Performance Testing the MultiNest Algorithm in BAGLE's Model Fitter

Abby Schleigh[1], Natasha Abrams[1], Jessica Lu[1]
[1]University of California, Berkeley

Abstract

BAGLE (Bayesian Analysis of Gravitational Lensing Events) is a tool for the modeling and fitting of gravitational microlensing events, especially those of long duration such as black hole events. BAGLE simultaneously treats both photometric and astrometric data for microlensing models, including PSPL (point-source, point-lens), BSPL (binarypoint-source, point-lens), and PSBL (point-source, binary-point-lens), all with or without parallax. We ran performance tests of run time and accuracy based on the number of live points used to fit simulated events with MultiNest (a nested-sampling alternative to MCMC). Running these performance tests allows us to optimize the accuracy and efficiency of BAGLE model fitting. These results were used in determining the optimal number of live points when introducing BAGLE to a new audience through a tutorial.

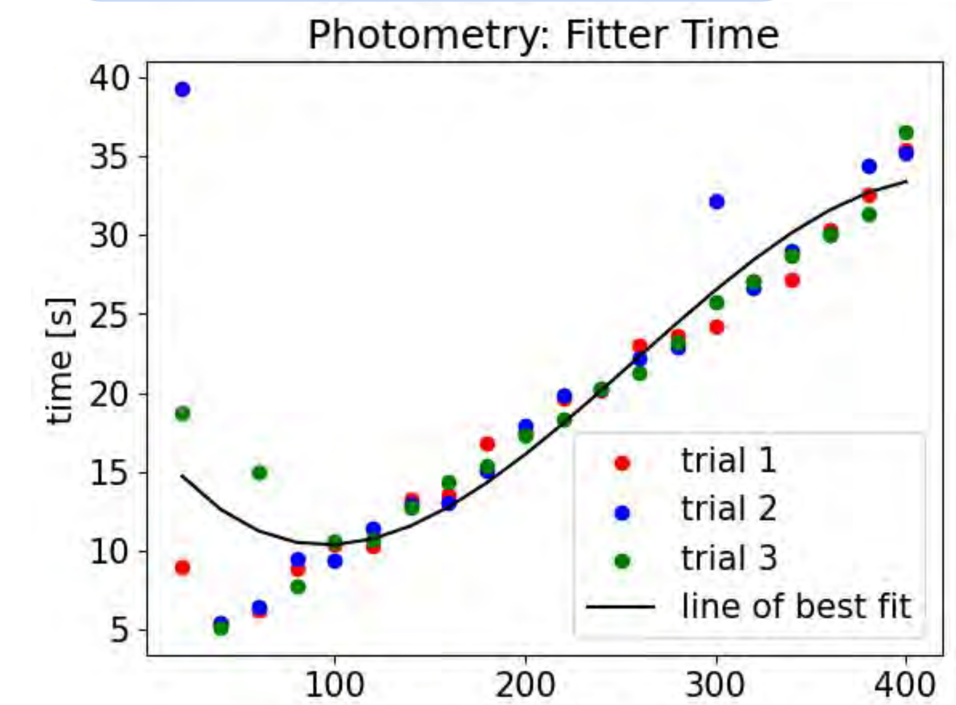
Method

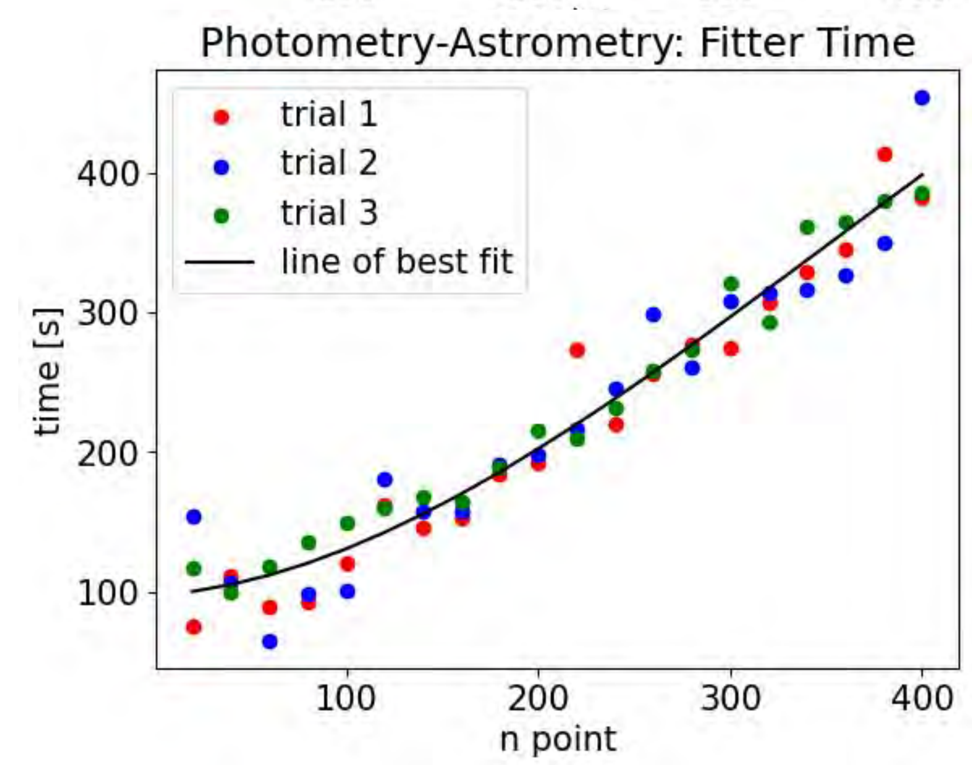
- **Model**: The main model of focus was Photometry no-Parallax and Photometry-Astrometry no-Parallax. This fitted model uses fake data, which is generated in BAGLE.
