MERCURY-T

Emeline Bolmont

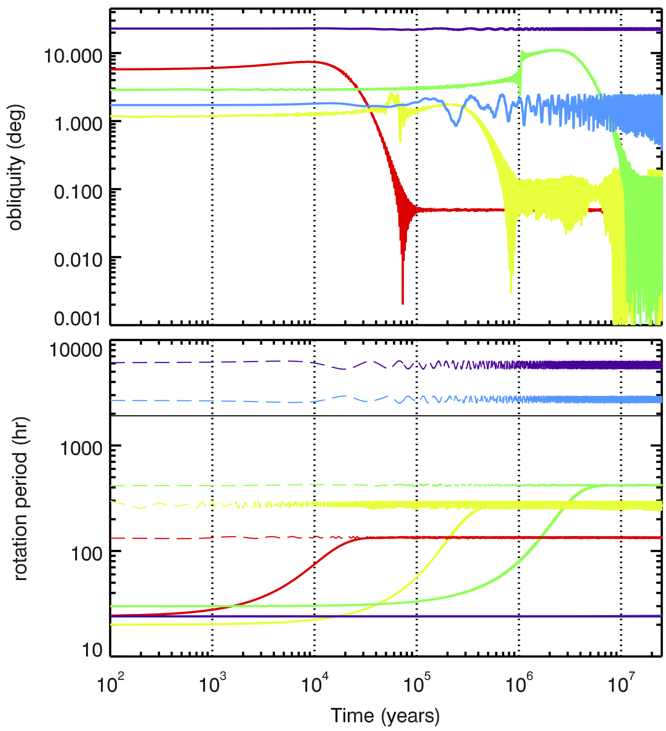

Figure 1: Evolution of the obliquities and rotation periods of the Kepler-62 system (Borucki et al. 2013).

Personal webpage and code:
`http://www.emelinebolmont.com/`
Paper:
`http://www.aanda.org/articles/aa/pdf/2015/11/aa25909-15.pdf`

Mercury-T is an *N*-body code which allows the user to compute the tidal evolution of planetary systems (Bolmont et al., 2015). It is based on the original *Mercury* code from Chambers (1999) and works exactly in the same way. If you don't know how *Mercury* works, I invite you to download the original one and read `mercury6.man`.

In order to download *Mercury-T*, please go there: http://www.emelinebolmont.com/research-interests and click on download!

In the directory, you should have all files needed to compile and execute the program[1]. Don't hesitate to contact me if any problem arises when trying to use it.

The notations used here can be found in Table 1. The dissipation σ measures how your star or planets dissipate heat from tidal deformation (e.g., see Hut, 1981). The dissipation σ_i of a body i depends on the product of its potential Love number of degree 2 $k_{2,i}$ and its time lag Δt_i:

$$\sigma_i = \frac{2}{3}\frac{\mathcal{G}k_{2,i}\Delta t_i}{R_i^5}, \tag{1}$$

where R_i^5 is the radius of this body.

Table 1: Notations

	Mass	Radius	Moment	Spin of inertia	Potential Love number deg 2	Fluid Love number	Dissipation factor
Star	$M_\star$	$R_\star$	$I_\star$	$\Omega_\star$	$k_{2,\star}$	$k_{2f,\star}$	$\sigma_\star$
Planet	M_p	R_p	I_p	Ω_p	$k_{2,\mathrm{p}}$	$k_{2f,\mathrm{p}}$	σ_p

[1]There are also some IDL scripts I use to plot the outputs of the code.

General comments and features:

1. MERCURY-T is a improvement of John Chambers's MERCURY code (Chambers, 1999). It uses the same input files: `big.in`, `small.in`, `param.in`. Note that for a good functioning of the code, start time has to be put to zero in `param.in`.

2. The version of MERCURY used here is an adaptation of the original fortran 77 code to fortran 90 done by Christophe Cossou (Laboratoire d'Astrophysique de Bordeaux, France). The pure MERCURY version of the code is available here:
https://mercury-90.googlecode.com/archive/master.tar.gz.

