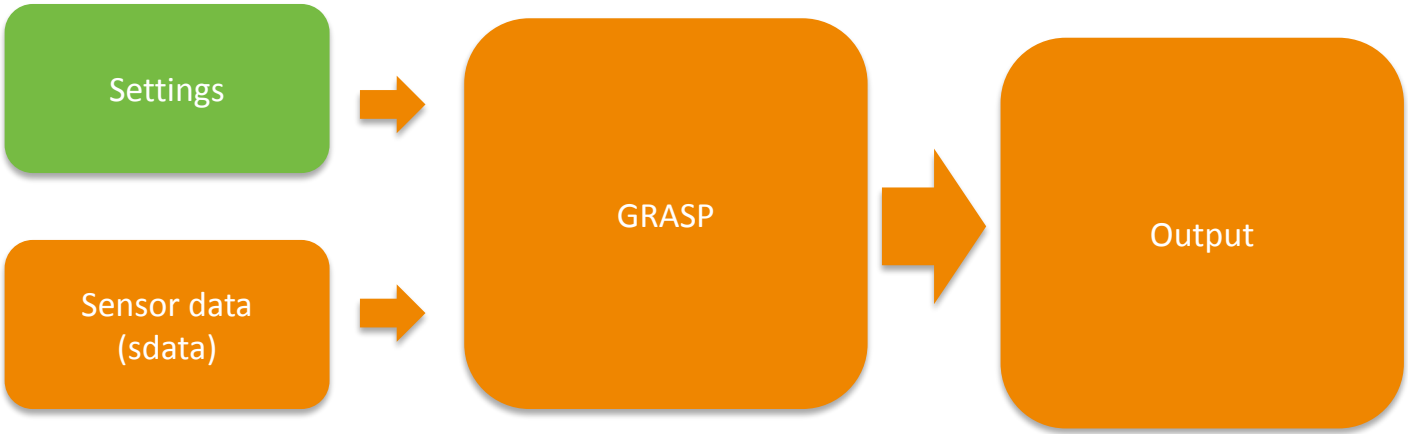
Iterative optimization



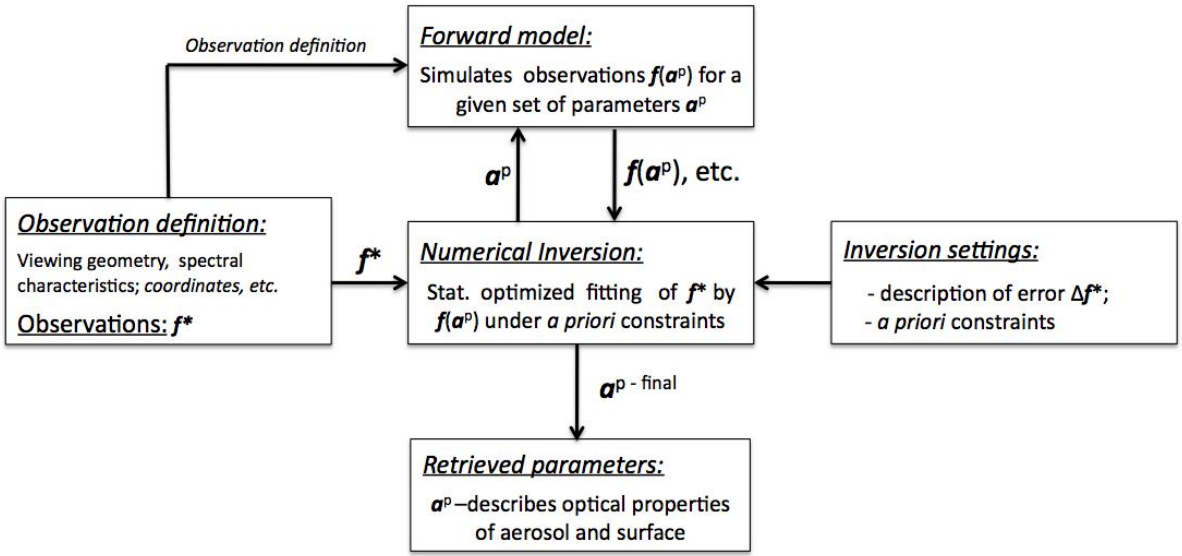
The dataflow



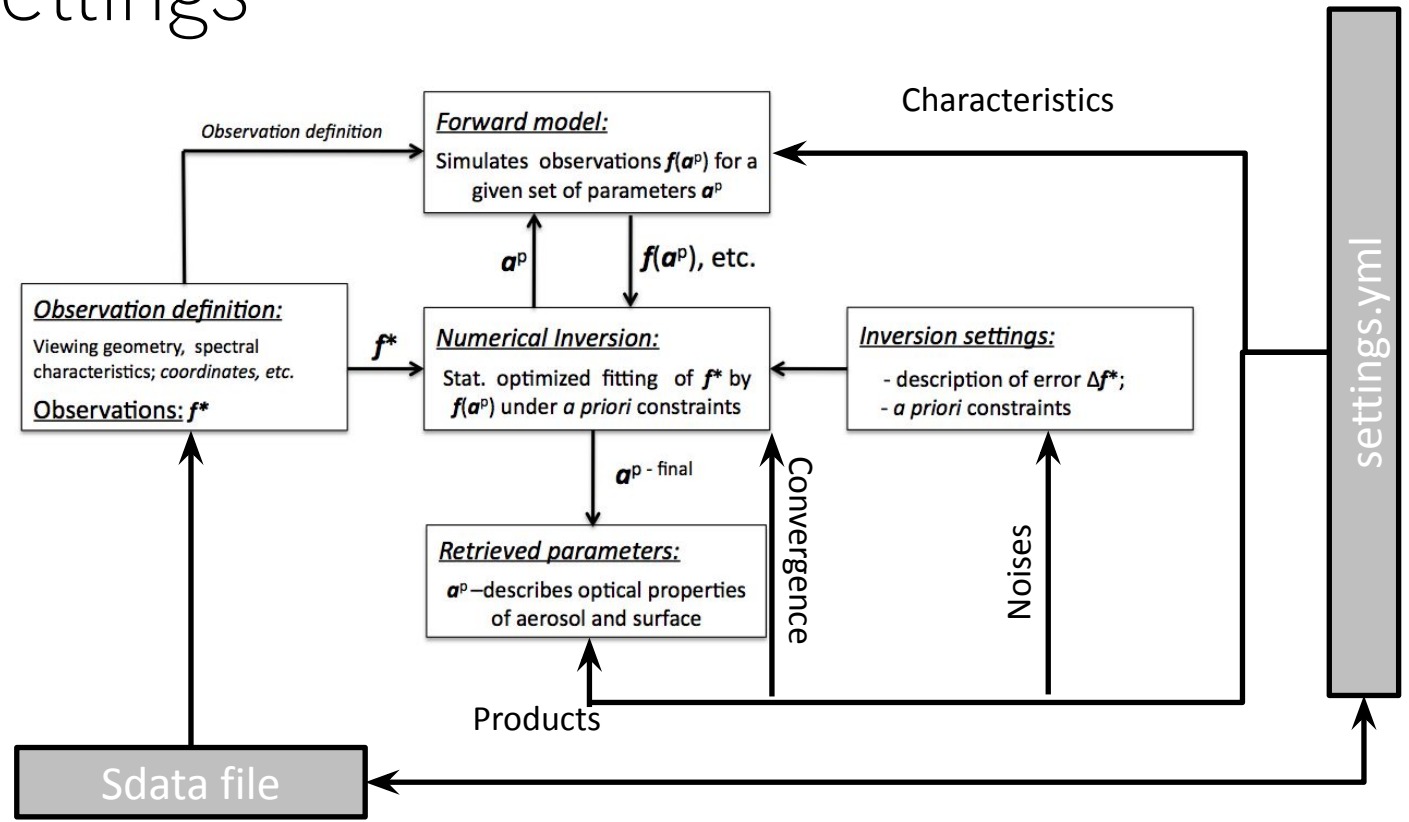
The dataflow



The settings



The settings

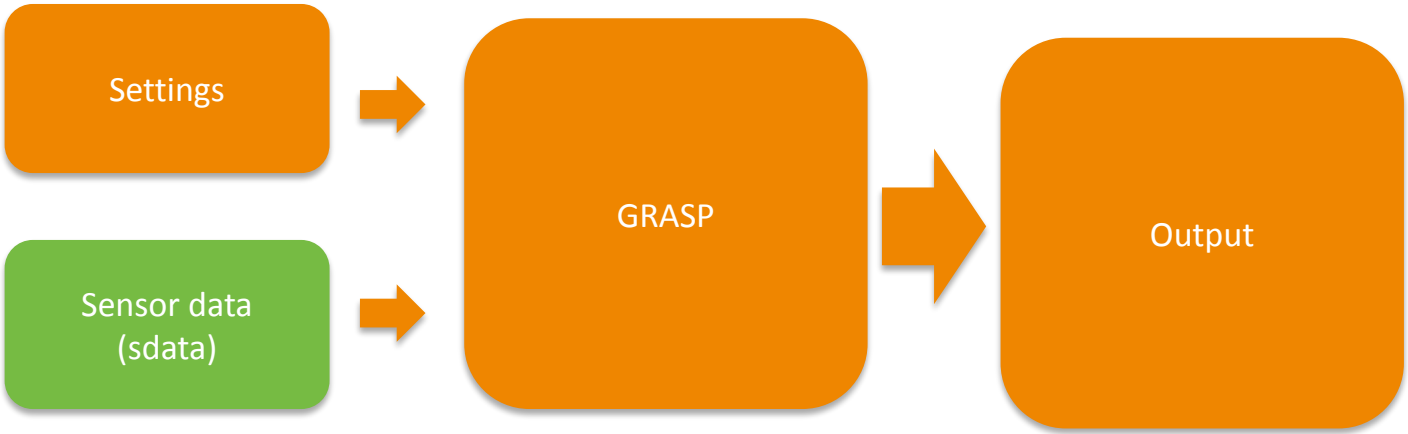


The settings

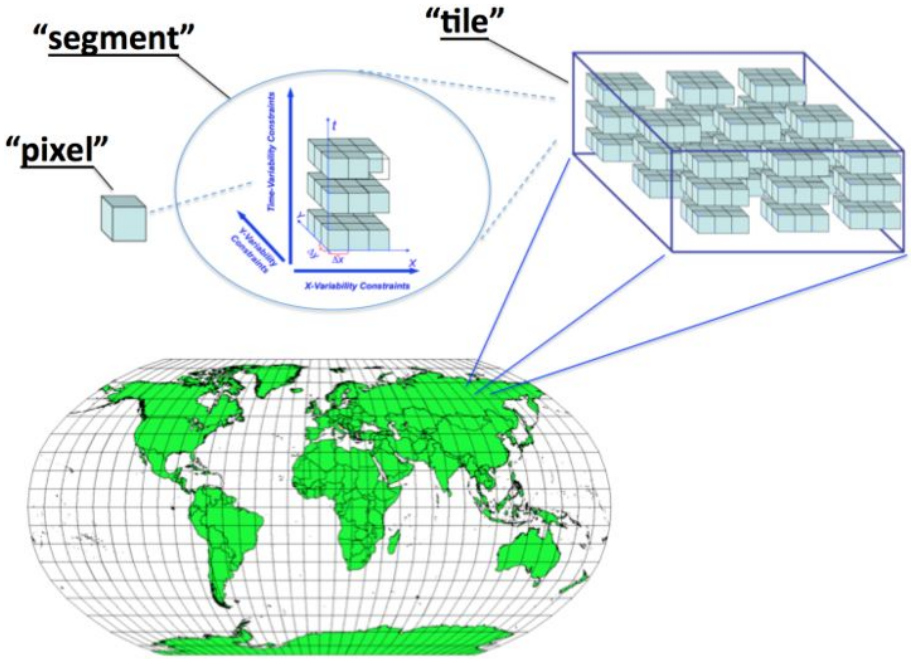
YAML file which describes the inversion strategy and its specific set up

```yaml
input:<...3 lines />

output:<...4 lines /> #example_lidar_sunphotometer_inversion_test.txt

retrieval:
    convergence:<...9 lines />

    regime_of_measurement_fitting:<...2 lines />

    product_configuration:<...2 lines />]

    regime_of_multipixel_constraints: <...2 lines />

    radiative_transfer:
        number_of_layers: 50
        molecular_profile_vertical_type: standard_atmosphere
        reference_plane_for_polarization: meridian
        simulating_observation:<...5 lines />
        simulating_derivatives:<...5 lines />

    noises:
        noise[1]:<...7 lines />]
        noise[2]:<...7 lines />]
        noise[3]:<...7 lines />]


    phase_matrix:<...11 lines />

    products:                   <...22 lines />

    debug:<...4 lines />

    constraints:
        characteristic[1]: #1
            type: size_distribution_triangle_bins
            retrieved: true
            mode[1]:
                initial_guess:     <...5 lines />]
                single_pixel:
                    smoothness_constraints:
                        difference_order: 3
                        lagrange_multiplier: 1.0e-2
                multi_pixel:
                    smoothness_constraints:
                       derivative_order_of_X_variability:    0
                       lagrange_multiplier_of_X_variability: 0.0
                       derivative_order_of_Y_variability:    0
                       lagrange_multiplier_of_Y_variability: 0.0
                       derivative_order_of_T_variability:    0
                       lagrange_multiplier_of_T_variability: 0.0
            mode[2]: <...18 lines />

        characteristic[2]: #2<...39 lines />
```

