Draft version December 19, 2021
Typeset using LaTeX default style in AASTeX63

AnalyticLC - User Manual

1. GENERAL

The motivation behind the code is to enable a fast translation of orbital elements to a photometric light curve, and/or RV/astrometry values. The calculation of the dynamics is based on expansion of the disturbing function up to fourth order in eccentricities and inclinations, and therefore one should expect its accuracy to be within these limits. The translation to light curve is based on the well-known Mandel-Agol occultation model. The translation to RV and astrometry signal is done by converting the orbital elements to planets velocities and positions, and then to their stellar barycentric counterparts. Due to the dimensionless nature of the parameters, the RV is returned in units of stellar radii/unit time, and the astrometry values in units of stellar radii. As explained below, the inputs can be of any time unit, as long as they are consistent with each other. The RV output time unit will correspond to the input units.

The code is written in `Matlab`, and it uses a several subfunctions, but the interface with the user is done only through one function - `AnalyticLC`.

2. PREPARATION

Before running the function `AnalyticLC`, some preparation steps need to be done:

- Copy all the .m files to your `Matlab` path.
- Copy the file `LaplaceCoeffs.mat` to your drive.
- Open `Matlab`, and type `LaplaceCoeffs('load',FilePath)`.

  `FilePath` should contain the full path including the file name, for example a valid usage is `LaplaceCoeffs('load','/MyFolder/LalaceCoeffs.mat')`, but not `LaplaceCoeffs('load','/MyFolder')`. If the file path is the default path `/Matlab/Data/LaplaceCoeffs.mat`, you can type `LaplaceCoeffs('load')`.

3. RUNNING

`AnalyticLC` receives a set of numerical arrays as obligatory inputs, and a set of optional inputs. All obligatory inputs are dimensionless apart from those related to times; therefore, one can choose any time units desired - we worked with seconds for all quantities, but that is up to the user selection.

Usage example with obligatory inputs only:

```
LC = AnalyticLC(P,Tmid0,ror,aor,mu,ex0,ey0,I0,Omega0,t,u1,u2);
```

The obligatory inputs are, to order:

number	symbol	description	units	length
1	`P`	orbital period	[time]	horizontal ascending vector; entry per planet
2	`Tmid0`	first time of mid-transit	[time]	horizontal vector; entry per planet
3	`ror`	planet-to-star radii ratio		horizontal vector; entry per planet
4	`aor`	first planet semi-major axis over stellar radius		1
5	`mu`	planet-to-star mass ratio		horizontal vector; entry per planet
6	`ex0`	free eccentricity x component at first point of time axis		horizontal vector; entry per planet
7	`ey0`	free eccentricity y component at first point of time axis		horizontal vector; entry per planet
8	`I0`	Inclination at first point of time axis		horizontal vector; entry per planet
9	`Omega0`	Longitude of ascending node at first point of time axis		horizontal vector; entry per planet
10	`t`	time stamps to evaluate light curve flux values	[time]	unlimited; needs to be a vertical vector
11	`u1`	Limb darkening parameter 1		1
4	`u2`	Limb darkening parameter 2		1

3.1. *Outputs*

`LC` - the obtained light curve, corresponding to each time stamp in `t`.

`RV_o_r` - radial velocity values per time stamp specified by the optional input `tRV`. Given in units of stellar radius per unit time.

`Y_o_r, Z_o_r` - Displacement of star on the sky plane; these two axes can be transformed to right ascention or declination if the transit chord orientation on the sky is known.

`O` - a structure including additional, optional, output variables, detailed below.

3.2. *Optional Inputs*

The optional inputs give the user further capabilities for using AnalyticLC. Usage is by specifying the required additional input name and then its value, for example:

```
LC = AnalyticLC(P,Tmid0,ror,aor,mu,ex0,ey0,I0,Omega0,t,u1,u2,'MaxTTV',12*3600,'BinningTime',120);
```

`BinningTime` - time discretization for flux binning. Enables employing a binning scheme (per cadence type - see below) with discrete time jumps specified by BinningTime.

`CadenceType` - Cadence type of each time stamp (vector with the same length as t). Cadence types are numbered by the integers 1,2 etc.

`MaxTTV` - Maximal allowed TTV amplitude. Can be set to each planet, or a single number applying for all planets. If the TTV amplitude of any of the planets exceeds `MaxTTV` the function returns nan for all time stamps. Useful if for restricting `AnalyticLC` when driven via some MCMC process.

`tRV` - time stamps for evaluating RV. Should be a vertical vector.

`ExtraIters` - Extra iterations to be employed when evaluating the forced elements via Lagrange's equations; can enhance accuracy to some extent. From our experience, more than 1 usually does not yield significant benefit.

`OrdersAll` - Number of orders to be calculated when truncating the series expansion of the forced orbital elements calculation in j. If not specified, the number will be selected automatically by the code by finding the nearest mean motion resonances.