3. In the package, there are many python scripts. In particular, `Makefile.py`, which allows to compile the code. This file can be changed if you want to use another compiler as the default one (gfortran). Usually I use the compiler ifort, which leads to faster integration time.

4. Each module of the MERCURY code has been isolated by Christophe and the one containing user forces (such as tides, general relativity, etc) is called `user_module.f90`. The file called `tides_constant_GR.f90` contains the parameter files and the spin initial conditions (more of that later)

5. When MERCURY-T is running, it creates files to keep track of the evolution of the spin and angular momentum:

   (a) spin of the star $\Omega_\star$ in day^{-1} in `spins.out`:
   time (yrs), $\Omega_{\star,x}$, $\Omega_{\star,y}$ $\Omega_{\star,z}$, $\mathrm{R}_\star$ ($\mathrm{R}_\odot$), $\mathrm{I}_\star/(\mathrm{M}_\star \mathrm{R}_\star^2)$, $k_{2,\star}$, $\sigma_\star$

   (b) spin of planet i Ω_p in day^{-1} in `spinpi.out`:
   time (yrs), $\Omega_{\mathrm{p},x}$, $\Omega_{\mathrm{p},y}$, $\Omega_{\mathrm{p},z}$, $\mathrm{R_p}$ ($\mathrm{R}_\odot$), $\mathrm{I_p}/(\mathrm{M_p R_p^2})$

   (c) Angular momentum of planet i h_orb in AU^{2}day^{-1} in `horbi.out`:
   time (yrs), $h_{\mathrm{orb},x}$, $h_{\mathrm{orb},y}$, $h_{\mathrm{orb},z}$

   (d) The instantaneous energy loss of planet i due to tides $\mathrm{d}E/\mathrm{d}t$ in $\mathrm{M}_\odot$.AU^2.day^{-3} in `dEdti.out`:
   time (yrs), $\mathrm{d}E/\mathrm{d}t$

6. The z-axis is defined as the direction of the spin of the star. So the angle between the spin of the star and the angular momentum vector of planet i is the inclination of the orbit of planet i. The obliquity of planet i is defined as the angle between its angular momentum vector and the spin of planet i.

More specific comments:

1. In `user_module.f90`, you can find the expressions of the forces, the integration of the spins of planets and star, etc
   The integration of the spins is done in this file (twice a Runge-Kutta of order 5 on each half timestep), independently of the hybrid integration of the orbital elements.

2. In `tides_constant_GR.f90`, you can choose:

   (a) The effects you want to consider:
   Tides or not (tides), GR or not (GenRel), rotation-induced flattening or not (rot_flat)

   (b) The number of planets that are going to experience these effects: ntid

   (c) The nature of the host body:
   - Evolving brown dwarf of mass 0.01, 0.012, 0.015, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.072, 0.075 and 0.08 $M_\odot$: brown_dwarf = 1;
   - Evolving low-mass star of 0.1 $M_\odot$: M_dwarf = 1;
   - Evolving star of 1 $M_\odot$: Sun_like_star = 1;
   - Evolving Jupiter mass host body: Jupiter_host = 1;
   - Non-evolving star: Rscst = 1.
   The parameters implemented in *Mercury-T* are given in Table 2. For the last option, the user has to indicate all parameters (Love numbers, dissipation...).

   (d) t_init is the initial time: in the case of an evolving object, you can choose at which age of the host body you want to start calculating the evolution. Note that start time in `param.in` is not equivalent to t_init. start time should always be put to zero.

   (e) The initial values of the spins of star and planet if the variable called crash is put to 1. Originally this feature is to be able to restart a simulation which crashed (due to a server crash for example). More about the use of crash in the next section.

   (f) In case of crash, you put the last line of the spin output file in the corresponding table, and give t_crash the value of the time of crash. In case of an evolving body, you should give t_init the value you took initially (the age of the host body you considered) added to the time of the crash.

