Gaia Astrometric Microlensing Events (GAME)

D2.4 – User guide for the processing software for astrometric model

Aalta Lab d.o.o.
Soška cesta 17, 5250 Solkan

Authors:
Uroš Kostić

Issue-Revision: 1.0
Issue date: 09/02/2024

Approval

Title GAME: D2.4 – User guide for the processing software for astrometric model	
Issue number 1	**Revision number** 0
Author Uroš Kostić	**Date** 09/02/2024
Approved by	**Date of approval**

Change log

Reason for change	Issue nr.	Revision number	Date
First issue	1	0	09/02/2024

Change record

Issue number	Revision number		
Reason for change	**Date**	**Pages**	**Paragraph**

Distribution

Name/Organisational unit

Contents

Summary

This is the *User guide for the processing software for astrometric model*, deliverable [D2.4].

The software is written in Python and can use multiple CPUs and GPUs to process the data. There are 2 programs, one for testing the simulated tracks and another for minimizing the MUWE function to obtain the microlensing parameters. The code runs in Linux.

1 Installation

1.1 Required modules

The code is written in Python3 and requires the following modules:

- `astromet`
- `jaxtromet`
- `astropy`
- `jax`
- `matplotlib`
- `numpy`
- `pandas`
- `scanninglaw`
- `tqdm`

`Astromet` is available at `https://github.com/zpenoyre/astromet.py` and `jaxtromet` is available at `https://github.com/maja-jablonska/jaxtromet.py`. Use their respective installation instructions. Other modules are available in the usual repositories.

If you plan to use also the GPUs, make sure you have installed the correct drivers and CUDA libraries before installing `jax`.

1.2 Applications

There are two applications available:

1. `minimize_one_example.py` which minimizes MUWE and retrieves the 11 parameters of the microlensing model,
2. `compare_tracks.py` which compares the tracks generated by the minimizer and the tracks used as input.

Since both are implemented as single Python3 scripts, no special installation is required.

2 Usage

Both applications are parallelized to run on multiple CPUs and GPUs using a Bash script `run.sh`. In the script, you can fine-tune the following:

- generation of input/output filenames,
- the number of used CPUs,

- the number of used GPUs,
- allocation to selected GPUs,
- whether to run the minimizer or comparison of the trucks.

The `run.sh` script takes 2 arguments: path to the directory with input data, and a flag, which defines whether we are processing microlensing data or not. To run the script, do the following in the terminal:

```
run.sh <path_to_input_dir> <True|False>
```

The input data for both applications should be in `.parquet` files, containing the following columns:

- `strip`
- `elapsedNanoSecs`
- `w`
- `theta`
- `wError`
- `refEpoch`

The output of `minimize_one_example.py` is split into one `.csv` file per CPU process. Each file contains the final parameters after the minimization, as well as the inittial and true values of MUWE:

- `set_id`: if the input data is organized in different sets, this is the `id` of each set.
- `lc_id`: the `id` of each track within the set.
- `ra`: right ascension of the source
- `dec`: declination of the source
- `u0`: microlensing parameter
- `t0`: microlensing parameter
- `tE`: microlensing parameter
- `piEN`: microlensing parameter
- `piEE`: microlensing parameter
- `pmrac_source`: proper motion in right ascension
- `pmdec_source`: propeer motion in declination
- `d_source`: $1/\mathrm{parallax}$
- `thetaE`: microlensing parameter

- MUWE: the resulting MUWE obtained with the minimization.
- MUWE_0: the value of MUWE for the initial values of the parameters.
- MUWE_true: the value of MUWE for the true values of the parameters (used in the simulator of the tracks).

The output of compare_tracks.py is split into one .parquet file per generated track. The files contain the tracks from jaxtromet, the true tracks from astromet and their differences:

- t_obs: time of observation
- x: the w variable variable from jaxtromet
- phi: the ϕ variable from jaxtromet
- ldracs: variable from jaxtromet
- lddecs: variable from jaxtromet
- ldracs_true: variable from astromet
- lddecs_true: variable from astromet
- x_obs: true values of w from astromet
- phi_obs: true values of w from astromet
- delta x: $w_{jax} - w_{ast}$
- delta phi: $\phi_{jax} - \phi_{ast}$
- delta racs: $racs_{jax} - racs_{ast}$
- delta decs: $decs_{jax} - decs_{ast}$