Mission Analysis in Python

A “how-to” crash course

Arnaud MULLER

Table of Contents

Introduction

I. Python Workflow

II. Tools Used & Recommended

III. Installation & Testing

IV. Collaborative Coding

Useful Links

Introduction

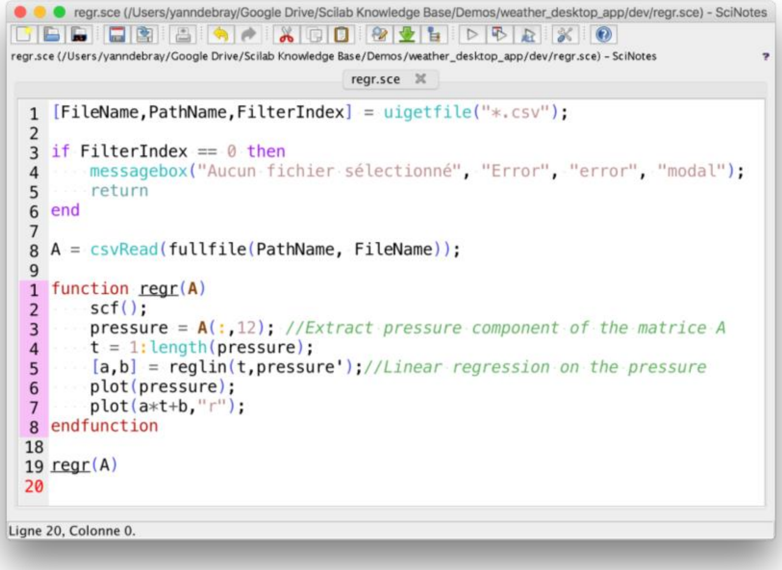
Some historical context

Celest Lab

The Scilab Space Mechanics Toolbox

cnes
Space for Earth

Scilab


regr.sce (/Users/yanndebray/Google Drive/Scilab Knowledge Base/Demos/weather_desktop_app/dev/regr.sce) - SciNotes
regr.sce (/Users/yanndebray/Google Drive/Scilab Knowledge Base/Demos/weather_desktop_app/dev/regr.sce) - SciNotes
regr.sce

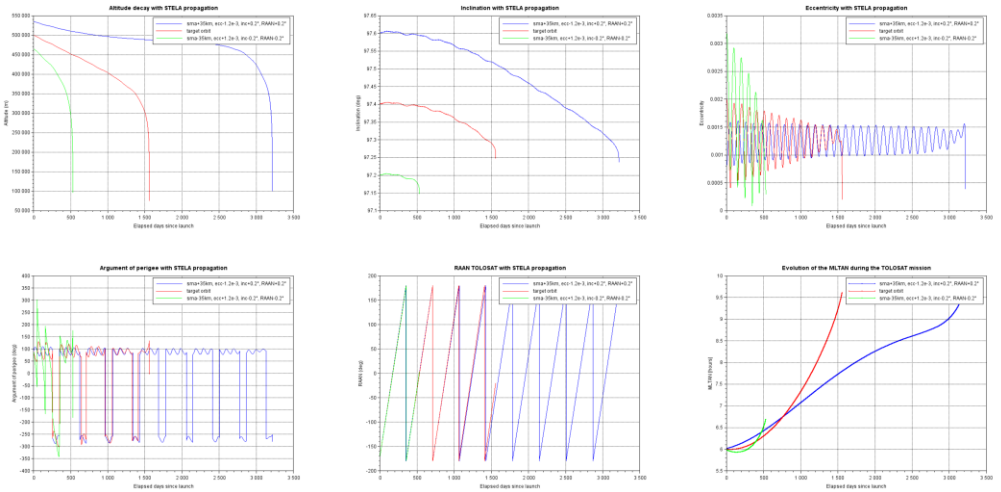
```
[FileName,PathName,FilterIndex] = uigetfile("*.csv");

if FilterIndex == 0 then
    messagebox("Aucun fichier sélectionné", "Error", "error", "modal");
    return
end

A = csvRead(fullfile(PathName, FileName));

function regr(A)
    scf();
    pressure = A(:,12); //Extract pressure component of the matrice A
    t = 1:length(pressure);
    [a,b] = reglin(t,pressure');//Linear regression on the pressure
    plot(pressure);
    plot(a*t+b,"r");
endfunction

regr(A)
```

