API

class batman.**TransitParams**

Object to store the physical parameters of the transit.

Parameters:
- **to** (*float, optional*) – Time of inferior conjunction.
- **t_secondary** (*float, optional*) – Time of secondary eclipse center.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius [in stellar radii].
- **a** (*float*) – Semi-major axis [in stellar radii].
- **inc** (*float*) – Orbital inclination [in degrees].
- **ecc** (*float*) – Orbital eccentricity.
- **w** (*float*) – Argument of periapse [in degrees]
- **u** (*array_like*) – List of limb darkening coefficients.
- **limb_dark** (*str*) – Limb darkening model (choice of “nonlinear”, “quadratic”, “exponential”, “logarithmic”, “squareroot”, “linear”, “uniform”, “power2”, or “custom”).
- **fp** (*float, optional*) – Planet-to-star flux ratio (for secondary eclipse models).

Note:

- Units for the orbital period and ephemeris can be anything as long as they are consistent (e.g. both in days).
- The orbital path is calculated based on *to* for primary transits and *t_secondary* for secondary eclipses.

Example:

```
>>> import batman
>>> params = batman.TransitParams()
>>> params.t0 = 0.                       #time of inferior conjunction
>>> params.per = 1.                      #orbital period
>>> params.rp = 0.1                      #planet radius (in units of stellar radii)
>>> params.a = 15.                       #semi-major axis (in units of stellar radii)
>>> params.inc = 87.                     #orbital inclination (in degrees)
>>> params.ecc = 0.                      #eccentricity
>>> params.w = 90.                       #longitude of periastron (in degrees)
>>> params.u = [0.1, 0.3]                #limb darkening coefficients
>>> params.limb_dark = "quadratic"       #limb darkening model
```

class batman.**TransitModel**(*params*, *t*, *max_err=1.0*, *nthreads=1*, *fac=None*, *transittype='primary'*, *supersample_factor=1*, *exp_time=0.0*)

Class for generating model transit light curves.

Parameters:
- **params** (a *TransitParams* instance) – A **TransitParams** object containing the physical parameters of the transit
- **t** (*ndarray*) – Array of times at which to calculate the model.
- **max_err** (*float, optional*) – Error tolerance (in parts per million) for the model.
- **nthreads** (*int, optional*) – Number of threads to use for parallelization.
- **fac** (*float, optional*) – Scale factor for integration step size
- **transittype** (*string, optional*) – Type of transit (“primary” or “secondary”)
- **supersample_factor** (*integer, optional*) – Number of points subdividing exposure
- **exp_time** (*double, optional*) – Exposure time (in same units as *t*)

Example:

```
>>> m = batman.TransitModel(params, max_err = 0.5, nthreads=4)
```

calc_err(*plot=False*)

Calculate maximum error for transit light curve calculation.

Parameters: **plot** (*bool*) – If `True`, plots the error in the light curve model as a function of separation of centers.
Returns: Truncation error (parts per million)
Return type: float

get_t_conjunction(*params*)
Return the time of primary transit center (calculated using *params.t_secondary*).

get_t_periastron(*params*)
Return the time of periastron passage (calculated using *params.to*).

get_t_secondary(*params*)
Return the time of secondary eclipse center (calculated using *params.to*).

get_true_anomaly()
Return the true anomaly at each time

light_curve(*params*)
Calculate a model light curve.

Parameters: **params** (A *TransitParams* instance) – Transit parameters
Returns: Relative flux
Return type: ndarray
Example:

```
>>> flux = m.light_curve(params)
```