FocusPOD, the new POD SW used at the CPOD Service

Abstract

GMV has developed a new POD SW: FocusPOD, as an internal R&D activity. FocusPOD has been written from scratch in **C++ & Python**, with a completely new design, with the goal of supporting different applications in the fields of Precise Orbit Determination, Flight Dynamics, Space Traffic Management, Geodesy and Simulation. The development started in January 2021 and reached the operational level of maturity in January 2023, when it became the computational layer of the Copernicus POD Service.

In terms of **architecture**, the core layers of the **FocusPOD** SW have been designed as a **library**, to support a flexible development of applications. One of the key design elements is the **data model** which aims at being a rationale representation of the reality that provides a common, coherent data layer for all algorithms. In terms of **performance**, it is shown that the new SW presents improvements in runtime, reducing the product generation timeliness, while reaching the same levels of accuracy as other state-of-the-art SW packages.

FocusPOD

What is *FocusPOD*?

- *FocusPOD* is a state-of-the-art POD SW developed from scratch by GMV starting on 2021.
- \Rightarrow In its core, it is designed as a geodesy and flight-dynamics **library** written in **C++ & python** following modern programming standards: C++ 17 standard, templates & generic programming based on the STL..
- ⇒ Inspired on other FD SW packages like NAPEOS and Orekit but using modern technologies as those used by GIPSY-X and GODOT.
- ⇒ It makes use of external, open-source **dependencies** whenever possible: fmt, spdlog, eigen, xerces..
- ⇒ **Parallelisation** needs considered since the beginning.

Use cases

⇒ The core library is flexible and readily supports **various use-cases**:

-End-to-end, ad-hoc **programs** (e.g. read Level O GNSS data, pre-process observations, generate preliminary orbit, carry out ambiguity fixing, generate product) – in support of dedicated Instrument Processing Facilities (IPF). Curren CPOD Service makes use of this architecture.

- -Micro-Services, with clear APIs and interfaces. Next evolutions of CPOD Service will be based on this approach.
- -**Python** scripts / Jupyter notebooks. Mostly useful for analysis and experimentation.

FocusPOD use case: CPOD3

The Copernicus POD Service is composed by a consortium of companies and institutions lead by GMV, which is responsible for the operations, quality control and SW evolutions with the support of the CPOD QWG. The CPOD Service aims at providing operational orbit products for the ground segment operations of the Copernicus Sentinel-1, -2 -, 3 and -6 for ESA and EUMETSAT. Since Jan. 2023, The CPOD Service is based on *FocusPOD*, the new state-of-the-art POD SW developed from scratch by GMV. All the service products are routinely generated with similar or better performance than the legacy system. Ever since the transition, the SW has proven its **operational robustness** and good performance.

CPOD based on CPOD based on FocusPOD NAPEOS 2023 – .. 2014 - 2022

CPOD products are publicly available at the **Copernicus Open Access Hub**





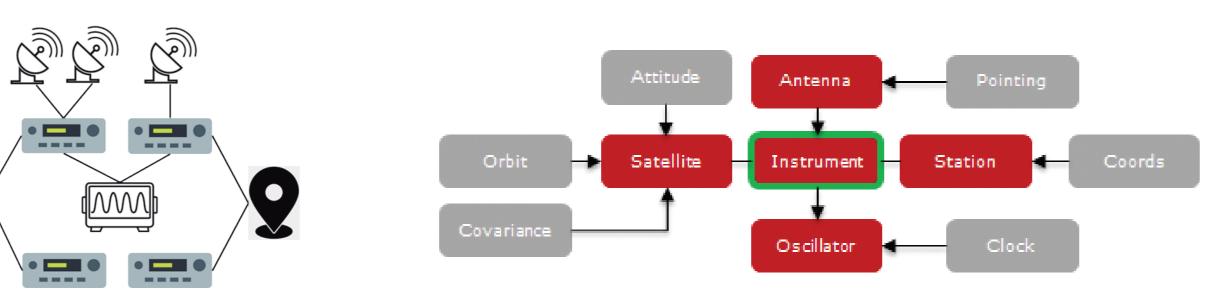
Current capabilities

- ➡ POD based on GNSS (multi-signal, multi-constellation) based on dynamic and reduced-dynamic approaches.
- SLR processing capabilities.
- ⇒ Single-receiver Ambiguity Fixing.
- ⇒ Support to advanced GNSS data **simulation**.
- ⇒ Geometric **events** calculation.
- ⇒ Space Surveillance and Tracking (SST) typical observations (radar, optical, angles) and basic algorithms:
- Initial orbit determination (IOD).
- Association track-to-orbit, track-to-track, orbit-to-c
- Preliminary manoeuvre detection.