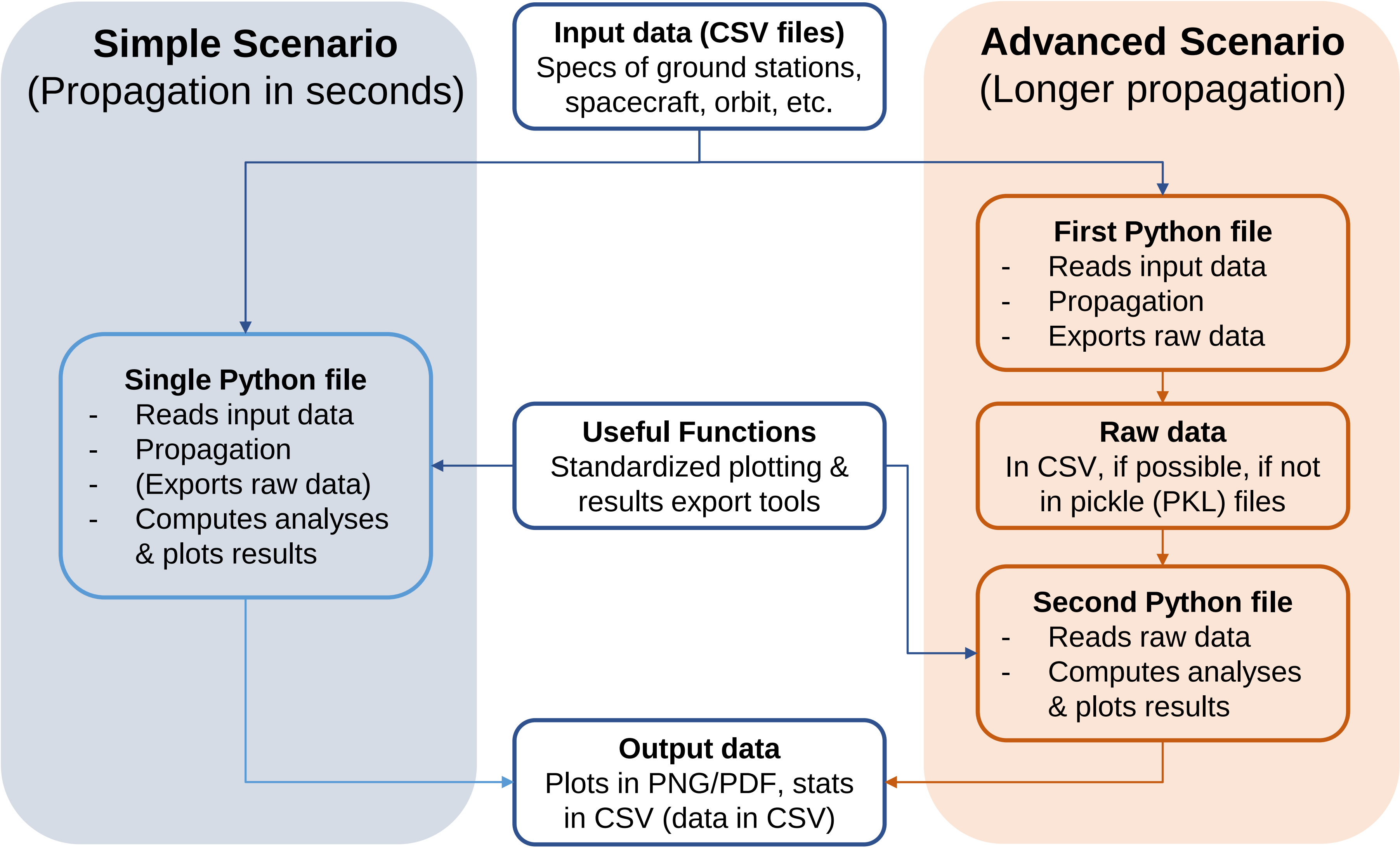


Introduction

Why move to Python ?

- More modern and flexible language
- Easy to learn and use, still allows for high-performance applications through interfaces
- Scilab very limiting and outdated (CNES also looking into moving to Python)
- Opens access to many free and open-source libraries available
- Free choice of IDE and development environment, can wield the power of plugins, etc.
- More modular code, easier to collaborate with other subsystems when needed

Python Workflow



Simple Scenario
(Propagation in seconds)

Input data (CSV files)
Specs of ground stations, spacecraft, orbit, etc.