  - The Photometry data has 4080 points with observations every 1 day.
  - The Photometry-Astrometry data has the same 4080 Photometry points and sparse, 153 Astrometry points with observations every 14 days.
- The testing method is done by iterating the fitter through live points in the range [20,400] in step sizes of 20 with 3 'trials' of each live points value. Values like time run, RAM percentage used, best fit parameters, and χ^2 value are collected.
- BAGLE outputs a χ^2 value, which then has to be manually reduced with the amount of data points used and number of free parameters. This is equation (1).
- While most of the components collected test BAGLE itself, collecting RAM gives context to its performance based on the strain on the machine.

Priors

Parameter	Range
Lens Mass (M_{sun})	(5.0, 15,0)
Time of Photometric Peak (t_o; MJD)	(56990, 57010)
Angular distance between PS & PL (mas)	(-0.5, -0.3)
RA Lens PM (mas/yr)	(-1, 1)
Dec Lens PM (mas/yr)	(-8,-6)
RA Source PM (mas/yr)	(0, 3)
Dec Source PM (mas/yr)	(-2, 1)
Observer distance to lens (pc)	(3000, 5000)
Ratio of observer distance to PL vs PS	(0.45, 0.55)
Ratio of PS flux vs total	(0.5, 1.1)
PS Photometric Magnitude	(18.9, 19.1)
RA of PS at t_o (arcsec)	(-10^{-3}, 10^{-3})
Dec of PS at t_o (arcsec)	(-10^{-3}, 10^{-3})

