Methods to calculate magnification in `MulensModel`

Radek Poleski
last update: Oct 2025

`MulensModel` offers a wide range of methods used to calculate magnifications. These methods are passed to `Model` class using `set_magnification_methods()` function. For each method one has to pass the time ranges when the method will be used. These parameters are passed in a list, e.g.:

```
model = Model(...)
model.set_magnification_methods(
    [2455745., 'Hexadecapole', 2455746., 'VBBL', 2455747., 'Hexadecapole', 2455748.])
```

There are two other useful functions. First, `set_default_magnification_method()` allows setting method that is used outside time ranges specified above. Second, `set_magnification_methods_parameters()` allows providing additional parameters for calculations. Currently, only `VBBL` and `Adaptive_Contouring` allow providing these parameters.

Point lens methods:

- `point_source` – the simplest thing that exists, also called "Paczyński curve": $A(u) = (u^2 + 2) / \left(u\sqrt{u^2 + 4}\right)$. Note that if shear and convergence are defined, then calculation follows Chang and Refsdal (1979).
- `finite_source_uniform_Gould94` – for the finite source with uniform profile (i.e., no limb-darkening effect) use approximation presented by Gould (1994). It works only for small ρ, i.e, $\rho \lesssim 0.1$.
- `finite_source_uniform_WittMao94` – for the finite source with uniform profile use method presented by Witt and Mao (1994). It interpolates pre-computed tables.
- `finite_source_LD_WittMao94` – for the finite source with limb darkening integrate many uniform profiles. For each uniform profile, `finite_source_uniform_WittMao94` method is used. For description on how the uniform profiles are combined see, e.g., Bozza et al. (2018).
- `finite_source_LD_Yoo04` – for the finite source with limb darkening use Yoo et al. (2004) approximation. It works only for small ρ, i.e, $\rho \lesssim 0.1$.
- `finite_source_uniform_Lee09` – for the finite source with uniform profile (Lee et al. 2009) but works well for large ρ as well (e.g., $\rho = 2$). It is significantly slower than approximate method `finite_source_uniform_Gould94`.
- `finite_source_LD_Lee09` – for the finite source with limb darkening that works well for large sources (e.g., $\rho = 2$). This method is much slower than `finite_source_LD_Yoo04`.

Please note that `finite_source_uniform_Gould94` and `finite_source_LD_Yoo04` interpolate pre-computed values. This interpolation should be very accurate. If you want to test it, then request direct calculations using `finite_source_uniform_Gould94_direct` or `finite_source_LD_Yoo04_direct`. For u/ρ that is outside pre-computed values the direct calculation is used as well.

Binary lens methods:

- `point_source` – assumes the source is just a point (hence not valid near caustics) and solves fifth order complex polynomial once.
- `quadrupole` – uses Taylor expansion – evaluates point-source magnification at 9 points. Works only outside caustic.
- `hexadecapole` – uses Taylor expansion – evaluates point-source magnification at 13 points. Works only outside caustic.
- `VBBL` – Bozza (2010) method – finite source with limb darkening. Most widely used method nowadays. Parameters that ca be set: `accuracy`.
- `twinkle` – Wang et al. (2025) method – finite source with limb darkening. Uses the GPU to calculate magnification. Parameters that can be set: `device_num`, `N_stream`, and `RelTol`.
- `Adaptive_Contouring` – Dominik (2007) method – finite source with limb darkening. Parameters that can be set: `accuracy` and `ld_accuracy`.
- `point_source_point_lens` – approximates binary lens as a single lens. It is useful when binary lens effects are negligible and binary lens calculations may cause numerical errors, e.g., $q \approx 10^{-6}$ and source far from caustics.

Note that if you define shear and convergence (Peirson et al. 2022), then `MulensModel` uses properly modified versions of: `point_source`, `quadrupole`, or `hexadecapole`.

Triple lens methods – under construction. Please come back later!