Advanced Scenario
(Longer propagation)

First Python file
- Reads input data
- Propagation
- Exports raw data

Single Python file
- Reads input data
- Propagation
- (Exports raw data)
- Computes analyses & plots results

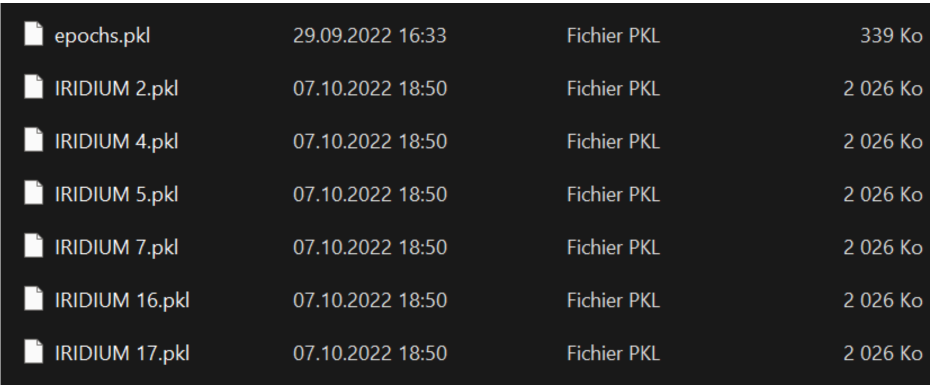
Useful Functions
Standardized plotting & results export tools

Raw data
In CSV, if possible, if not in pickle (PKL) files

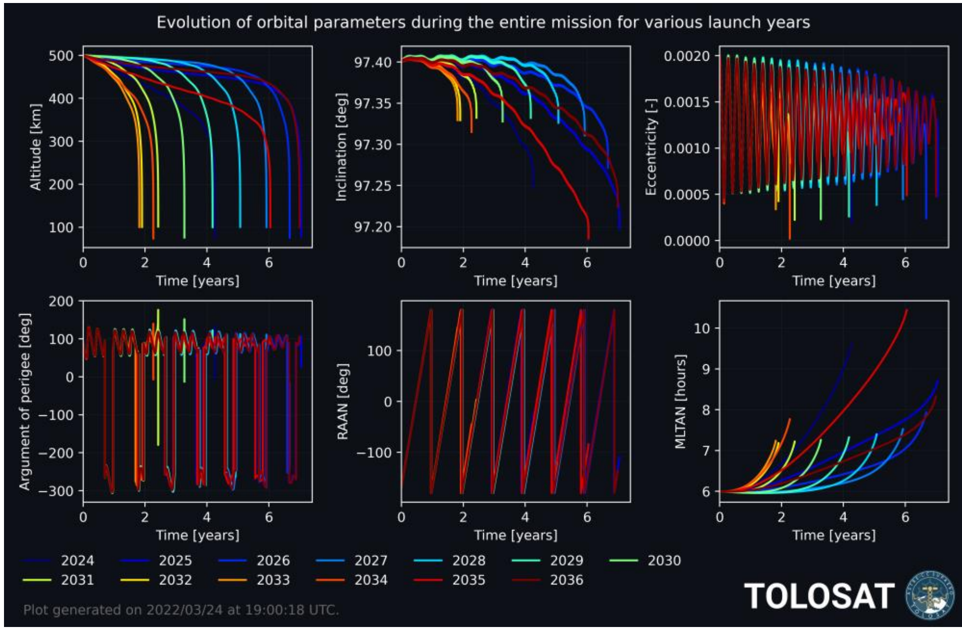
Second Python file
- Reads raw data
- Computes analyses & plots results

Output data
Plots in PNG/PDF, stats in CSV (data in CSV)

