UTM, a universal simulator for lightcurves of transiting systems

Hans Deeg

Instituto de Astrofísica de Canarias, C. Via Lactea S/N, 38205 La Laguna, Tenerife, Spain
email: `hdeeg@iac.es`

Abstract. The Universal Transit Modeller (UTM) is a light-curve simulator for all kinds of transiting or eclipsing configurations between arbitrary numbers of several types of objects, which may be stars, planets, planetary moons, and planetary rings. Applications of UTM to date have been mainly in the generation of light-curves for the testing of detection algorithms. For the preparation of such test for the Corot Mission, a special version has been used to generate multicolour light-curves in Corot's passbands. A separate fitting program, UFIT (Universal Fitter) is part of the UTM distribution and may be used to derive best fits to light-curves for any set of continuously variable parameters. UTM/UFIT is written in IDL code and its source is released in the pubic domain under the GNU General Public License.

Keywords. methods: numerical, planetary systems, occultations, planets and satellites: general, planets: rings

1. Overview

A number of investigations related with transiting extra-solar planets depend on a reliable modelling of the light-curves that occulting system may exhibit. In this paper, we present the UTM package, which is short for Universal Transit Modeller. Its major strength is its flexibility which allows the modelling of synthetic light-curves of a very varied range of transiting configurations. These may range from simple planet-star transits to situations with planets orbiting close double stars; but UTM may also model the effects of planetary moons or rings onto a transit signature. Applications of UTM to date have been the fitting of model-curves to observed data (Deeg & Garrido 2001) and the generation of light-curves for the evaluation of detection algorithms (Aigrain & Favata 2002, Aigrain & Irwin 2004, Carpano, Aigrain & Favata 2003, Carpano & Fridlund 2008). For the Corot mission, UTM has been used in simulations of transits of Earth-like planets in the face of stellar microvariability (Bonomo & Lanza 2008) and for the preparation of several 'blind tests' to determine the mission's capability of transit detection, with detections from monocolour light-curves discussed in Moutou et al.(2005) and multicolour light-curves in Moutou et al.(2007), with a further test about the detectability of 'Super-Earths' currently being under way.

UTM is a light-curve modeller for all kinds of transiting or eclipsing configurations between an arbitrary number of different object-types, which may be stars, planets, planetary moons, and planetary rings. All input to UTM is controlled through setup files with a simple syntax. UTM calculates the orbital positions of the objects at each time-point and derives the resulting brightness. The simulations may be obtained for equidistant time-points or for time-points taken from an input file (e.g. from an existing light-curve). There are several levels of visualisations available, with animations of the bright objects of the system, of the positions of objects, and of the resultant light-curve (See Fig. 1), as well as a fast 'quiet mode'. Model light-curves may be produced in units of
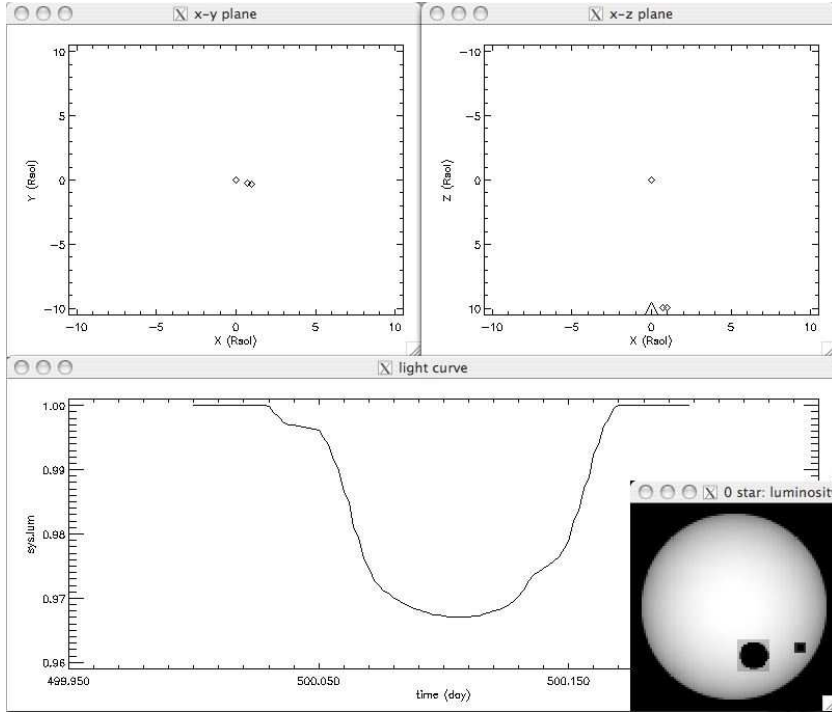
Figure 1. Visualisations shown by UTM for the case of a transit of a planet with a moon across a star: in the upper center and right are displays of objects positions in the x-y 'sky-plane' (as seen by observer) and in the x-z plane (as seen from top, with the triangle at x=0 indicating the direction to the observer). The three diamonds correspond to the three objects. The lower panel shows the output light-curve at the end of the simulation, were the offset between the lunar and the planetary transit is visible. The insert in the lower right shows the surface of the star in a moment during the planet-moon transit. While running UTM, these windows show a continuously updated animation.

