

¹ Mamma Mia: A Python library for simulating underwater platform payloads using ocean model data

³ Thomas Prime  ¹

⁴ 1 National Oceanography Centre, Southampton, United Kingdom

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

⁵ Summary

⁶ Autonomous underwater vehicles (AUVs) and ocean gliders rely on a wide variety of onboard payloads—such as CTDs, ADCPs, fluorometers, and passive acoustic sensors—to collect ⁷ high-resolution in situ ocean data. Before deployment, mission planners and researchers often ⁸ need to simulate how these payloads will behave when embedded in complex 3D or 4D ocean ⁹ environments. This includes anticipating sampling resolution, environmental conditions, and ¹⁰ sensor-specific responses to varying water masses. Mamma Mia is an open-source Python ¹¹ library designed to address this need. It provides a flexible framework for simulating the ¹² behaviour of glider and AUV payloads within model-derived ocean fields, using data from ¹³ online sources such as the National Oceanography Centre and the Copernicus Marine Service ¹⁴ or locally from downloaded model data. The library supports multiple platforms at once and ¹⁵ allows users to simulate campaigns with a diverse range of platforms. Optionally the library ¹⁶ can also simulate the behaviour of specific platform types, Slocum gliders and Autosub's. The ¹⁷ library has been developed with an emphasis on usability, extensibility, and performance. It is ¹⁸ built on widely used scientific Python libraries—NumPy, Pyinterp, Xarray, and Zarr—and can ¹⁹ be integrated into real-time or offline mission planning workflows. It supports exporting to ²⁰ Zarr, a format suitable for visualization tools such as ParaView.
²¹

²² Statement of Need

²³ Ocean scientists often rely on numerical ocean models to plan missions and interpret observational data. Yet, while many community tools exist for working with model outputs, there is a ²⁴ lack of streamlined, open-source software aimed specifically at simulating sensor payloads on ²⁵ mobile marine platforms. Researchers currently rely on ad-hoc scripts, isolated institutional ²⁶ software, or manual workflows to: * Interpolate 4D model fields onto observed or simulated ²⁷ glider trajectories * Apply realistic sensor sampling behaviour (e.g., burst sampling, averaging, ²⁸ noise models) * Estimate expected data volumes and environmental variability along planned ²⁹ missions * Combine multiple payloads into a single simulated dataset
³⁰

³¹ Mamma Mia fills this gap by providing an accessible, well-tested, well-documented Python ³² toolkit for payload simulation using model data. It enables researchers, engineers, and students ³³ to test mission strategies, evaluate scientific return, and develop improved onboard algorithms, ³⁴ without requiring access to the physical vehicles.

³⁵ Features

- ³⁶ ▪ 4D interpolation engine supporting temperature, salinity, currents, biogeochemical tracers, ³⁷ and more
- ³⁸ ▪ Support for glider-sawtooth trajectories or arbitrary paths
- ³⁹ ▪ Built-in payloads:

- ```

40 – CTD sampling with noise and response-time emulation
41 – Radiometers (Chlorophyll and Downwelling Radiative Flux)
42 – Dissolved Gas (Oxygen)
43 ■ Able to create custom sensors, platforms and parameters
44 ■ Model input formats:
45 – CMEMS (Copernicus marine toolbox)
46 – NOC (OceanDataStore)
47 * Local NetCDF/Zarr
48 ■ Trajectory input formats
49 – CSV waypoints
50 – NetCDF with trajectory (e.g. from real deployment)
51 ■ Simulated platform (Slocum gliders and Autosubs)
52 ■ Export formats:
53 – Zarr