Data model

- ⇒ **Key design principle** of the library
- Aims at being a **rationale and unique** representation of the physical reality
- ⇒ Enforces separation of data and algorithm
- ⇒ Establishes **relationships** between elements (e.g., 1 instrument on-board a satellite connected to an antenna) & optimised look-up mechanisms
- ⇒ Designed to handle large volumes of data (e.g., POD observations, SST catalogue, mega-constellations, ...)
- ⇒ To **centralise** interactions with external data (interfaces: files, DB, sockets...)
- ⇒ It is manifested as a **Scenario**, which is the central container of all data of an execution.

A



POD models & algorithms

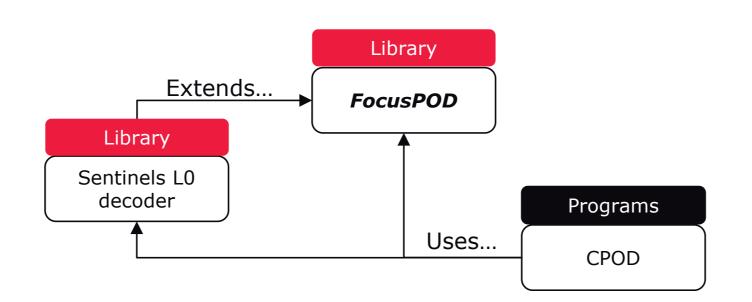
1115.		
	Category	Models
orbit	Frames	IERS 96 and 2010 conventions Linear mean pole
	Gravity	COST-G, EIGEN, EGM Support to time-varying terms Ocean tides (OTIERS): FES2014 Solid tides (STIERS): IERS 2010 Atmospheric gravity (AOD1B)
	Third-body	JPL ephem. DE405, DE421
	Density	MSISE00
	Space weather proxies	RSGA, Celestrak
	Radiation	Solar Radiation Pressure (SRP), albedo, infra-red
nt	Empiricals	Constant-Per-Revolution (CPR)
	Ionosphere	Nequick
	Troposphere	Mendes-Pavlis, Niell, Saastamoinen
	Satellite	Fixed area, macro-model Theoretical or real attitude laws Impulsive and long manoeuvres
iblo	Station	North-East-Up or ad-hoc pointing Post-Seismic Deformations

References

- ⇒ IERS 2010 conventions: Petit and Luzum (2010) IERS Conventions, IERS Technical Note 36.ISBN 3-89888-9898-6. ⇔ COST-G: Peter et al. (2022) COST-G gravity field models for precise orbit determination of Low
- Earth Orbtining Satellites. Adv Space Res 69(12), 4155–4168. DOI:10.1016/j.asr.2022.04.005 ⇒ EIGEN: Lemoine et al. (2019) CNES/GRGS RL04 Earth gravity field models, from
- GRACE and SLR data. DOI:10.5880/ICGEM.2019.010 ⇒ FES2014: Lyard et al. (2021) FES2014 global ocean tides atlas: design and
- performance. Ocean Sci 17, 615-649. DOI:10.5194/os-17-615-2021
- ⇒ JPL DE421: Folkner et al. (2009) The Planetary and Lunar Ephemeris DE421. JPL Interplanetary Network Progress Report 42-178
- ⇒ MSISE00: Picone et al. (2002) NRLMSISE-00 empiricial model of the atmosphere: Statistical comparisons and scientific issues. JGR 107(A12). DOI:10.1029/2002JA009430
- ⇒ Mendes-Pavlis: Mendes & Pavlis (2004) High-accuracy zenith delay prediction at optical wavelengths. GRL 31(14). DOI:10.1029/2004GL020308
- ⇒ Niell: Niell (1996) Global mapping functions for the atmosphere delay at radio wavelengths, JGR 101, 3227-3246. DOI:10.1029/95JB03048
- Saastamoinen: Saastamoinen (1973) Contribution to the theory of atmospheric refraction. Bulletin Géodesigue, 107, 13-34.
- ⇒ NAPEOS: Springer et al. (2011) NAPEOS: The ESA/ESOC Tool for Space Geodesy. Geophys. Res. Abstracts 13, EGU2011-8287.