Python Workflow

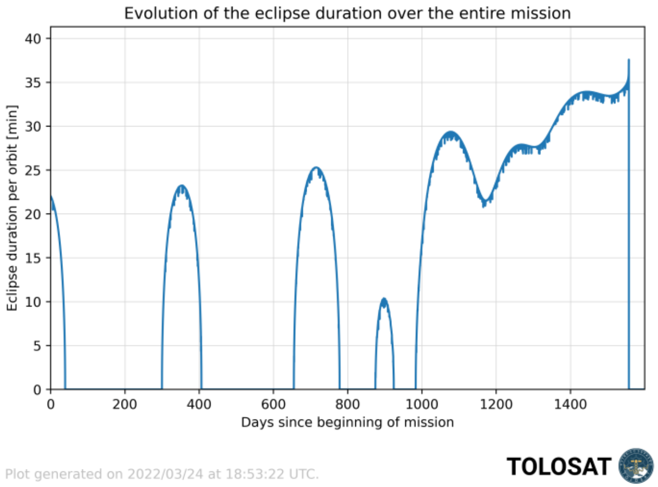


epochs.pkl	29.09.2022 16:33	Fichier PKL	339 Ko
IRIDIUM 2.pkl	07.10.2022 18:50	Fichier PKL	2 026 Ko
IRIDIUM 4.pkl	07.10.2022 18:50	Fichier PKL	2 026 Ko
IRIDIUM 5.pkl	07.10.2022 18:50	Fichier PKL	2 026 Ko
IRIDIUM 7.pkl	07.10.2022 18:50	Fichier PKL	2 026 Ko
IRIDIUM 16.pkl	07.10.2022 18:50	Fichier PKL	2 026 Ko
IRIDIUM 17.pkl	07.10.2022 18:50	Fichier PKL	2 026 Ko



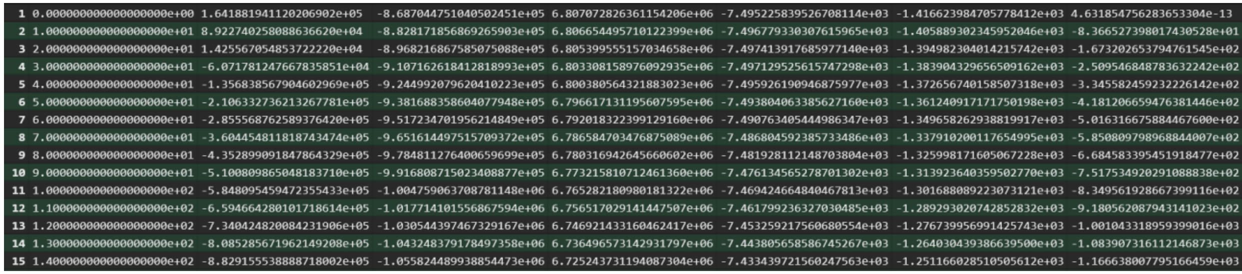
Evolution of orbital parameters during the entire mission for various launch years

Plot generated on 2022/03/24 at 19:00:18 UTC.