```

## Example usage

```

from mamma_mia import Campaign
from mamma_mia import inventory

print(f"Available groups in inventory: {inventory.list_inventory_groups()}")
print(f"Available platform types: {inventory.list_platform_types()}")
print(f"Available parameters: {inventory.list_parameters()}")
print(f"Available sensor types: {inventory.list_sensor_types()}")
print(f"Parameters Alias: {inventory.list_parameter_aliases()}")
print(f"sensors of type CTD: {inventory.list_sensors(sensor_type='CTD')}")
print(f"sensor info: {inventory.get_sensor_info(platform_type='Slocum_G2', sensor_type='')

print("=> starting Mamma Mia AUV Campaign test run <=>")
create campaign
campaign = Campaign(name="RAPID array virtual mooring",
 description="single Slocum glider deployment at a RAPID mooring",
 verbose=False,
)
create platform entity (mutable)
Churchill = inventory.create_platform_entity(entity_name="Churchill", platform="Slocum_G2")

register sensor to platform
Churchill.register_sensor(sensor_type="CTD")
register platform to the campaign for use in missions
campaign.register_platform(entity=Churchill)

add mission
campaign.add_mission(mission_name="RAD24_01",
 title="Churchill with CTD deployment at RAPID array mooring eb1l2n",
 summary="single glider deployed to perform a virtual mooring flight",
 platform_name="Churchill",
 trajectory_path="data/RAPID-mooring/rapid-mooring.nc",
 source_location="MSM",
 mission_time_step=60,
 apply_obs_error=True)

Set interpolators to automatically cache as dat files (no need to regenerate them, use

```

```
#campaign.enable_interpolator_cache()

build missions (search datasets, download datasets, build interpolators etc)
campaign.build_missions()

run/fly missions
campaign.run()

visualise the results
campaign.missions["RAD24_01"].plot_trajectory()
campaign.missions["RAD24_01"].show_payload()
campaign.export()
```

## 55 Figures

Glider Payload: PRACTICAL\_SALINITY

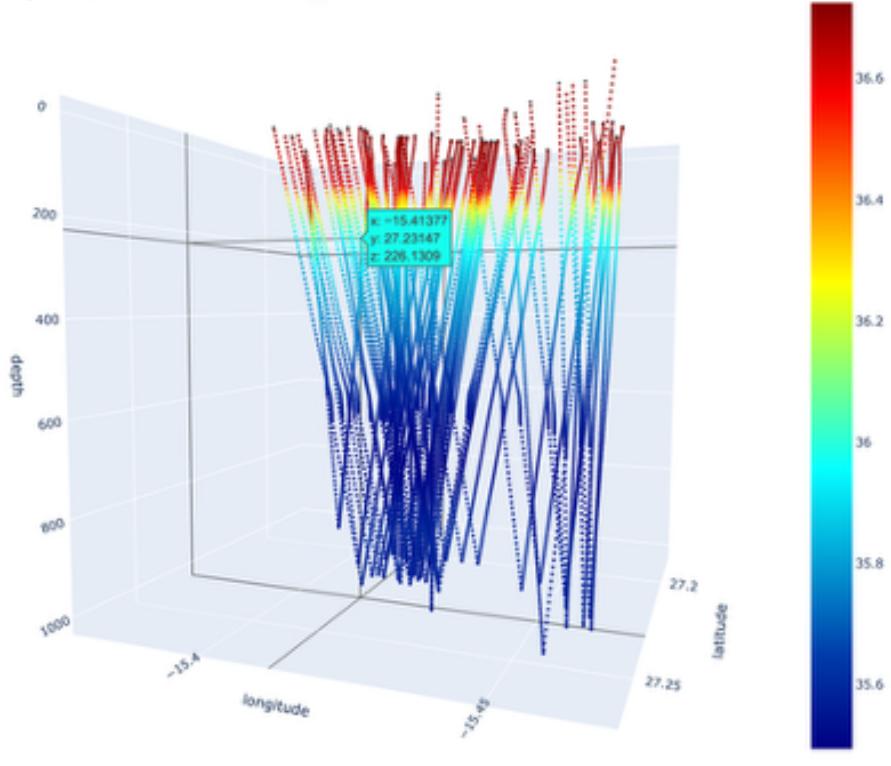


Figure 1: output\_example

## 56 Community Guidelines

57 Mamma Mia welcomes contributions through pull requests, GitHub issues, and community  
58 discussions.

<sup>59</sup> **Acknowledgements**

<sup>60</sup> We acknowledge the developers of Xarray, NumPy, PyInterp and glidersim without which  
<sup>61</sup> Mamma Mia wouldn't be possible.

<sup>62</sup> **References**

DRAFT