

Icarus

Icarus is a stellar binary light curve synthesis tool initially developed by Rene Breton while being a postdoctoral fellow at the University of Toronto in the research group of Marten van Kerkwijk.

Icarus provides a set of basic tools that:

1. Generates a star given some basic binary parameters
 - 1.1. Solves the gravitational potential equation
 - 1.2. Creates a discretized stellar grid
 - 1.3. Populates the stellar grid with physical parameters (temperature, surface gravity, etc.)
2. Evaluates the outgoing flux from the star given an observer's point of view (i.e. orbital phase and orbital orientation)

The code is compartmented in different layers:

1. The stellar surface solver
 - 1.1. The primitives generator of the discretized stellar grid: the code currently uses a triangular tessellation based the subdivision of an icosahedron. The primitives (vertices, face association, etc.) can be read from pre-calculated values stored in a file or dynamically generated using the external program "pygts" (<http://pygts.sourceforge.net/>), which is distributed separately and not essential to Icarus.
 - 1.2. The actual surface solver.
2. The flux calculator
 - 2.1. The actual flux calculator tools: integrated surface flux. Supported for different modes is already provided (spectroscopy, photometry, Doppler shifting, Doppler boosting).
 - 2.2. The flux calculator makes use of an atmosphere backend, which returns the specific intensities given a set of input parameters (temperature, surface gravity, velocity, etc.). The atmosphere backend can be anything (analytical blackbody, lookup table to an atmosphere model, etc.). The current backend reads data from NextGen atmosphere models (distributed separately).
3. The binary system super-class
 - 3.1. A super-class making use of the two above layers to treat with a "proper" binary, which sums the flux of each component and includes the calculation of eclipses, transits, partial occultations, etc.

The original aim of Icarus was to model the light curves (photometry and spectroscopy) of irradiated neutron star companions, hence the name Icarus (the Greek mythology hero who flew too close to the Sun and melting the wax off his wings). The flux calculator therefore supports the contribution of an external source of energy (from the other binary component) which contributes to increasing the dayside temperature of the modeled star.

Here are a short, non-exhaustive list of publications related to the binary light curve synthesis.

Breton et al., 2012, ApJL, 748, 115

Orosz, J. A., & Hauschildt, P. H. 2000, A&A, 364, 265

Hendry, P. D., & Mochnacki, S. W. 1992, ApJ, 388, 603

Installation

1. **Download the Icarus package from the github repository (<https://github.com/bretonr/Icarus>).**
 - 1.1. You may download the package as a zip/tarball file.
 - 1.2. Or you can clone the repository using git (preferred option) which will allow you to stay in sync with the latest package version. To do so, go to the disk location where you want to install the package and type:

```
>>> git clone git://github.com/bretonr/Icarus.git
```

To update it from the latest github version afterwards, you can type:

```
>>> git pull
```

2. Get yourself some atmosphere models or write your own atmosphere backend (e.g., to generate a blackbody function). I cannot be of much help here unfortunately. I might try to write a basic blackbody backend eventually but I do not have time for now.
3. Install the required packages (see Requirements section below).
4. Make sure to add the path to Icarus in your PYTHONPATH shell environment variable. If you use bash, you should add a line like that to your .bashrc file: `export PYTHONPATH=$PYTHONPATH:/home/breton/local/python/Icarus`. Here I have been assuming that you installed Icarus in `/home/breton/local/python/Icarus`. With that you should be able to load the Icarus module from anywhere. Make sure you test it: start a python prompt from your home directory and type "import Icarus". If the package does not load, you have made a mistake somewhere.
5. Do some cool light curves!

Requirements

I usually keep my packages up-to-date using Macport (on Mac) and Synaptic (on Ubuntu). Versions are provided for indicative purposes.

1. Python (<http://www.python.org/>; version >2.6, ideally 2.7).
2. Scipy (<http://scipy.org/>).
3. Numpy (<http://numpy.scipy.org/>).

Optional

1. Matplotlib (<http://matplotlib.org/>; version >1.1.0).
2. PyGTS to generate surface geodesic primitives instead of reading the pre-generated one (<http://pygts.sourceforge.net/>). Also useful for calculating occultations and transits in eclipsing binaries.

If you use it

If you intend to use the code, please cite the paper in which it was first introduced: R. P. Breton, S. A. Rappaport, M. H. van Kerkwijk, J. A. Carter, "KOI 1224, a Fourth Bloated Hot White Dwarf Companion Found With Kepler", 2012, ApJL, 748, 115.

Also, please provide a link to the Icarus webpage <https://github.com/bretonr/Icarus>.

The author (<superluminique + at + gmail.com>) would be happy to receive feedback, constructive comments, bug fixes, etc., from people using Icarus. Unfortunately, only very limited support can be provided due to the author's busy research schedule.

Acknowledgements

Note that the author would like to acknowledge the immense help of Marten van Kerkwijk, who contributed via frequent discussions and who also provided a Fortran program to synthesize photometric light curves of irradiated binaries, which Icarus initially aimed to reproduce.

License

Please note that this project is protected against a 3-clause BSD license. Please see the content of the folder "licenses" for more information.