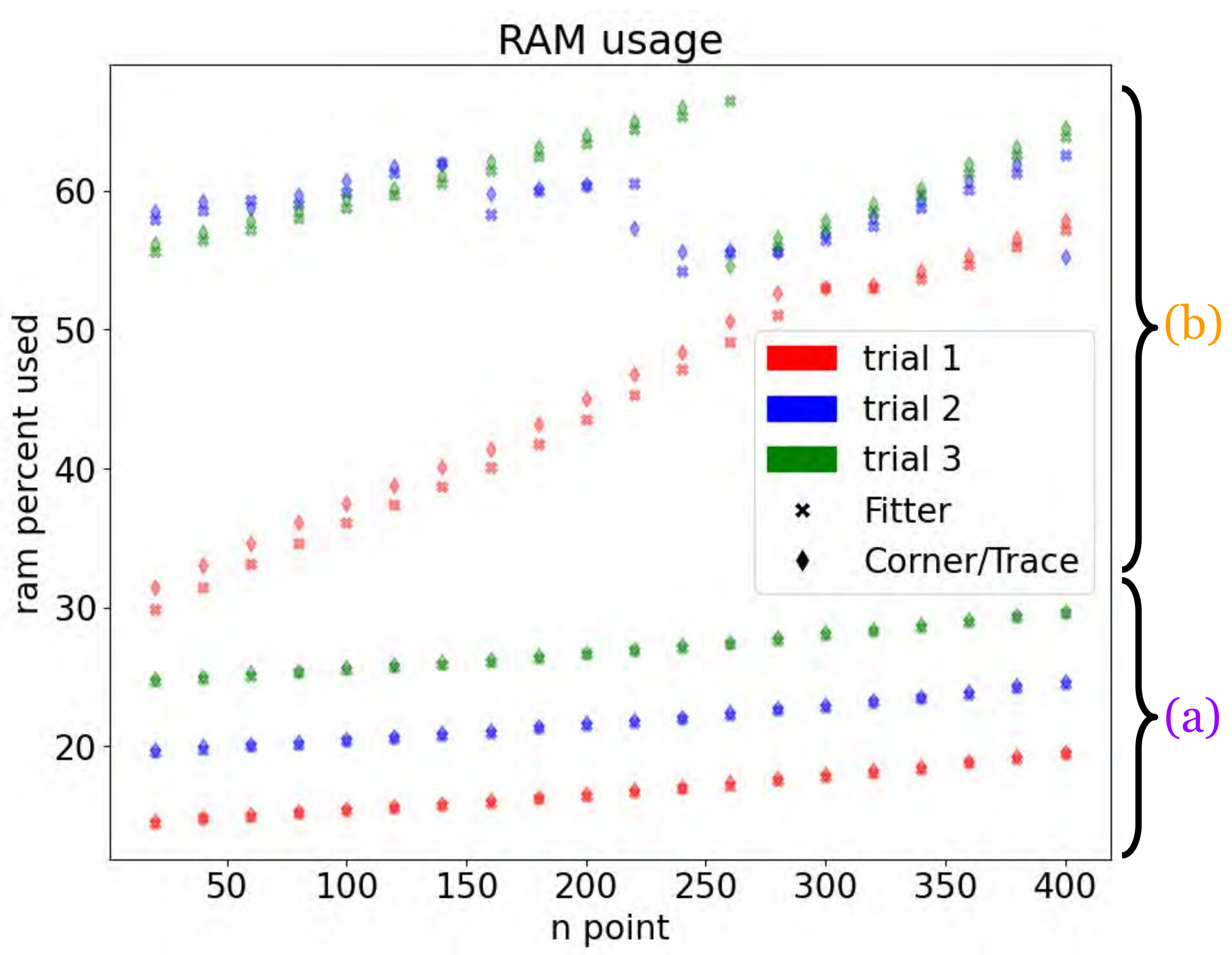
Run time





- Cubic function used for best fit line, as it provides insight into the high run time for lower live points.
- While almost at a linear relationship for the fitter time on both Photometry and Photometry-Astrometry, there is a period of lower value live points that takes longer before regulating to the linear relationship due to these values being closer to the number of free parameters:
  - In Photometry=7
  - In Photometry-Astrometry=13
- Future inspection will look at if there is a plateau in the fitter run time, and whether this run time is heavily affected by the performance of the machine (seen in the RAM data).

