`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": $A(u) = (u^2 + 2) \,/\, \left(u\sqrt{u^2 + 4}\right)$.
- `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_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`
- `quadrupole`
- `hexadecapole`
- `VBBL` – parameters `accuracy`
- `Adaptive_Contouring` – parameters `accuracy` and `adaptive_contouring`
- `point_source_point_lens`

Triple lens methods – under construction. Come back soon!