Evolution of the eclipse duration over the entire mission

Plot generated on 2022/03/24 at 18:53:22 UTC.



```
0.000000000000000000e+00 1.641881941120206902e+05  -8.687044751040502451e+05 6.807072826361154206e+06 -7.495225839526708114e+03 -1.416623984705778412e+03 4.631854756283653304e-13
1.000000000000000000e+01 8.922740258088636620e+04  -8.828171856869265903e+05 6.806654495710122399e+06 -7.496779330307615965e+03 -1.405889302345952046e+03 -8.366527398017430528e+01
2.000000000000000000e+01 1.425567054853722220e+04  -8.968216867585075088e+05 6.805399555157034658e+06 -7.497413917685977140e+03 -1.394982304014215742e+03 -1.673202653794761545e+02
3.000000000000000000e+01 -6.071781247667835851e+04 -9.107162618412818993e+05 6.803308158976092935e+06 -7.497129525615747298e+03 -1.383904329656509162e+03 -2.509546848783632242e+02
4.000000000000000000e+01 -1.356838567904602969e+05 -9.244992079620410223e+05 6.800380564321883023e+06 -7.495926190946875977e+03 -1.372656740158507318e+03 -3.345582459232226142e+02
5.000000000000000000e+01 -2.106332736213267781e+05 -9.381688358604077948e+05 6.796617131195607595e+06 -7.493804063385627160e+03 -1.361240917171750198e+03 -4.181206659476381446e+02
6.000000000000000000e+01 -2.855568762589376420e+05 -9.517234701956214849e+05 6.792018322399129160e+06 -7.490763405444986347e+03 -1.349658262938819917e+03 -5.016316675884467600e+02
7.000000000000000000e+01 -3.604454811818743474e+05 -9.651614497515709372e+05 6.786584703476875089e+06 -7.486804592385733486e+03 -1.337910200117654995e+03 -5.850809798968844007e+02
8.000000000000000000e+01 -4.352899091847864329e+05 -9.784811276400659699e+05 6.780316942645660602e+06 -7.481928112148703804e+03 -1.325998171605067228e+03 -6.684583395451918477e+02
9.000000000000000000e+01 -5.100809865048183710e+05 -9.916808715023408877e+05 6.773215810712461360e+06 -7.476134565278701302e+03 -1.313923640359502770e+03 -7.517534920291088838e+02
1.000000000000000000e+02 -5.848095459472355433e+05 -1.004759063708781148e+06 6.765282180980181322e+06 -7.469424664840467813e+03 -1.301688089223073121e+03 -8.349561928667399116e+02
1.100000000000000000e+02 -6.594664280101718614e+05 -1.017714101556867594e+06 6.756517029141447507e+06 -7.461799236327030485e+03 -1.289293020742852832e+03 -9.180562087943141023e+02
1.200000000000000000e+02 -7.340424820084231906e+05 -1.030544397467329167e+06 6.746921433160462417e+06 -7.453259217560680554e+03 -1.276739956991425743e+03 -1.001043318959399016e+03
1.300000000000000000e+02 -8.085285671962149208e+05 -1.043248379178497358e+06 6.736496573142931797e+06 -7.443805658586745267e+03 -1.264030439386639500e+03 -1.083907316112146873e+03
1.400000000000000000e+02 -8.829155538888718002e+05 -1.055824489938854473e+06 6.725243731194087304e+06 -7.433439721560247563e+03 -1.251166028510505612e+03 -1.166638007795166459e+03
```

Tools Used & Recommended

Used

CONDA

tudat
ad astra per aspera

GitHub

Recommended

PC PyCharm

GitHub Desktop

GitHub Copilot

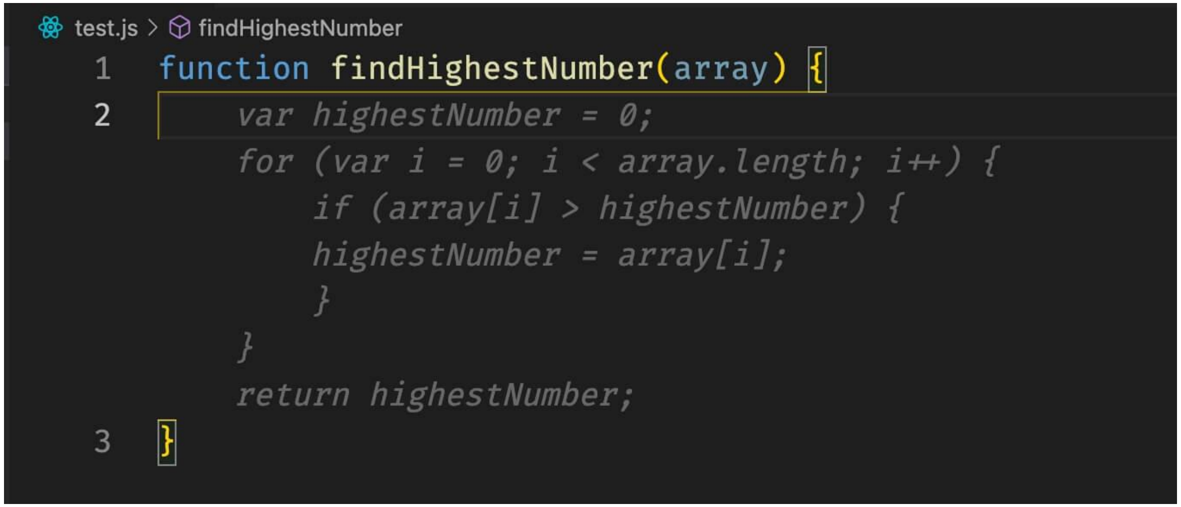
Tools Used & Recommended

Important notes

- **CONDA** → You can use **ANACONDA**, or **Miniforge** for a lighter install
- **PyCharm** → Free professional licenses are awarded to students (details in the Python setup guide)
- **GitHub Copilot** → Available for free for students through the GitHub Global Campus program