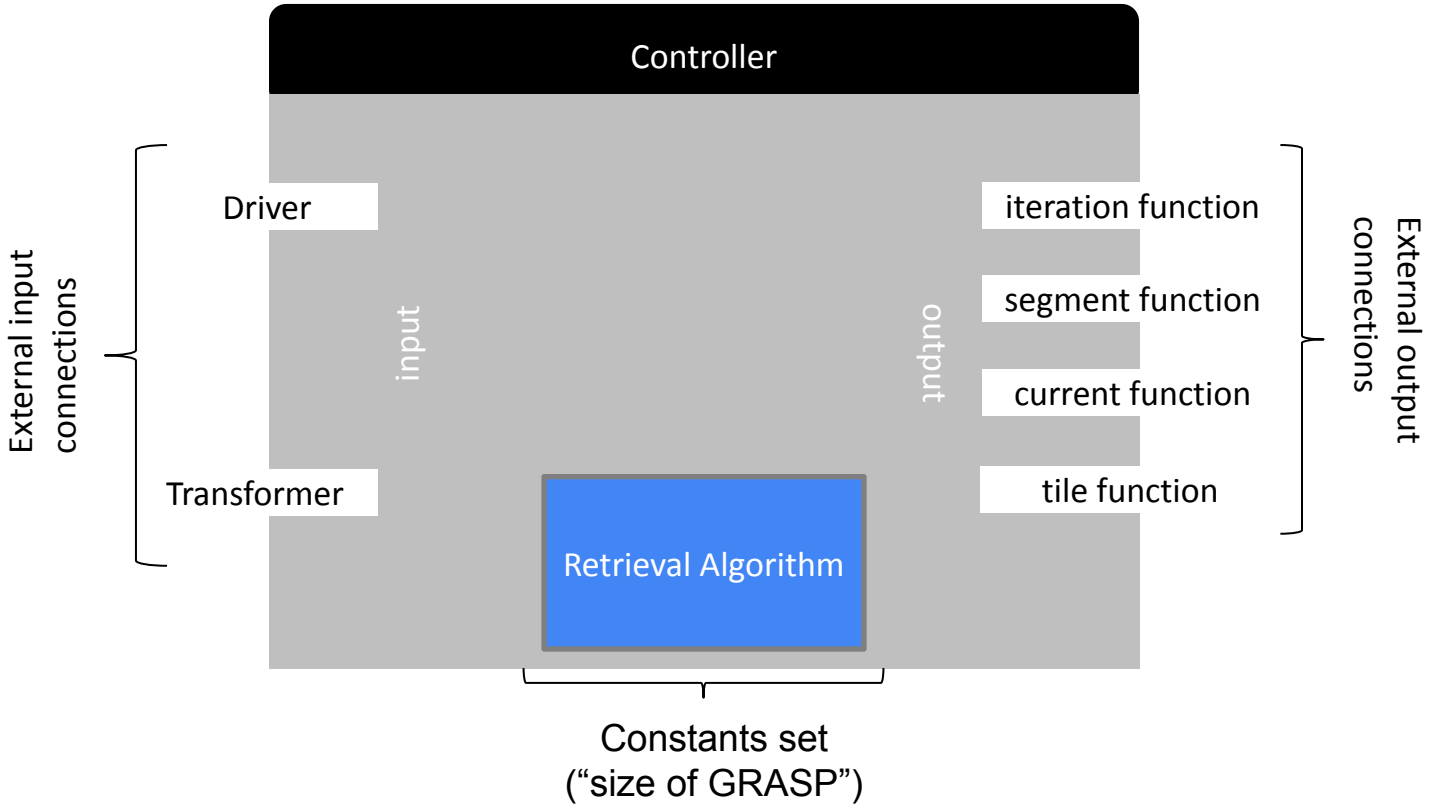
The dataflow



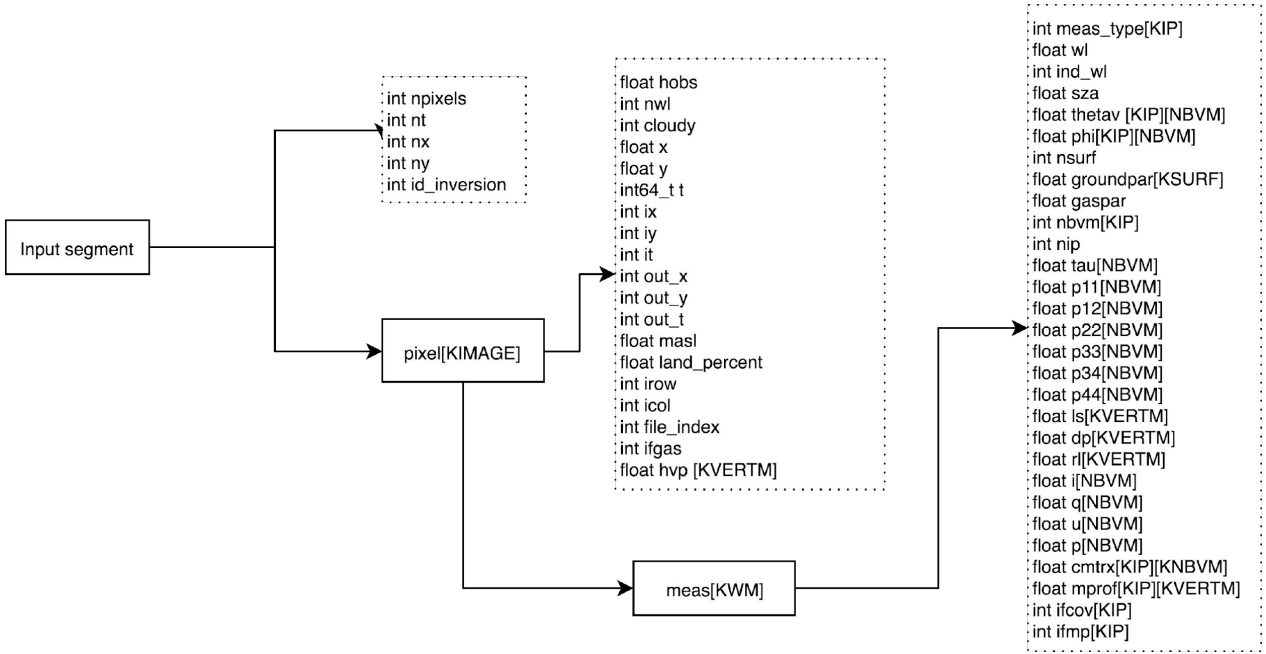
Data organization

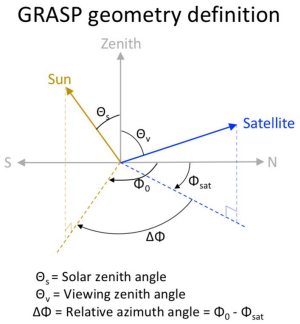


Input/output extensions



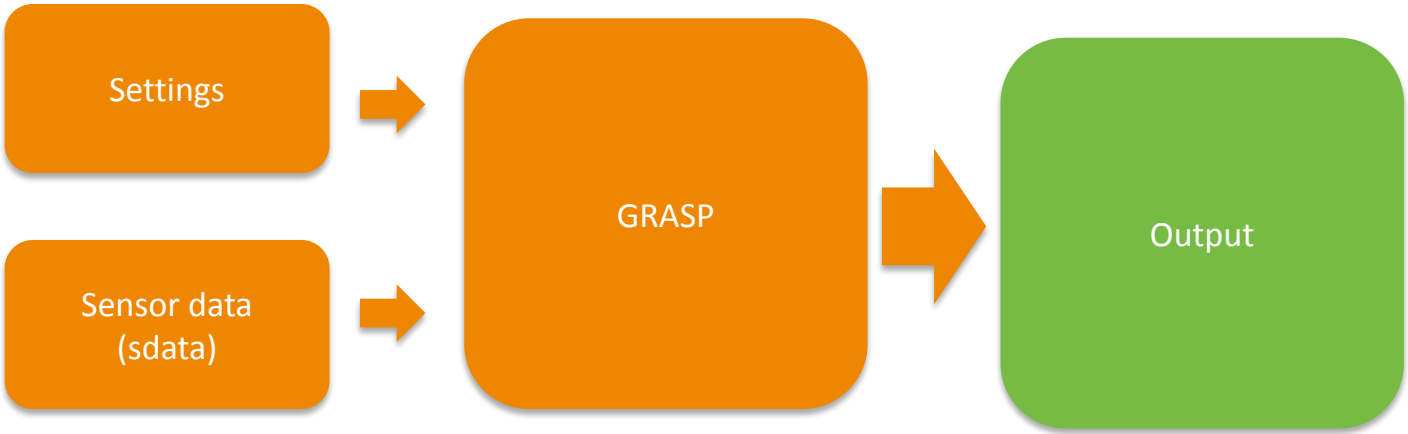
Input





SDATA description in the official documentation (chapter 4.2): https://www.grasp-open.com/doc/ch04.php