the model-system's luminosity, as magnitude variations, or as relative flux variations, and are either saved directly; after the adding of random noise; or after having been added to an existing light-curve. UTM generates monocolor light-curves only, but a script to call UTM repeatedly for the simulation of multicolour data (such as generated in Moutou et al. 2007) is available from the author. A separate fitting program, UFIT (Universal Fitter) is part of the UTM distribution and may be used to derive best fits to observed data for any continuously variable parameter.

UTM is written in the Interactive Data Language (IDL[1]) using object-oriented programming techniques, which requires IDL version 5.1 or higher. The source of UTM and associated programs is freely distributed under the GNU Public License[2] and is available through the author's homepage[3].

2. How UTM works

All input to UTM is given in a setup file in which general parameters and the various bodies with their physical parameters and initial orbital positions are defined in a very

---

[1] IDL is a product of ITT Visual Information Solutions, http:www.rsinc.com/idl/

[2] The GNU Public License can be found at http:www.gnu.org/licenses/gpl.txt

[3] http://www.iac.es/galeria/hdeeg/hdeeghome.html; there follow link to UTM

```
#general parameters
sunit  Rsol    units for size
tunit  day     unit for time
lunit  Lsol    unit for luminosity
munit  Msol    unit for mass

wflag  0        1: write lightcurve to file, 0: dont't do it
tinit  0        initial time
tfin   10       final time
tinc   0.02     stepsize of time in units 'tunit'
gsize  12       sidelength of 2D array representing bodies' geometric form.
limblaw quad     limbdarkening law: lin or quad, default is lin

#below entries for the objects
#star
0type  star       type of object
0radi  1          radius of object
0mass  0.1        mass of obj
0period  0.0    period: if 0.0 then is fixed
0phase   0.0    posital phase [0-1]
0epoch   0.0    epoch
0inclin  90         inclination of orbit;0=face-on, 90=edge-on
0rdist   0        distance from coord. ctr
0lum     1        luminosity of object
0limbd 0.6        limb-darkening coeficcient, for stars only
0limbd2 0.3       2nd limb-darkening coeficcient, for quadrat law

#planet
1type  planet       type of object
1radi  0.2          radius of object
1mass  0.1          mass of obj
1period  10.0    period: if 0.0 then is fixed
1phase   -0.05   posital phase [0-1]
1epoch 10.4       epoch in tunit
1inclin  88         inclination of orbit;0=face-on, 90=edge-on
1rdist   5        distance from coord. ctr
1pa      15       clockwise angle from +y axis
```

Figure 2. A simple UTM setup file for a transiting star-planet system. Text behind #
symbols is ignored, as well as any text behind the second token in a line.