"you can't just have an AI write your code for you"
Thats where you're wrong kiddo
GitHub Copilot

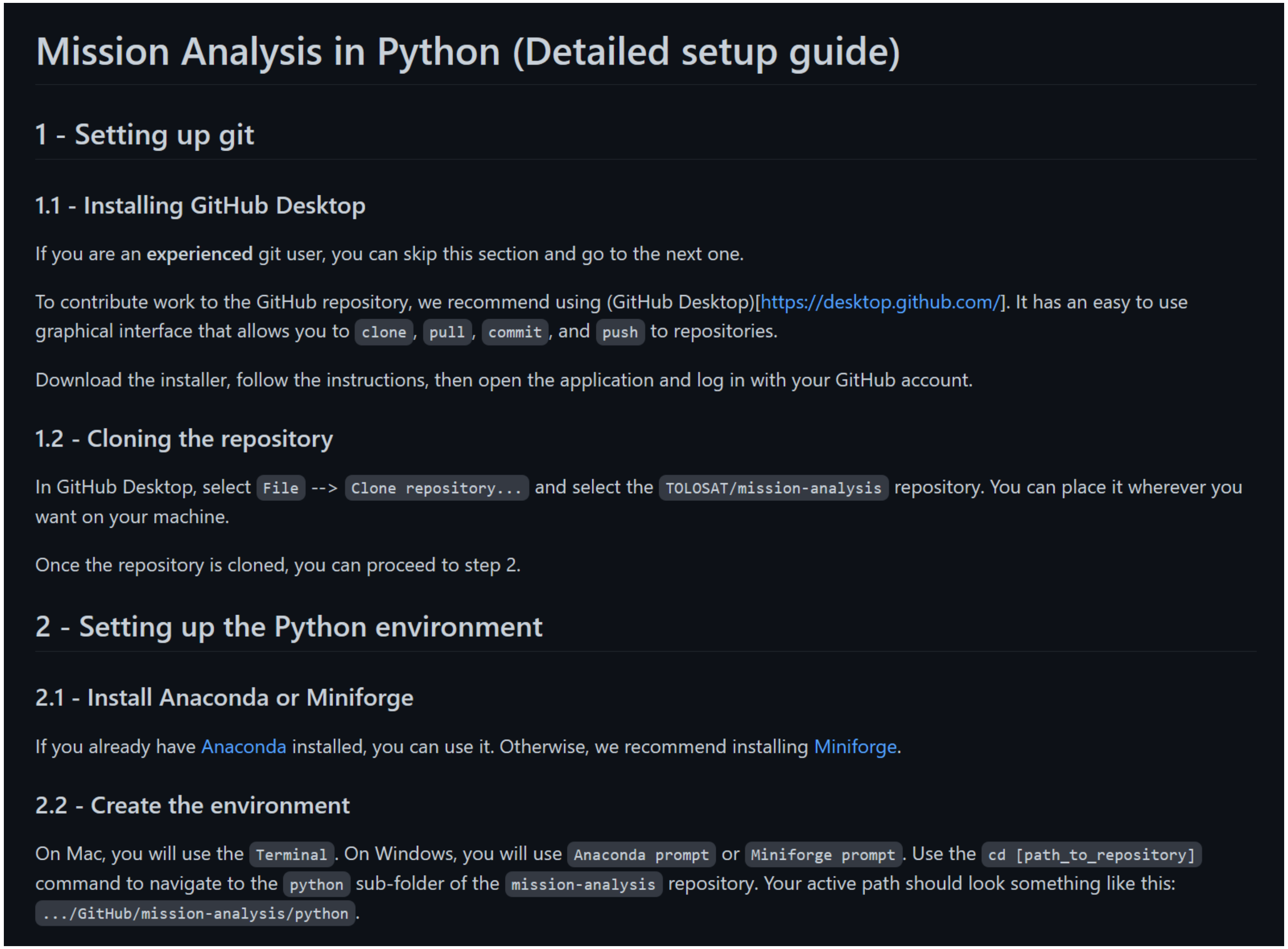
```
// okay copilot invert a binary tree
function invertTree(tree: any): any {
  if (tree.left) {
    invertTree(tree.left);
  }
  if (tree.right) {
    invertTree(tree.right);
  }
  tree.left = tree.right;
  tree.right = tree.left;
}
```



```
function findHighestNumber(array) {
    var highestNumber = 0;
    for (var i = 0; i < array.length; i++) {
        if (array[i] > highestNumber) {
        highestNumber = array[i];
        }
    }
    return highestNumber;
}
```

Installation & Testing

Follow the setup guide on GitHub → here ←



Mission Analysis in Python (Detailed setup guide)

1 - Setting up git

1.1 - Installing GitHub Desktop

If you are an **experienced** git user, you can skip this section and go to the next one.