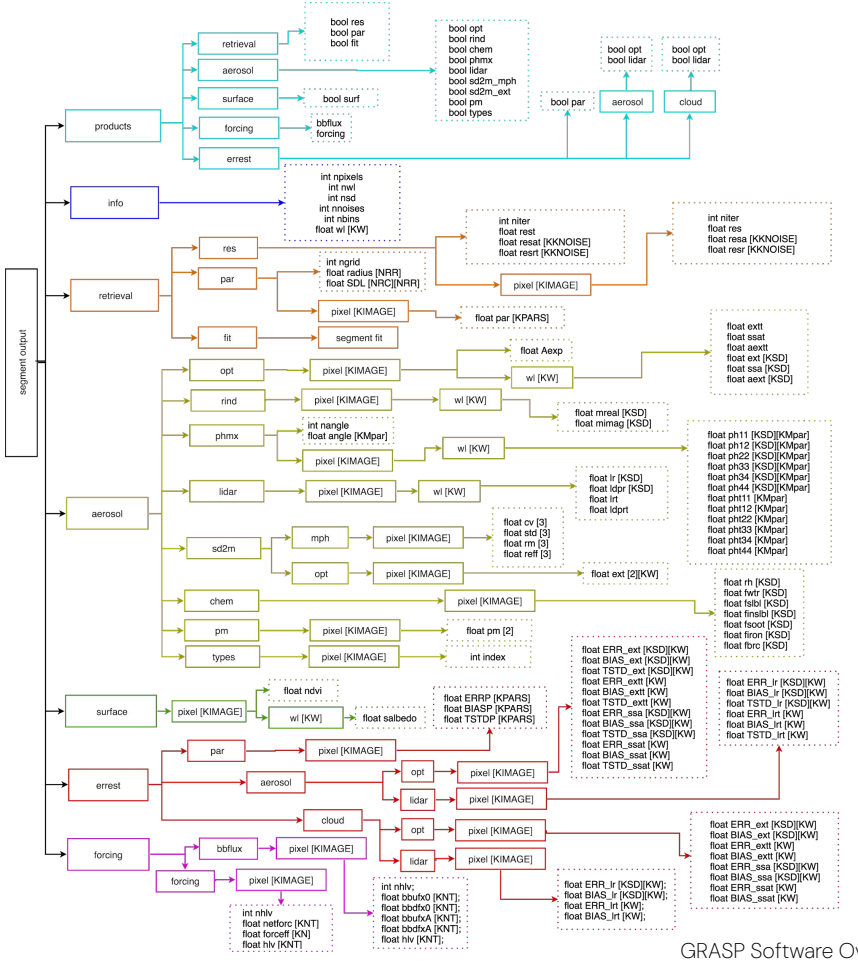
The dataflow



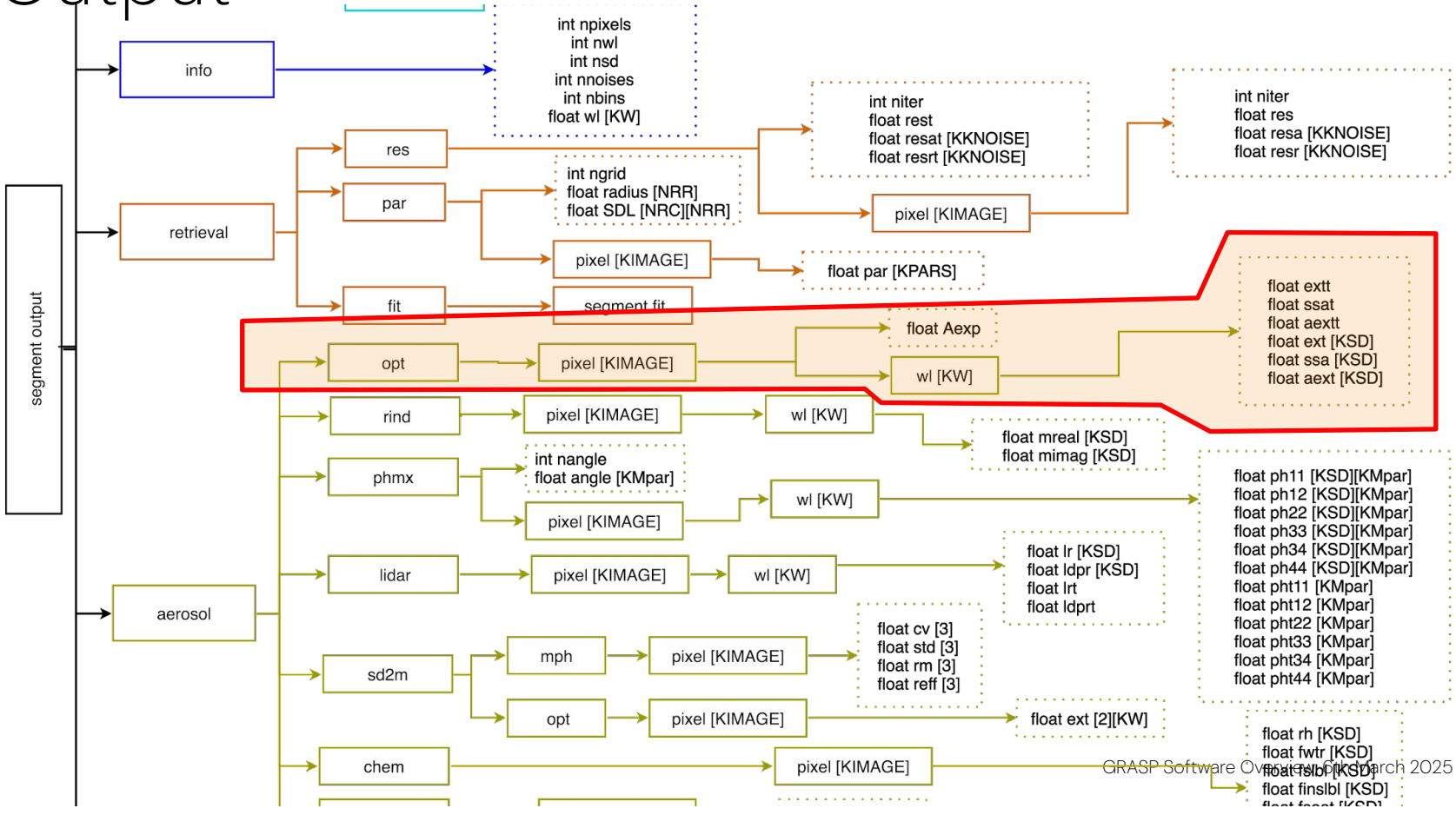
Output

It can be dumped in different formats:

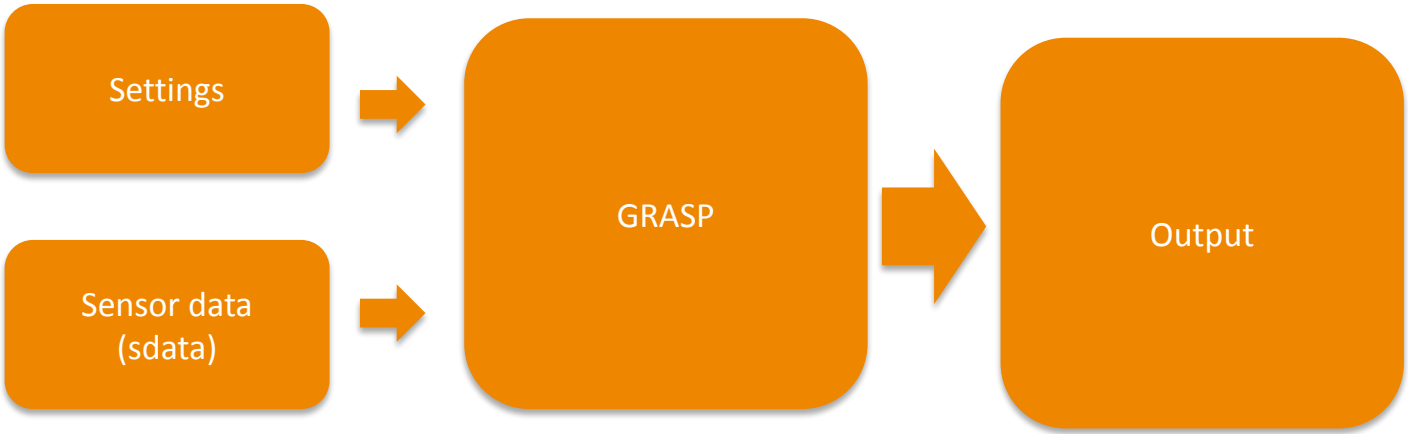
- Classic
- CSV
- …
- **python!**



Output

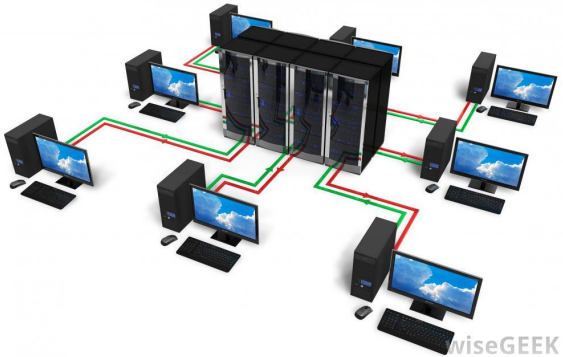

segment output
info
int npixels
int nwl
int nsd
int nnoises
int nbins
float wl [KW]
retrieval
res
par
fit
int niter
float rest
float resat [KKNOISE]
float resrt [KKNOISE]
int niter
float res
float resa [KKNOISE]
float resr [KKNOISE]
int ngrid
float radius [NRR]
float SDL [NRC][NRR]
pixel [KIMAGE]
pixel [KIMAGE]
float par [KPARS]
segment fit
float Aexp
opt
pixel [KIMAGE]
wl [KW]
float extt
float ssat
float aextt
float ext [KSD]
float ssa [KSD]
float aext [KSD]
rind
pixel [KIMAGE]
wl [KW]
float mreal [KSD]
float mimag [KSD]
phmx
int nangle
float angle [KMpar]
pixel [KIMAGE]
wl [KW]
float ph11 [KSD][KMpar]
float ph12 [KSD][KMpar]
float ph22 [KSD][KMpar]
float ph33 [KSD][KMpar]
float ph34 [KSD][KMpar]
float ph44 [KSD][KMpar]
float pht11 [KMpar]
float pht12 [KMpar]
float pht22 [KMpar]
float pht33 [KMpar]
float pht34 [KMpar]
float pht44 [KMpar]
aerosol
lidar
pixel [KIMAGE]
wl [KW]
float lr [KSD]
float ldpr [KSD]
float lrt
float ldprt
sd2m
mph
pixel [KIMAGE]
float cv [3]
float std [3]
float rm [3]
float reff [3]
opt
pixel [KIMAGE]
float ext [2][KW]
chem
pixel [KIMAGE]
float rh [KSD]
float fwtr [KSD]
float fslb [KSD]
float finslbl [KSD]

The dataflow: Let’s run it!