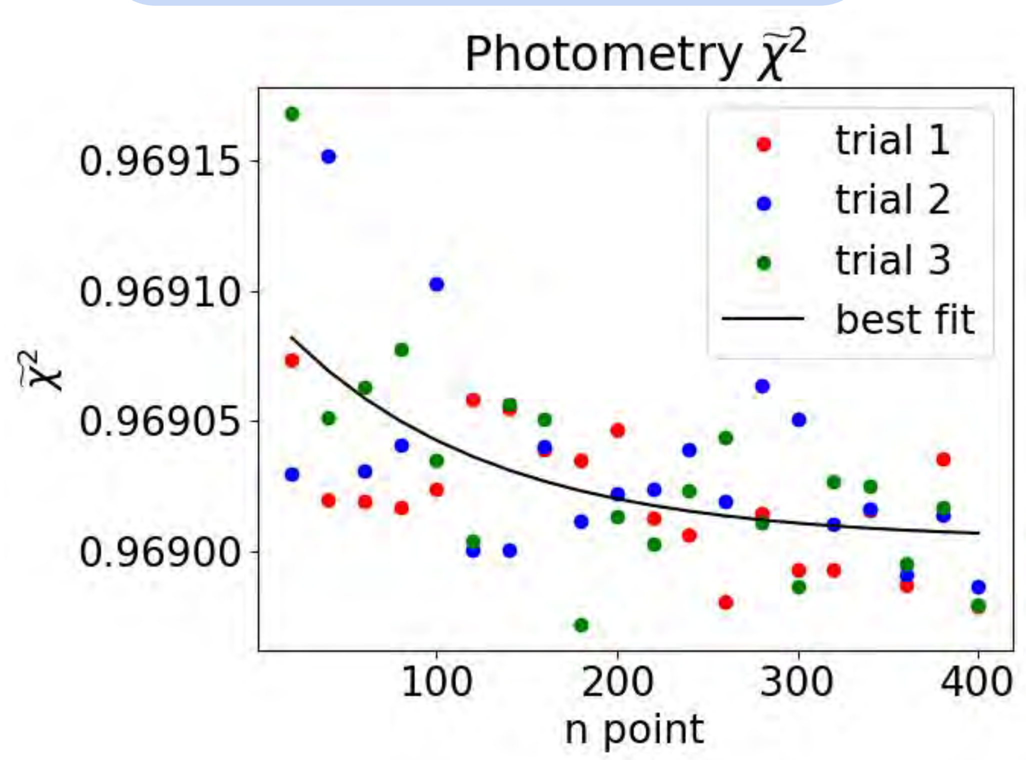
RAM

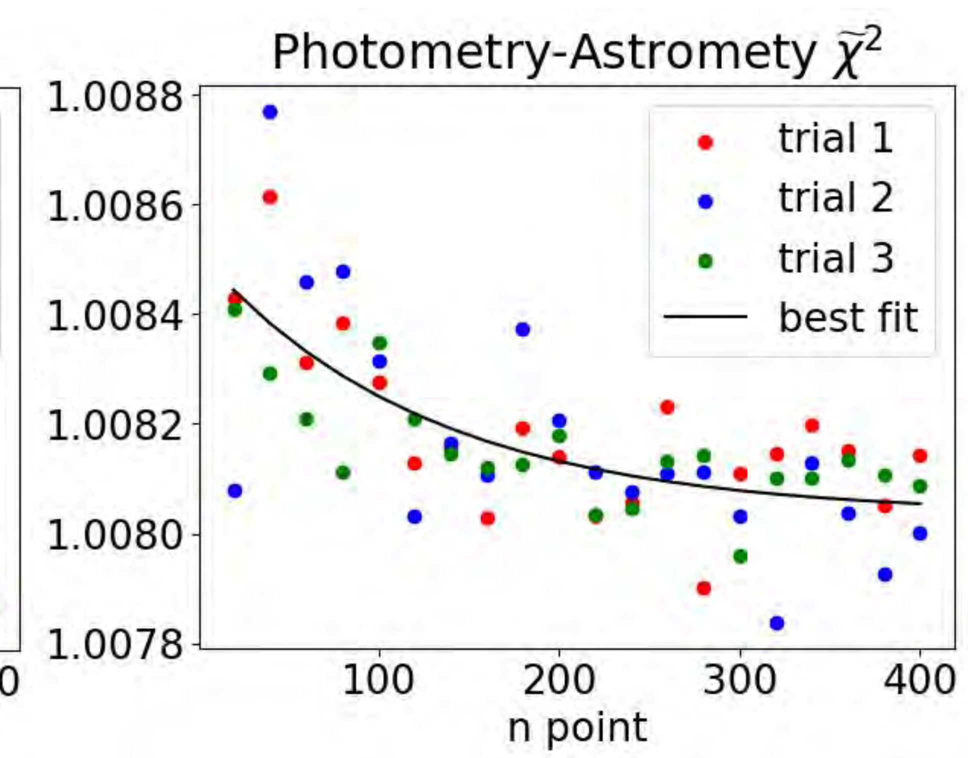


(a) Photometry trials
(b) Photometry-Astrometry trials

- With every new iteration and trial, the higher the RAM percentage.
  - Suggesting RAM leakage.
- Photometry-Astrometry uses more RAM than at a faster rate than Photometry.
  - Photometry-Astrometry creates more plots
- Dip of RAM percentage consistently at ~60% RAM, suggesting a built in garbage collector or other programs being run by a different user.
- Future inspection will involve how and why these RAM leaks are happening, and how to slow it down.

Accuracy





- Exponential function used for best fit line.
- There was a visible plateau of the reduced $\tilde{\chi}^2$ value within the range of the iterated live points.
- There is a point that simultaneously optimizes for $\tilde{\chi}^2$ value and time run.
- Future inspection will involve testing other models and cross checking the $\tilde{\chi}^2$ values with time run and RAM.

$$\tilde{\chi}^2 = \frac{\chi^2}{n_{data} - n_{param}} \quad (1)$$

Links

QR code has links to BAGLE's GitHub and documentation, along with a link to the Moving Universe Lab website.

abbyschleigh@berkeley.edu