To contribute work to the GitHub repository, we recommend using (GitHub Desktop)[https://desktop.github.com/]. It has an easy to use graphical interface that allows you to `clone`, `pull`, `commit`, and `push` to repositories.

Download the installer, follow the instructions, then open the application and log in with your GitHub account.

1.2 - Cloning the repository

In GitHub Desktop, select `File` --> `Clone repository...` and select the `TOLOSAT/mission-analysis` repository. You can place it wherever you want on your machine.

Once the repository is cloned, you can proceed to step 2.

2 - Setting up the Python environment

2.1 - Install Anaconda or Miniforge

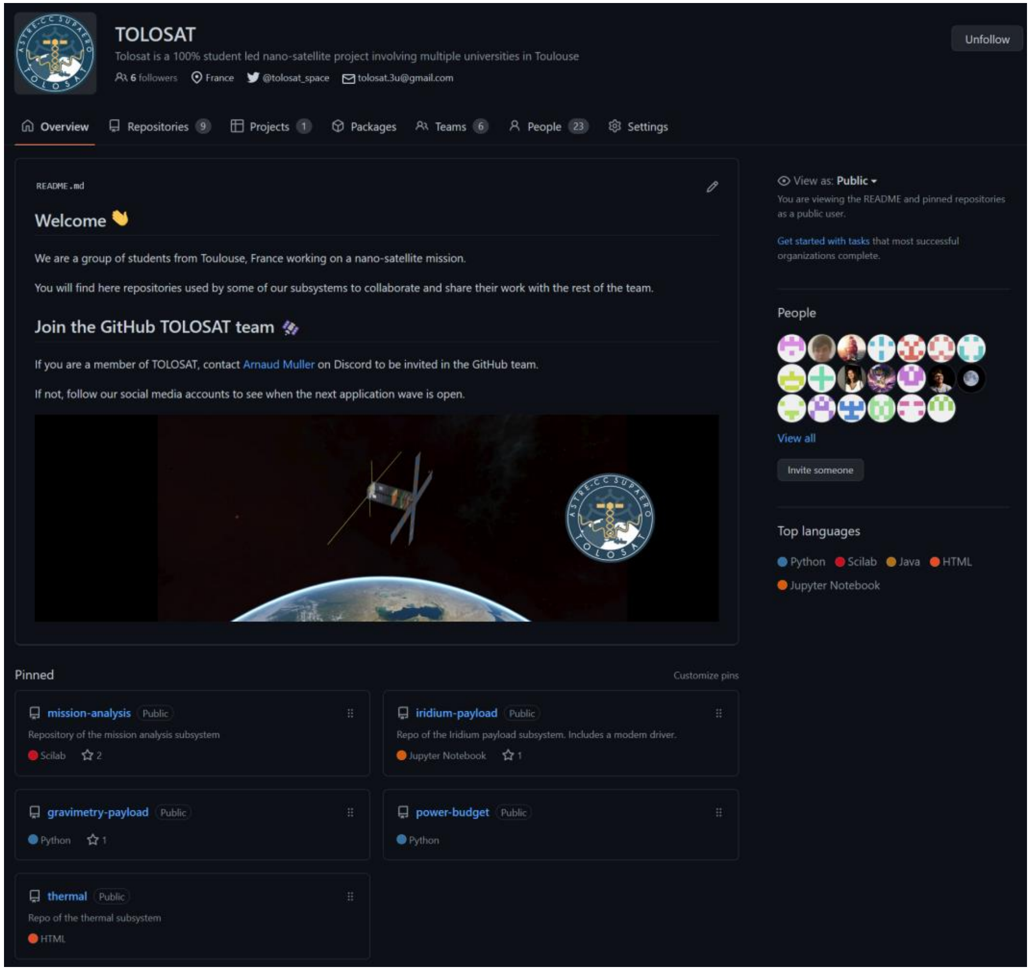
If you already have Anaconda installed, you can use it. Otherwise, we recommend installing Miniforge.