GRASP-Cloud

GRASP CLOUD

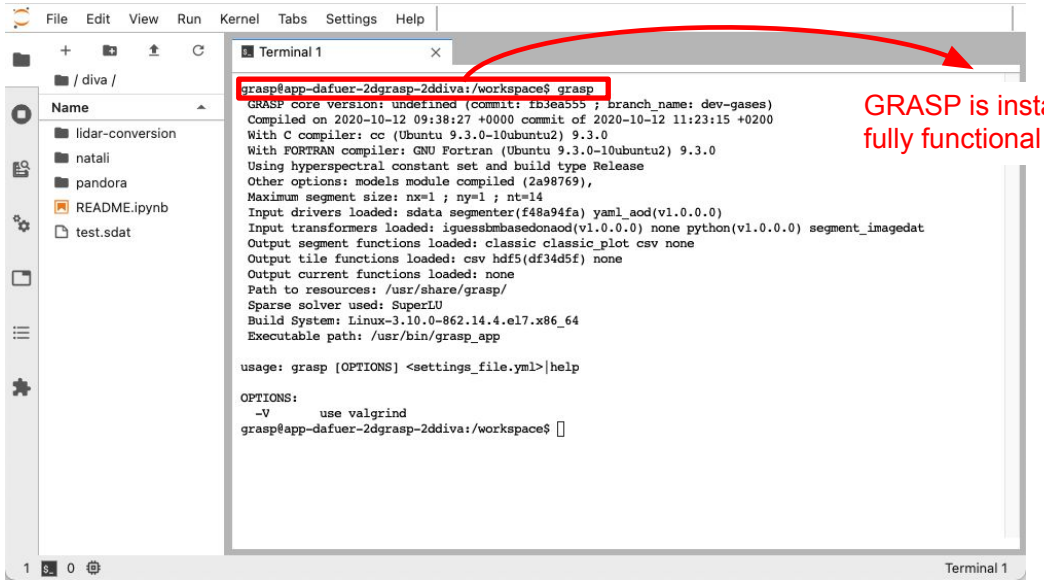


GRASP IS IN THE ~~AIR~~ CLOUD

GRASP-Cloud

Standalone Operation

1. Download GRASP-Open Code
2. Install Compilers and Libraries
3. Compile GRASP
4. Prepare Input Data (e.g. SDATA)
5. Prepare Configuration
6. Run GRASP — **Where focus should be !**
7. Load and Interpret Output



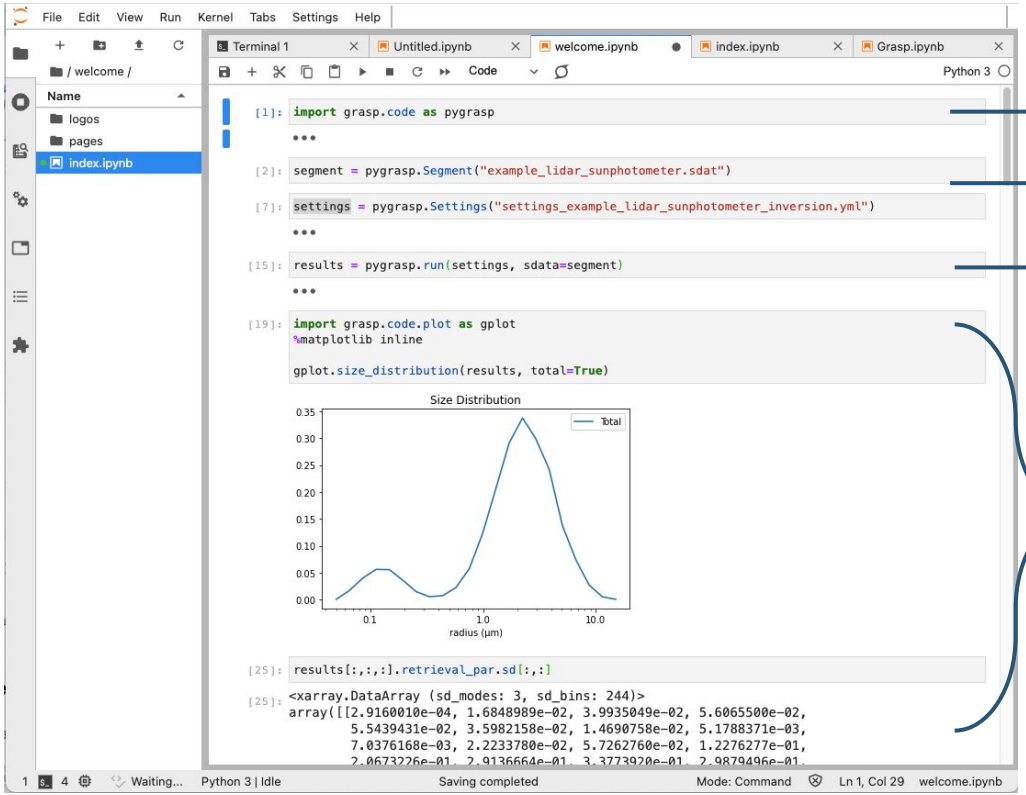
How to access to the platform (prototype):

1. Register at GRASP-Open website
2. Access: https://hub.grasp-cloud.com/diva

How to run GRASP in the terminal:

$> grasp settings.yml

GRASP-Cloud



Import the algorithm

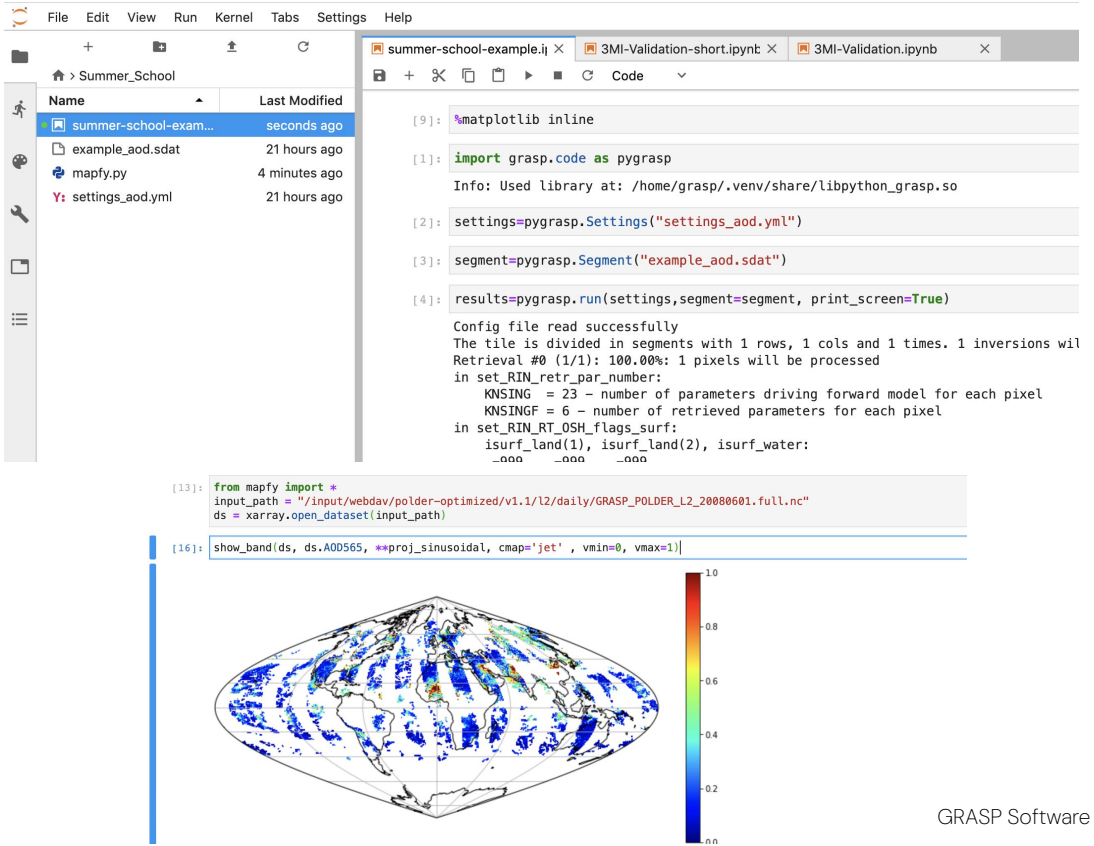
Prepare the input

Execute de algorithm

Check the results:

- Visualization
- Access to data

GRASP-Cloud


File Edit View Run Kernel Tabs Settings Help
Summer_School
Name
Last Modified
summer-school-exam...
seconds ago
example_aod.sdat
21 hours ago
mapfy.py
4 minutes ago
settings_aod.yml
21 hours ago
summer-school-example.ip
3MI-Validation-short.ipynb
3MI-Validation.ipynb
Code
[9]: %matplotlib inline
[1]: import grasp.code as pygrasp
Info: Used library at: /home/grasp/.venv/share/libpython_grasp.so
[2]: settings=pygrasp.Settings("settings_aod.yml")
[3]: segment=pygrasp.Segment("example_aod.sdat")
[4]: results=pygrasp.run(settings,segment=segment, print_screen=True)
Config file read successfully
The tile is divided in segments with 1 rows, 1 cols and 1 times. 1 inversions wil
Retrieval #0 (1/1): 100.00%: 1 pixels will be processed
in set_RIN_retr_par_number:
KNSING = 23 – number of parameters driving forward model for each pixel
KNSINGF = 6 – number of retrieved parameters for each pixel
in set_RIN_RT_OSH_flags_surf:
isurf_land(1), isurf_land(2), isurf_water:
[13]: from mapfy import *
input_path = "/input/webdav/polder-optimized/v1.1/l2/daily/GRASP_POLDER_L2_20080601.full.nc"
ds = xarray.open_dataset(input_path)
[16]: show_band(ds, ds.AOD565, **proj_sinusoidal, cmap='jet' , vmin=0, vmax=1)
1.0
0.8
0.6
0.4
0.2
0.0

Before finishing....

Resources at GRASP Open Web Page

Access to the code

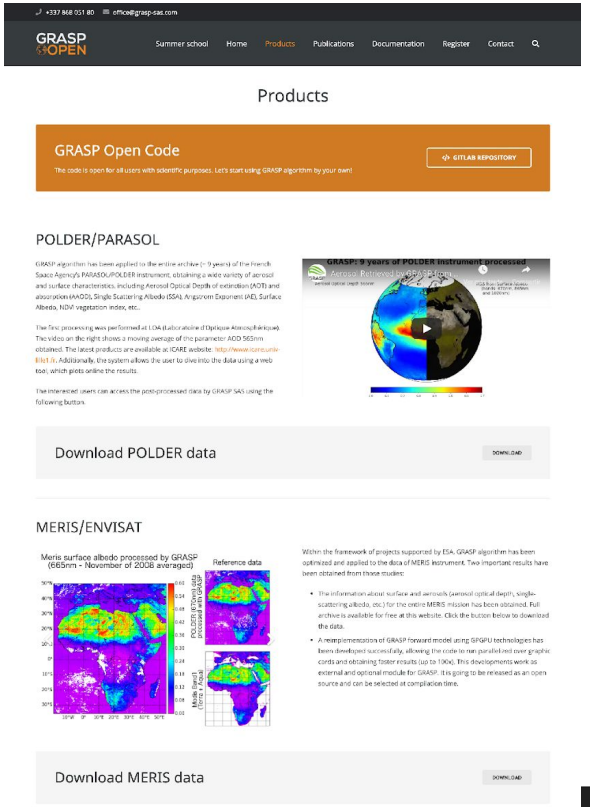
Data processed ready for:

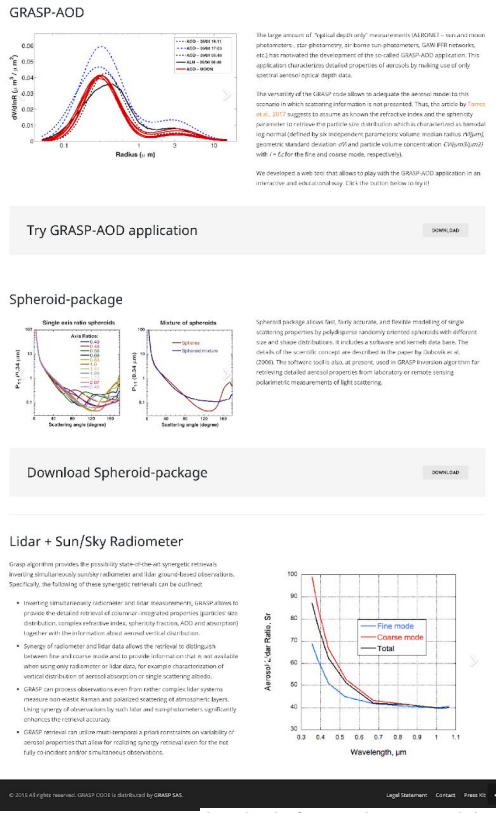
- Easy download
- Visualization

Documentation

Publications

Tools





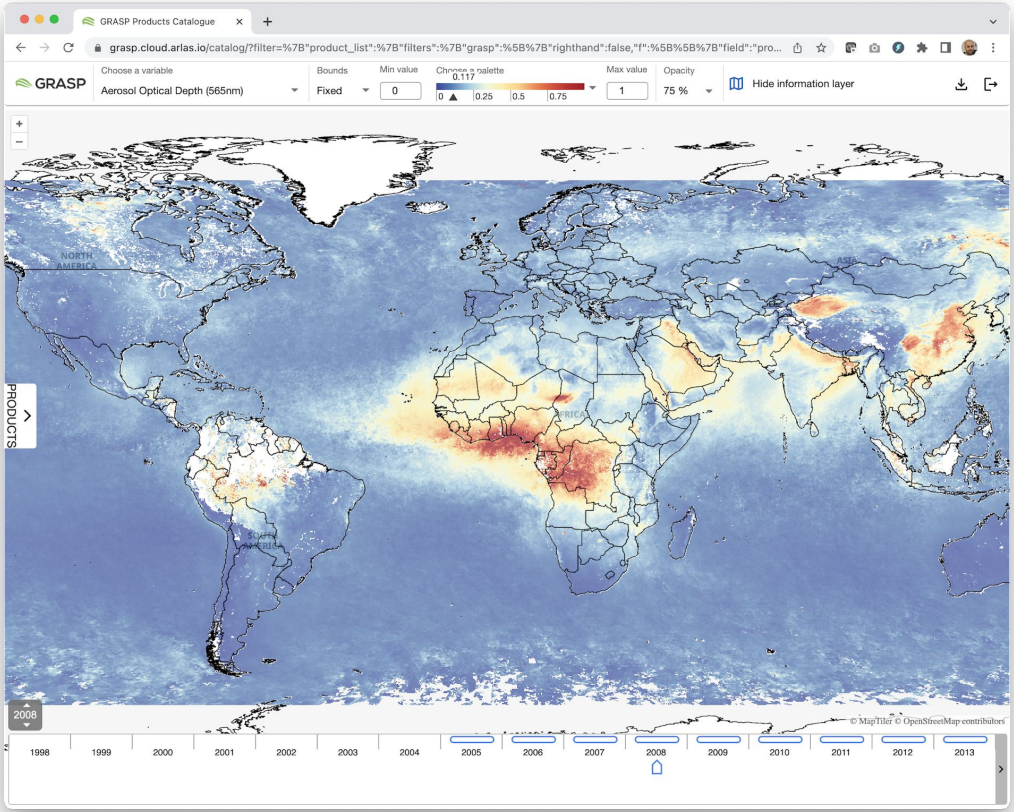
Resources at GRASP Open Web Page

POLDER 1,2 and 3:
- Components
- Models

Sentinel-3/OLCI

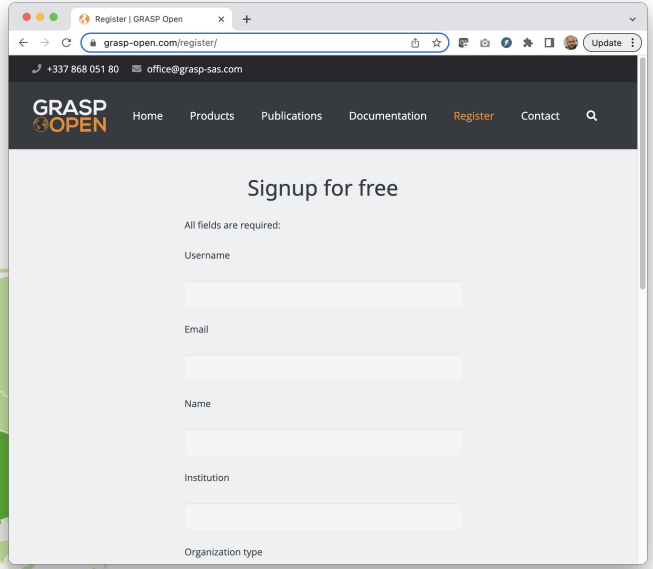
Sentinel-5p/Tropomi

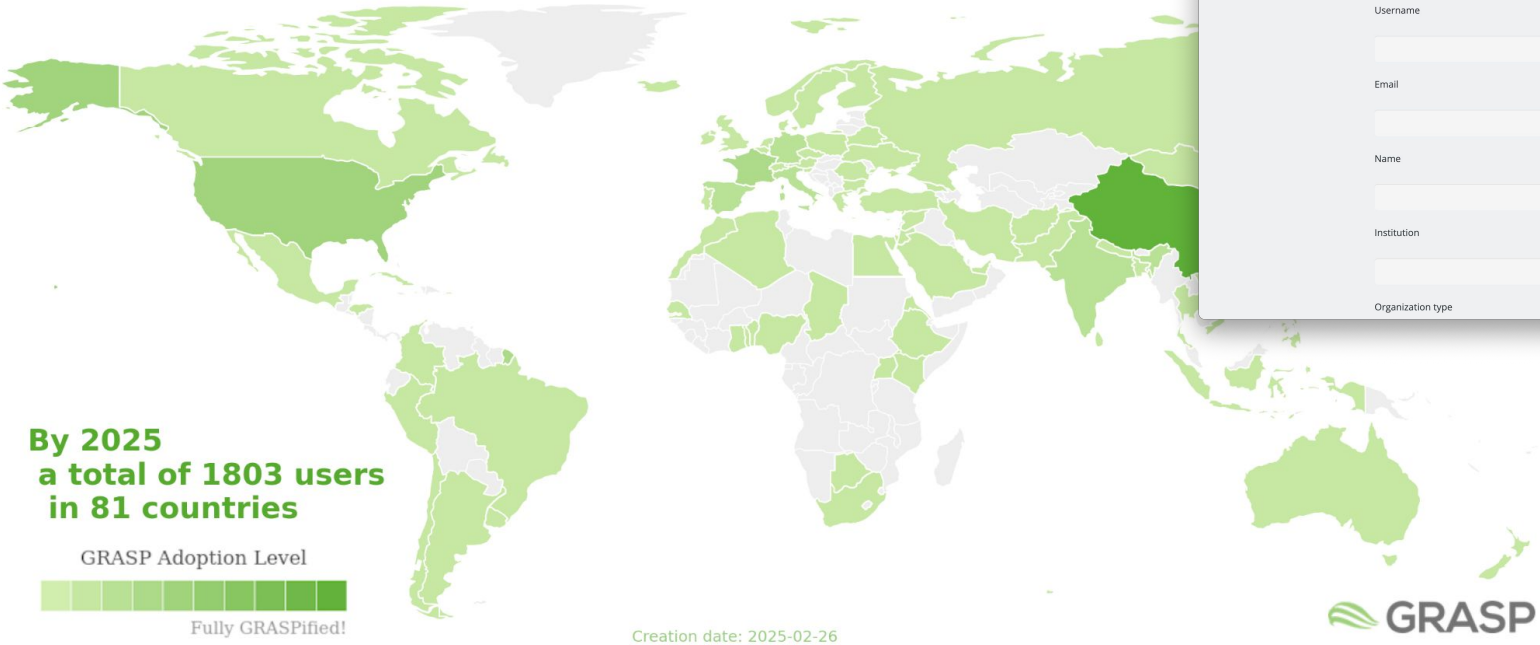
Envisant:
- Meris
- AATSR
- Synergy Meris + AATSR



Registration

https://www.grasp-open.com/register





GRASP

Thank you very much...

... and good luck with the training!

Funded by
the European Union