simple syntax. A minimalist setup-file for the simulation of a transit of a large planet
across a star is shown in Fig. 2. Several further example files may be found in the
software distribution of UTM. There are two general kinds of bodies: dark ones (planets,
moons, rings) and luminous ones (stars). For each defined body, a pixelized 2-dimensional
representation in a plane vertical to the observer (the x-y plane) is created. For all of them,
there is a representation of their opacity. For luminous objects (stars), a representation
of their surface brightness is created as well, with the linear, square and quadratic limb-
darkening laws (as defined in Claret 1998) being available. While running UTM, an
orbital simulation is generated and the 2D representations are being placed at their
corresponding positions into a 3D coordinate system, with the observer being at x,y,z =
$(0,0,\infty)$. The summed brightness of all luminous objects is then calculated, taking into
account any occultation that may occur. To do this, UTM checks for each bright object if
there is any overlap with another object positioned at larger z values. If there is one, the
overlapping parts of the luminous object's brightness array and of the occulter's opacity
array are regridded to the same scale, with the coarser (sub)array being the one that
is regridded. This procedure ensures high precision in the resultant brightness even for
transits of objects with large size differences, such as Earth-Sun transits. The precision
of UTM may also be controlled by defining the basic size of the pixelized representations
of the various objects. Both the precision and the level of visualisations have a strong
impact on the computation speed, and a quite mode without any terminal output is also
available.

3. The UFIT fitter

UFIT is a universal fitting program which can in principle fit any continuously variable parameter to any arbitrary univariate function. This function may be an analytic one or may be generated by some separate algorithm. In the UTM software, UFIT has been implemented to find best fits for *any* continuously variable parameter in a simulation made by UTM. To this end, UFIT reads a UTM-like setup-file that describes an initial model, in which additionally the input light-curve (to which models are being fitted) and the parameters to be fitted have to be specified. UFIT then employs the Levenberg-Marquardt algorithm, which calls UTM repeatedly with setup-files with slightly varying parameters. From the resultant sets of model light-curves, UFIT derives then partial derivatives of the goodness-of-fit (χ^2 value) against the parameters to be fitted; leading to the best fit in an iterative process.

4. Development of UTM

Since the initial version of UTM, written in early 1999, a number of improvements and additions have been made. These were always compatible with setup-files written for earlier versions, so that results obtained with previous versions can be reproduced reliably in any later one. In the 'traditional' use of UTM, brightness, time and size parameters (object radii, orbital halfaxis) can be given in any units; but this was limited to circular orbits. A recent addition has been a Keplerian code for the generation of either circular or elliptic orbits; this requires however the use of a specific set of units for masses, time and distances (old setup-files, however, still work the same). Current plans for further updates include the addition of star spots as a further class of objects and the possibility to save animations as 'movie-files'. Also foreseen is the inclusion of a further fitting kernel in the UFIT program and the possibility to define constraints among parameters that are being fitted. For the simulation of some more complex objects (e.g. contact binaries), their full 3D representation would be desirable, with a conversion to 2D 'on the fly', pending on their orientation relative to the observer. Since UTM is released to the public domain, anyone is free to supply further additions to this code; however, the author welcomes suggestions as well.

Acknowledgement

References

Aigrain, S., & Favata, F. 2002, *A&A*, 395, 625

Aigrain, S., & Irwin, M. 2004, *MNRAS*, 350, 331

Bonomo, A. S., & Lanza, A. F. 2008, *A&A*, 482, 341

Carpano, S., Aigrain, S., & Favata, F. 2003, *A&A*, 401, 743

Carpano, S., & Fridlund, M. 2008, *A&A*, 485, 607

Claret, A. 1998, *A&A*, 335, 647

Deeg, H. J., Garrido, R., & Claret, A. 2001, *New Astron.*, 6, 51

Moutou, C., Pont, F., Barge, P., Aigrain, S., Auvergne, M., Blouin, D. et al. 2005, *A&A*, 437, 355

Moutou, C., Aigrain, S., Almenara, J., Alonso, R., Auvergne, M. et al. 2007, *ASP-CS*, 366, 127