   (g) The planets' characteristics:

      i. its initial rotation period:
      pseudo_rot = 0 means the rotation period of the planet is given by Pp0,
      pseudo_rot = toto, means that the rotation period is toto $\times$ the pseudo synchronisation period),

      ii. its initial obliquity (oblp),

      iii. the nature of the planet (see also Table 3):
      - planet_type = 0: Earth-like planet, the radius is calculated from the mass of the planet following Fortney et al. (2007). I_p, $k_{2,\mathrm{p}}$, $k_{2f,\mathrm{p}}$ and $k_{2,\mathrm{p}}\Delta t_\mathrm{p}$ are assumed to be Earth's (see example for a 10 $M_\oplus$ planet in Table 3);
      - planet_type = 1: you need to indicate the value of R_p in radius_p, I_p, $k_{2,\mathrm{p}}$, $k_{2f,\mathrm{p}}$ and $k_{2,\mathrm{p}}\Delta t_\mathrm{p}$ are assumed to be Earth's;
      - planet_type = 2, you need to indicate the value of R_p in radius_p, the values of I_p, $k_{2,\mathrm{p}}$ and σ_p are fixed, $k_{2f,\mathrm{p}}$ is taken equal to $k_{2,\mathrm{p}}$;
      - planet_type = 3, you need to indicate the value of R_p and to give the values of $I_\mathrm{p}/(M_\mathrm{p}R_\mathrm{p}^2)$: rg2_what, $k_{2,\mathrm{p}}$: k2tp_what, $k_{2f,\mathrm{p}}$: k2fp_what, and $k_{2,\mathrm{p}}\Delta t_\mathrm{p}$: k2pdeltap_what.

(h) The host body's characteristics: dissstar allows you to multiply the nominal dissipation of the star by the number you give to dissstar;

3. How to resume a simulation:

    (a) You might need this in case you want to simulate the evolution for a longer time that you initially put in `param.in` or if the server crashed.

    (b) In `tides_constant_GR.f90`, you need to put crash = 1, adapt t_crash and t_init to your problem, and put the values of the spins of the star and planets.

    (c) If you want to integrate longer than you first wanted, you also need to change the final stop time in `param.dmp`.

    (d) Do not forget to re-compile before re-launching your simulation!

Table 2: Host body parameters implemented in *Mercury-T*

Type of host body	$M_\star$	$R_\star$	$k_{2,\star}$	$I_\star/(M_\star R_\star^2)$	$\sigma_\star$ ($\mathrm{g^{-1}cm^{-2}s^{-1}}$)
Jupiter	1 M_J	**evolving**	**evolving**	**evolving**	7.024×10^{-59}
BD	0.01–0.08 $M_\odot$	**evolving**	0.379–0.307	**evolving**	2.006×10^{-60}
dM	0.1 $M_\odot$	**evolving**	0.307	0.2	2.006×10^{-60}
Sun	1 $M_\odot$	**evolving**	0.03	0.059	4.992×10^{-66}

Table 3: Planetary parameters implemented in *Mercury-T*

Type of planet	M_p	R_p	$k_{2,p}$	$I_p/(M_p R_p^2)$	Dissipation factor ($\mathrm{g^{-1}cm^{-2}s^{-1}}$)
Earth-mass planet	1 $M_\oplus$	1 $R_\oplus$	0.305	0.3308	8.577×10^{-50}
Super-Earth	10 $M_\oplus$	1.8 $R_\oplus$	0.305	0.3308	4.691×10^{-51}
Jupiter	1 M_J	1 R_J	0.380	0.254	2.006×10^{-60}

References

Bolmont, E., Raymond, S. N., Leconte, J., Hersant, F., Correia, A. C. M. 2015. Mercury-T: A new code to study tidally evolving multi-planet systems. Applications to Kepler-62. Astronomy and Astrophysics 583, A116.

Chambers, J. E. 1999. A hybrid symplectic integrator that permits close encounters between massive bodies. Monthly Notices of the Royal Astronomical Society 304, 793-799.

Fortney, J. J., Marley, M. S., Barnes, J. W. 2007. Planetary Radii across Five Orders of Magnitude in Mass and Stellar Insolation: Application to Transits. The Astrophysical Journal 659, 1661-1672.

Hut, P. 1981. Tidal evolution in close binary systems. Astronomy and Astrophysics 99, 126-140.