TiDE documentation

Zsófia Kovács-Stermeczky

April 22, 2022

1 About the TiDE code

TiDE (TIdal Disruption Event) is a C++ code that computes the light curves or spectrum of tidal disruption events. The physics behind the code is described in the paper by Kovács-Stermeczky & Vinkó (2022).

2 Structure of TiDE and installation

The code is decomposed into four `hpp` and `cpp` file.

- `tde_assistant`: contains the used physical constants, some unit conversions, parameter file and argument management, Planck function and a print help function of the code.
- `tde_lcurve`: this is the principal file of the code. It contain the parameters, the two parts of the light curve routine (wind and disk part) and various classes for the computation of the light curves and spectra.
- `Trapezoidal_rule/trapezoidal`: contains the integrator that uses the trapezoidal rule.
- `tde_lcurve_run`: this `cpp` file contain the main function of the code.

To run the program one needs to compile all of these `cpp` files. For convenience, there is a `Makefile` that can be used to compile and link the different parts of the code together. IMPORTANT: ONE MUST USE AN AT LEAST C++17 COMPLIANT compiler. The code was tested with `g++ version 9.4.0` compiler with `-g -Wall -Wextra -std=c++17` flags, which meets this requirement (c++17 standard).

3 Settable program parameters

3.1 Settable parameters with parameter file or command-line arguments

Numerical values for most of the parameters of the code can be set either via a parameter file or via command-line argument. The list of these parameters is the following:

- M_6 (M6; -M6) [$10^6 M_\odot$]: Mass of the black hole in $10^6 M_\odot$ units;
- m_* (mstar, Mstar; -mstar, -Mstar) [$M_\odot$]: Mass of the star in $M_\odot$ units;
- x_* (rstar, Rstar; -rstar, -Rstar) [$R_\odot$]: Radius of the star in $R_\odot$ units. One may specify an independent value for the radius, or use the one of the built-in mass-radius relations for main sequence star (MS or ms) or white dwarf star (WD or wd);
- η (eta; -eta): radiative efficiency of the black hole, $0 < \eta < 1$;
- β (beta; -beta): penetration factor ($\beta = r_t/r_p$), i.e. the ratio of the tidal radius and the closest approach to the black hole. For physically self-consistent models one must use $\beta = 1$;
- f_{out} (fout; -fout): fractional mass of the debris that is supposed to leave the system via super-Eddington wind. $(1 - f_{out})$ part of the debris is assumed to form an accretion disk. $0 < f_{out} < 1$;

- f_v (fv; -fv): ratio between the velocity of the super-Eddington wind and the escape velocity at the base of the wind (at r_L distance from the black hole). Note that the code will check whether the wind velocity exceeds the speed of light, and sets it to the highest possible value if necessary;

- distance (d; -d) [m]: Distance of the event in meters. It scales the calculated luminosity with $1/(4 \cdot \pi \cdot d^2)$. If you use d=0, the code will set $1/(4 \cdot \pi \cdot d^2) = 1$;

- inclination (i; -i) [rad]: Inclination of the disk (modifies the disk luminosity with cos(i));

- N (N; -N): Number of concentric rings used to calculate the accretion disk during the numerical integration;

- t_{diff} (tdiff; -tdiff) [day]: Diffusion timescale in days;

- t_{start} (tstart; -tstart) [day]: the moment of the beginning of the light curve (used for computing the monochromatic light curve and the bolometric light curve);

- t_{end} (tend; -tend) [day]: the end time of the light curve (applied for all 3 possible light curves);

- t_{end} relative to t_{min} (tend_rt_tmin; -tend_rt_tmin) [day]: the difference between the start and the end time, i.e. $t_{end} - t_{min}$;

- dt (dt; -dt) [day]: Time step of the light curve (used for computing all 3 possible light curves);

- ν (nu; -nu) [Hz]: The frequency for computing the monochromatic light curve (used for computing the monochromatic light curve or monochromatic light curve with diffusion);

- ν_{start} (nustart; -nustart) [Hz]: The start of the frequency interval (used for computing spectra);

- ν_{end} (nuend; -nuend) [Hz]: The end of the frequency interval (used for computing spectra);

- $d\nu$ (dnu; -dnu) [Hz]: The frequency step of the spectra (used for computing spectra);

- t (time; -time) [day]: the epoch for the computed spectrum;

These previous parameters can also be set through the parameter file (parameters.txt) or through command-line arguments. In the parameter file one should use the name of the parameter without the "-" character. Each line contains one parameter specification, the value of the parameter should be separated from its name with a TABULATOR. The parameter file (parameters.txt) must exist (even if it is an empty file) in order to run the program.