Library vs Program

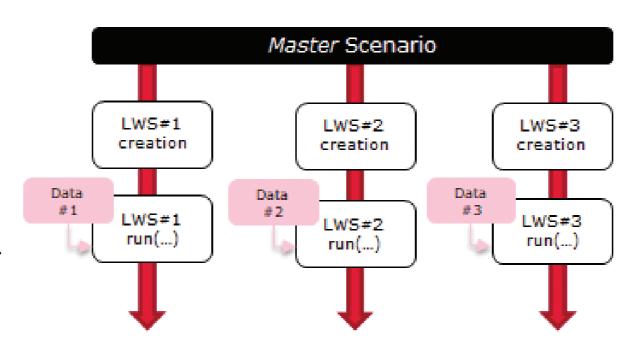
- ⇒ Big efforts made to implement generic algorithms, supported by a sensible and rational data model, to create a **library** that may support any type of application.
- ⇒ *FocusPOD* is not a collection of programs, each based on its own data model.
- ⇒ The library is **extensible** and **flexible** to allow to build additional modules and applications by using FocusPOD as a dependency. E.g., the Sentinels LO decoder extends the library observation provider as a new interface, which are then both used by the CPOD programs.



EGU General Assembly 2023 23–28 April 2023, Vienna, Austria

Parallelisation

- \Rightarrow Parallelisation needs have been considered since the conception of the central data model.
- ⇒ The central data container (Scenario) is not a singleton (i.e. several Scenarios can be created), to allow to have different blocks of data totally independent from one another – this readily supports paralellising between different Scenarios without any concurrency issues.
- ➡ To avoid processing data multiple times (e.g. load the Gravity Coefficients, pre-compute the frame conversion matrices...) FocusPOD supports the concept of "Light Weight Scenario", which is a Scenario that copies certain data from a Master Scenario. Then, these LWS run independently.
- Additionally, some algorithms (association, event computation...) internally implement parallel processing of some calculations making use of the "thread-pool" library.

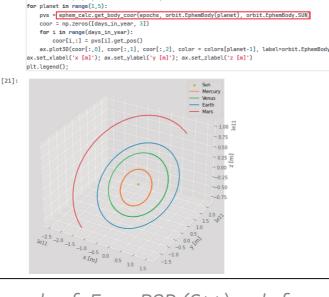


Wrappers

- ⇒ Bindings in python (based on **pybind11**) and on java (based on JNI automatically generated by SWIG) have been implemented to integrate the library algorithms in other languages.
- ⇒ Binding implemented on a **need-basis** for each project.
- Allows to favour usage of the library for analysis (e.g. in Jupyter Notebooks) and to exploit tools available in other languages (e.g. servers, visualisation tools...).
- Performance penalty mitigated by binding algorithms "entry-points", reducing data transfer between languages.

pybind11

JNI: Java Native Interface



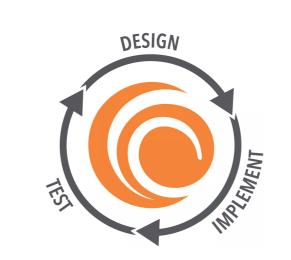
Example of FocusPOD (C++) code from binding & plotting tools from python



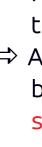
History of the development

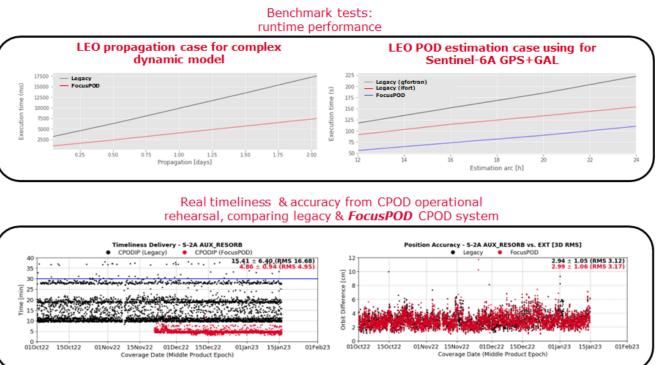
AGILE methodology with 2-week sprints: tackling complexity little by little.