`OutputList` - A list of additional outputs to be returned to the fifth output variable (see below). Can include the orbital elements, times of mid-transit, impact parameters etc. See full list below. Default is 'all'; in that case if the fifth output is called for then all outputs specified below will be returned.

`Calculate3PlanetsCrossTerms` - if different than zero, this flag tells the code to perform a correction relevant for more than 2-planets systems to the second order in mass. Relevant when two super-periods are close to each other; see details in paper.

3.3. *Additional Optional Outputs*

- `tT` - nominal transit times (without TTVs)
- `tRV` - time stamps of RV
- `zT` - complex eccentricities at times `tT`; one column per planet
- `zRV` - complex eccentricities at times `tRV`; one column per planet
- `uT` - complex inclination at times `tT`; one column per planet
- `uRV` - complex inclination at times `tRV`; one column per planet
- `D` - transit durations associated with each time stamp; one column per planet

- `Tau` - transit ingress-egress times associated with each time stamp; one column per planet
- `w` - angular velocity at mid transit associated with each time stamp; one column per planet [RAD/time]
- `d` - star-planet separation at mid transits associated with each time stamp; one column per planet
- `b` - impact parameters associated with each time stamp; one column per planet
- `AssociatedTmid` - the associated time of mid transit per time stamp; one column per planet
- `AssociatedInd` - the associated transit number per time stamp; one column per planet
- `AllLC` - individual planetary light curve (whose sum is the total light curve); one column per planet
- `TmidActualCell` - a cell array of actual transit times, after counting for the TTV
- `free_e_T` - complex free eccentricities at transit times
- `free_I_T` - complex free inclinations at transit times
- `free_e_RV` - complex free eccentricities at RV times
- `free_I_RV` - complex free inclinations at RV times
- `TTVCell` - a cell array of the TTV's, one cell per planet
- `dzCell` - cell array of forced eccentricities at transits, one cell per planet
- `dLambdaCell` - cell array of forced mean longitudes at tranits, one cell per planet
- `zfreeCell` - cell array of free eccentricities at transits, one cell per planet
- `dbdt, b0` - mean slope and intersection with t=0 of impact parameter variation; one per planet Lambda_T - mean longitudes at transits
- `BeginIdx, EndIdx` - beginning and ending indices for the lists ending with T: `free_e_T`, `zT`, `Lambda_T` etc. For example, the values corresponding to the second planet are those at positions `BeginIdx(2)` until `EndIdx(2)`.

4. SPEED-UP MECHANISMS

4.1. *Laplace Coefficients*

One of the most time-consuming computation step of any analytic treatment of orbital elements and TTV's is the calculation of the Laplace coefficients, which are integrals that cannot be solved analytically. Because the code is designed to be able to run multiple times by some Monte-Carlo driving algorithm, the Laplace coefficients can be calculated in advance and saved in the memory as a large table. A table is provided with this code in the file LaplaceCoeffs.mat for the user, with steps of 0.5×10^{-5} in α, with j values from 0 to 30, with s values of 1/2, 3/2 and 5/2, and with derivatives up to fifth order. These should be sufficient for any standard scenario; however, the user can construct his own table at other parametrizations. To give the user who wishes to re-calculate the table a time estimate, the calculation time for the current table was about 24 hours.

4.2. *Mandel-Agol look up table*

A main goal of this code is to enable fast fitting of a model to a light curve. If the limb-darkening parameters are not being fitted for, then the only two variables entering the Mandel-Agol calculation are the plant-to-star sky-projected separation, and the planet-to-star radii ratio. Hence, we added the possibility to construct in advance a look-up table of flux values in the separation-radius plane to enable fast calculation of the light curve when the limb darkening parameters remain constant. If you wish to use such a look-up table, you can construct it by typing `MALookupTable('init',rvec,zvec,u1,u2)`, where `rvec` is the grid planet-to-star radii ratio, `zvec` is the grid of sky-separation in stellar radii, and `u1` and `u2` are the limb darkening coefficients parameter. From our experience, using a grid in jumps of 2×10^{-3} in radii ratio and 2×10^{-3} in separation yields an accuracy better than 10^{-6}, but the user can set his own values. If the table has been constructed and now needs to be cleared to transfer to direct calculation, type `MALookupTable('clear')`. If `AnalyticLC` receives `u1,u2` values different than those stored in the look-up table, it will perform the direct calculation for the requested `u1,u2` values.

4.3. *Comparison vs. N-body*

For the user who wishes to translate the free values to instanteneous ones in order to feed an N-body integrator, the subfunction `ForcedElements1` will do the work. This function receives a set of free elements and returns their forced counterparts. The instanteneous elements will be the free values plus the forced values. Note that the instanteneous semi-major axis (and hence the orbital period) should sometimes be optimized for in order to obtain matching between the N-body integration and the analytic calculation, as the sensitivity of transit times to the orbital period is larger than for the other elements.