For the command-line arguments one has to use a "-" character prior to the name of the parameter, and the value of the parameter must be separated from its name with a whitespace. WARNING! Command-line arguments will overwrite the parameter values specified in the parameter file.

3.2 Only command-line argument settings

There are some parameters that can set only with command-line arguments.

- built-in f_{out} (-bi_fout): using this argument instructs the code to use a time dependent, built-in f_{out} parameter;

- change $\dot{M}_p$ (-Mdot_Guillochon): use the accretion model of Guillochon & Ramirez-Ruiz (2013) for $\dot{M}_p$. Use -Mdot_Guillochon "4per3" or "5per3" for specifying the polytropic index as 4/3 (for stars) or 5/3 (for white dwarfs). If the polytropic index is not set, the code tries to guess it automatically: for a white dwarf star the polytropic index is 5per3, while it is 4per3 for all other cases;

- extras (-e, -extra): the program will create an extra output file (extras.dat) with some additional time dependent information, like photospheric radius, photospheric temperature, disk temperature at half of the disk, f_{out} (it will be time dependent only if use build-in f_{out} calculation) and t_{edge}. The code creates this file only when modeling a monochromatic light curve.

To create any output file, one should use at least one of the following arguments:

- light curve (-lc): the code calculates monochromatic light curve of a TDE event. The results will be stored in an output file (lcurve.dat);
- diffused light curve (-diffusion): the code calculates monochromatic TDE light curve with diffusion. The results are written into an output file (lcurve_diffusion.dat);
- bolometric light curve (-b, -bolometric): the code calculates the bolometric light curve, i.e. the integral of the monochromatic fluxes along wavelength (from 100 nm to 2500 nm, in 24 steps). The result will be stored in an output file (L_bolometric.dat);
- spectra (-s, -spectra): the code calculates the spectrum of a TDE event at a given time. The resulting spectrum is written into an output file (spectra.dat).

3.3 Default values of the program parameters

M_6	1
m_*	1
x_*	1
mass-radius relation	none
η	0.1
β	1
f_{out}	0.1
f_v	1
d	0
i	0
N	1000
t_{diff}	$t_{min} \cdot 0.06$
t_{start}	t_{min}
t_{end}	100
dt	0.5
ν	$6.3 \cdot 10^{14}$
ν_{start}	$3e13$
ν_{end}	$3e17$
$d\nu$	$2.9997e13$
t	t_{min}

Table 1: The values of the default parameter settings

4 Available models in `TiDE`

In the current version of `TiDE` code available four different output type.

The first one is the monochromatic light curve. The output file will start many comment lines starting with "#" character and will contain the used parameter settings. The output (lcurve.dat) file will consists of four column: time (in day units), total luminosity (in erg/s/Hz units), wind luminosity (erg/s/Hz) and disk luminosity (erg/s/Hz).

The second opportunity is to use diffusion during the calculation of monochromatic light curve. The output file in this case will be "lcurve_diffusion.dat" and also will start with (almost) the same comment lines as the previous one. This output file contain just two columns: first is the time (day) and second is the total luminosity (erg/s/Hz). Because during this method the code must execute integration every step, this method will grows a lot. The time step during the integration is 0.1 day.

Third available model is the bolometric luminosity. In that case the the code will integral the monochromatic fluxes along the wavelength (from 100 nm to 2500 nm, in 24 steps). The output file (L_bolometric) will start with comment lines and than will contain two columns: time (day) and total bolometric luminosity (erg/s). In the current version of `TiDE` you could not create bolometric luminosity with diffusion.

```
M6	1.
mstar	1.
rstar	ms
eta	0.1
beta	1.
fout	0.1
fv	1.
d	0.
i	0.
nu	6.3e+14
N	1000
dt	0.5
tend_rt_tmin	1000
```

Figure 1: An example for parameters.txt file. One line contain one parameter, the separator character must be a tabulator.

```
TiDE_PUBLIC$  ./TiDE -lc
M6 = 1  RS = 2.94961e+09
Mstar = 1       Rstar = 1
startype = Main sequence
fout = 0.1      fv = 1  eta = 0.1       beta = 1
d = 0   i = 0   nu = 6.3e+14    N = 1000
tmin = 41       tstart = 41     tend = 1041     dt = 0.5
rin = 8.84884e+09       rout = 1.41e+11
Mdot_peak = 1.87171e+23
```

Figure 2: To run the code and create a monochromatic light curve use -lc flag.