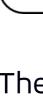
- \Rightarrow Peer-review & participatory design.
- ⇒ **Continuous integration** & validation are key for success.
- ➡ Different levels of validation:
- Unit-tests for all SW components.
- Integration tests and comparisons with legacy system for end-to-end cases.
- Benchmark and runtime performance analysis since early-on.
- Deploys in "operational" testing environments as soon as possible.
- ➡ Current size (excluding tests): 108 KLOC (lines of code).



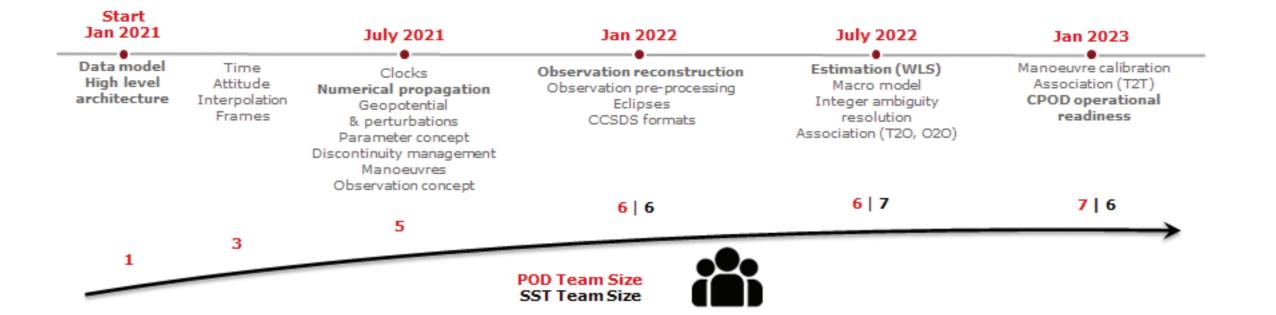








The Copernicus POD Service is financed under ESA contract no. 4000139509/22/I-BG, which is gratefully acknowledged. The work performed in the frame of this contract is carried out with funding by the European Union. The views expressed herein can in no way be taken to reflect the official opinion of either the European Union or the European Space Agency. The Copernicus POD Service is grateful to the CPOD QWG members for their support in the validation of the results, as well as to the International Laser Ranging Service (ILRS) and the International GNSS Service (IGS).



- J. Fernández (1) J. Berzosa (1) L. Bao (1) H. Peter (2) P. Féménias (3) M. A. Muñoz (1) M. Fernández (1) C. Nogueira (4) S. Lara (1)
- C. Fernández (1) E. Terradillos (1) (1) GMV AD., Tres Cantos, Spain
 - (2) PosiTim UG, Seeheim–Jugenheim, Germany
 - (3) ESA/ESRIN, Frascati, Italy
 - (4) EUMETSAT, Darmstadt, Germany

FocusPOD validation

⇒ **FocusPOD** has been validated in different levels: unit-tests, integration tests, end-to-end tests (e.g., POD, catalogue build-up...), benchmark & performance tests, operational scenarios...

⇒ As a showcase of the successful validation. we show below several cases with a common conclusion: state-of-the-art POD accuracy & very good runtime

performance. FocusPOD allows to beat CPOD legacy system runtime performance by a factor of 2!

Next steps

⇒ Research & advances in **POD** algorithms:

- DORIS processing
- SLR bias estimation based on multiple satellites (Sentinels + other LEOs + "geodetic" satellites like Lageos, Lares...)
- Kinematic GNSS processing
- Network OD (GNSS + Sentinels)
- Sequential estimators (Extended Kalman Filter, SRIF...)
- ➡ Implementation of Flight Dynamics algorithms:
- Manoeuvre planning and optimization
- Platform handling
- Cis-lunar and inter-planetary
- ⇒ Research & advances in Space Traffic Management & **SST**: - Improved association algorithms
- Collision risk analysis & close approach analysis
- ⇒ Enhance support to micro-services and distributed architectures, socket / real-time interfaces, DB-interaction at library level...

The objective is that the *FocusPOD* library becomes the backbone of a new set of services and products offered by GMV among different projects, becoming a reference in FD algorithms and geodesy, and staying aligned to modern technologies.

Acknowledgements

The development of FocusPOD has been done with internal R&D funding of GMV.