2.2 - Create the environment

On Mac, you will use the `Terminal`. On Windows, you will use `Anaconda prompt` or `Miniforge prompt`. Use the `cd [path_to_repository]` command to navigate to the `python` sub-folder of the `mission-analysis` repository. Your active path should look something like this: `.../GitHub/mission-analysis/python`.

Collaborative Coding

To join the TOLOSAT GitHub organization → give your username/email to the person linked on the page



TOLOSAT

Tolosat is a 100% student led nano-satellite project involving multiple universities in Toulouse

6 followers · France · @tolosat_space · tolosat.3u@gmail.com

README.md

Welcome 👋

We are a group of students from Toulouse, France working on a nano-satellite mission.

You will find here repositories used by some of our subsystems to collaborate and share their work with the rest of the team.

Join the GitHub TOLOSAT team 🛰

If you are a member of TOLOSAT, contact Arnaud Muller on Discord to be invited in the GitHub team.

If not, follow our social media accounts to see when the next application wave is open.

View as: Public

You are viewing the README and pinned repositories as a public user.

Get started with tasks that most successful organizations complete.

Top languages

● Python ● Scilab ● Java ● HTML ● Jupyter Notebook

Pinned

mission-analysis · Public
Repository of the mission analysis subsystem
● Scilab ☆ 2

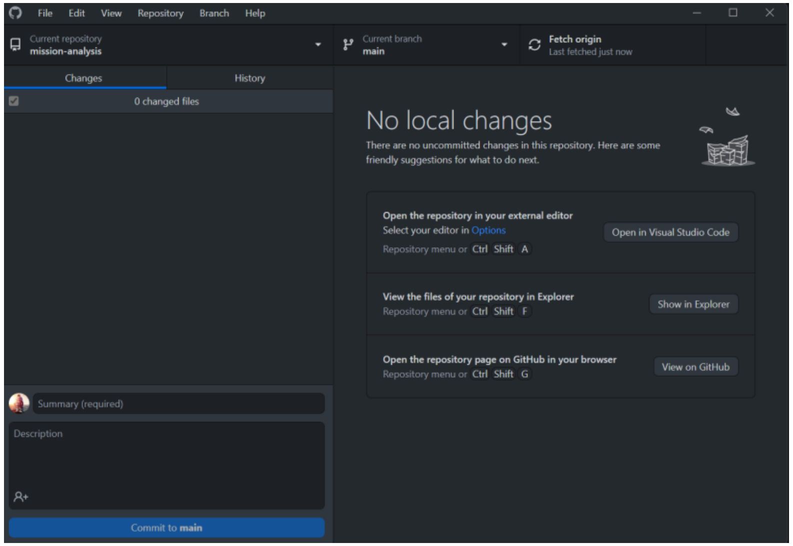
iridium-payload · Public
Repo of the Iridium payload subsystem. Includes a modem driver.
● Jupyter Notebook ☆ 1

gravimetry-payload · Public
● Python ☆ 1

power-budget · Public
● Python

thermal · Public
Repo of the thermal subsystem
● HTML

You can use GitHub desktop to push/pull changes between the mission-analysis repository and your machine.



Current repository: mission-analysis

Current branch: main

Fetch origin
Last fetched just now

☑ 0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl Shift A

View the files of your repository in Explorer
Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

Useful Links

- TOLOSAT GitHub : https://github.com/TOLOSAT
- Mission Analysis Repository : https://github.com/TOLOSAT/mission-analysis
- Detailed setup guide (with more links) : https://github.com/TOLOSAT/mission-analysis/blob/main/python/setup_guide/README.md
- GitHub Desktop : https://desktop.github.com/
- ANACONDA : https://www.anaconda.com/
- Miniforge : https://github.com/conda-forge/miniforge
- PyCharm : https://www.jetbrains.com/pycharm/
- PyCharm license for students : https://www.jetbrains.com/shop/eform/students
- GitHub Global Campus for students : https://education.github.com/globalcampus/student
- Tudat documentation : https://docs.tudat.space/en/stable/
- Tudatpy API Reference : https://py.api.tudat.space/en/latest/

The End

... or the beginning !

Arnaud MULLER