The last opportunity is to calculate a spectrum at a given time. The output file (spectra.dat) after the comment lines will contain four columns: ν frequency (Hz), λ wavelength (nm), luminosity of the wind part (erg/s/Hz) and luminosity of the disk part (erg/s/Hz).

5 Some examples

5.1 The Parameters file

An example for the parameter file (parameters.txt) is showed at Figure 1. One line contain one parameter and its value, the separate character must be a tabulator. The order of the lines are irrelevant. This file could be empty, but must exist at the same directory where you run the `TiDE` program (but the runnable file, `TiDE` could be anywhere else).

5.2 Create a monochromatic light curve

To create a monochromatic light curve you must use -lc flag. An example of this can see at Figure 2. In this case the parameter file was like in 1. During this process, the program also will write out some information from the run: the set parameters (M_6, M_{star} $[M_\odot]$, R_{star} $[R_\odot]$, the type of the star mass-radius relation, f_{out}, f_v, η, β, d $[m]$, i $[rad]$, ν $[Hz]$, N, t_{start} $[day]$, t_{end} $[day]$, dt $[day]$), and also will write out some values which can calculate from this fiducial set parameters (R_S $[m]$: the Schwarschild radius, t_{min} $[day]$, r_{in} $[m]$: the inner radius of the disk, r_{out} $[m]$: the outer radius of the disk, $\dot{M}_p$ $[kg/s]$: the peak of the accretion rate).

The first 20 lines of the output file of the previous run (lcurve.dat) could see in Figure 3. The unit of λ (lambda) parameter is nm.

```
TiDE_PUBLIC$  head -n 20 lcurve.dat
#M6 1
#m* 1
#x* 1
#eta 0.1
#beta 1
#fout 0.1
#fv 1
#d 0
#i 0
#nu 6.3e+14
#lambda 476.19
#N 1000
#
#time (day)     Ltotal (erg/s/Hz)       Lwind (erg/s/Hz)        Ldisk (erg/s/Hz)
41      1.04984e+27     1.03689e+27     1.29519e+25
41.5    1.01917e+27     1.00622e+27     1.29511e+25
42      9.89737e+26     9.76787e+26     1.29503e+25
42.5    9.61485e+26     9.48535e+26     1.29495e+25
43      9.34351e+26     9.21402e+26     1.29486e+25
43.5    9.08279e+26     8.95332e+26     1.29478e+25
```

Figure 3: The first 20 lines of the output lcurve.dat file.

```
TiDE_PUBLIC$  ./TiDE -diffusion -tdiff 7.2
TiDE_PUBLIC$  head -n 20 lcurve_diffusion.dat
#tdiff  7.2
#M6 1
#m* 1
#x* 1
#eta 0.1
#beta 1
#fout 0.1
#fv 1
#d 0
#i 0
#nu 6.3e+14
#lambda 476.19
#N 1000
#
#time (day)     Ltotal (erg/s/Hz)       Lwind (erg/s/Hz)        Ldisk (erg/s/Hz)
41.1    1.58872e+25
41.6    8.37954e+25
42.1    1.45149e+26
42.6    2.00468e+26
43.1    2.51525e+26
```

Figure 4: To create diffused monochromatic light curve use -diffusion flag. The output file (lcurve_diffusion.dat) first 20 lines almost the same as previous case.

5.3 Create diffused light curve

To create a diffused monochromatic light curve must use -diffusion flag (see Figure 4.). In this example we also set the diffusion time, but it is not obligatory: if you not set it, the code will use $t_{min} \cdot 0.06$ day. The output file (lcurve_diffusion.dat) almost the same as previous case.