TiDE documentation

Zsófia Kovács-Stermeczky

February 17, 2023

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 (2023) [2].

2 Structure of TiDE and installation

2.1 The structure

The code is decomposed into six `hpp` and `cpp` file. (These could be found at src directory)

- `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_parameters`: this is one of the principal file. It contains the parameter structures, the usable classes for different models (e.g. accretion rate, t_{min}, f_{out})
- `tde_lcurve`: this is the other principal file of the code. It contains the two parts of the light curve routine (wind and disk part) and various classes for the computation of the light curves and spectra.
- `Linear_interpol/linear_interpol`: contains the code for the linear interpolation method
- `Trapezoidal_rule/trapezoidal`: contains the integrator that uses the trapezoidal rule.
- `tde_lcurve_run`: this `cpp` file contains the main function of the code.

To run the program one needs to compile all of these `cpp` files. 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 -O2` flags, which meets this requirement (c++17 standard).

2.2 Installation with autoconf

The needed files for installation with autoconf can be found at github. For installation follow the instructions below:

- Download the files from github (git clone https://github.com/stermzsofi/TiDE.git)
- Create a directory for the installed program (eg.: mkdir ∼/TiDE)
- In the download directory use autoreconf (autoreconf --install) (if needed)
- Run the created configure file: (./configure --prefix ∼/TiDE)
- make
- make install

TiDE need two data files when somebody use L09 accretion rate model (see Section 4.2.3). These files will be placed at share subdirectory of the installation directory (eg. ∼/TiDE/share). To found this place, TiDE will use an environment variable, called TIDE_PATH. After the installation we need to define this variable as the installation directory (eg. : export TIDE_PATH=∼/TiDE).

For simplification recommended to add these two lines for ∼/bashrc file:

- export TIDE_PATH=∼/TiDE : This will create the needed environment variable after all restart
- export PATH=$PATH:∼/TiDE/bin : This will add the place of the runnable program to the PATH variable, simplify the run of TiDE

3 Settable program parameters

3.1 Settable parameters with a 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 arguments. 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.
- 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). IMPORTANT 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;
- β_d (beta_limit; -beta_limit): This parameter need for corrected t_{min} calculation (for more details see Section 4.4.2);
- 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)), face-on: i=0, edge-on: i=$\pi/2$;
- N (N; -N): Number of concentric rings used to calculate the accretion disk during the numerical integration;
- t_{diff} (tdiff; -tdiff) [day]: Photon diffusion timescale in days;
- t_{start} (tstart; -tstart) [day]: the moment of the beginning of the light curve. Because of physical reasons it could not be less than t_{min} time (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;
- ϵ_{rep}: the efficiency of the reprocessing;
- politrop: this is a string variable. It stores the polytropic index of the star. Two available values are: "4per3" and "5per3";

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 exists (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

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).

IMPORTANT NOTE: current version was primarily test for -lc and -diffusion. The bolometric light curve and spectra may not work correctly in this version.

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

- help function (-h; -help): print the help function of TiDE
- extra information (-e; -extras): need to use together with lc. 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), current accretion rate
- set f_{out} calculation method (-bi_fout): using this argument instructs the code to use a time dependent, built-in f_{out} parameter (more details: Section 4.5.2);
- set $\dot{M}_{fb}$ calculation method (-Mdotfb_classic; -Mdotfb_L09_const; -Mdotfb_L09) : specify the accretion rate model. For more details see Section 4.2;
- set $\dot{M}_p$ calculation method (-Mdotp_classic; -Mdotp_L09_const; -Mdotp_GR13): set the model of the peak accretion rate. You should set this only if use classical or constant L09 accretion rate model. More details are in Section 4.3
- set t_{min} calculation method (-tmin_classic; -tmin_with_rp; -tmin_from_GR13) More details are in Section 4.4;
- set rph calculation method (-rph, -rtr, -rtr_full, -rc) More details are in **??**

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
β_d	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}
ϵ_{rep}	0
politrop	4per3

Table 1: The values of the default parameter settings

4 Available models in `TiDE`

4.1 The types of the output

In the current version of `TiDE` four different output types are available.

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

The second possibility is to use diffusion during the calculation of a 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 contains just two columns: the time (in days) and the total luminosity (in erg/s/Hz). Because during this method the code must execute an integration at every step, the running time will be longer. For the integration it uses a quadrupole trapezoidal rule with $\epsilon = 0.01$ [4].

The third available model is the bolometric luminosity. In that case the code will integrate the monochromatic fluxes along the wavelength axis (from 100 nm to 2500 nm) in 24 steps. The output file (L_bolometric) will start with comment lines followed by two columns: time (in days) and total bolometric luminosity (in erg/s). In the current version of `TiDE` one cannot create a bolometric light curve with diffusion.

The last possibility 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).

4.2 Fallback accretion rate types and time dependence

4.2.1 Classical model (default) (-Mdotfb_classic)

In the classical case the fallback accretion rate can be calculated as:

$$\dot{M}_{fb} = \dot{M}_p \left(\frac{t}{t_{min}}\right)^{n_\infty}, \tag{1}$$

,where $n_\infty = -5/3$ in the classical case. The flag is: -Mdotfb_classic

4.2.2 L09 constant ρ (-Mdotfb_L09_const)

In Lodato et. al (2009) (therefore: L09) [3], if assuming a constant $\rho(x)$, we get the fallback accretion rate as:

$$\dot{M}_{fb}(t) = \dot{M}_p \left[1 - \left(\frac{t}{t_{min}}\right)^{-4/3}\right] \cdot \left(\frac{t}{t_{min}}\right)^{n_\infty}, \tag{2}$$

,where $n_\infty = -5/3$ in classical case again. The flag is: -Mdotfb_L09_const

4.2.3 L09 result (-Mdotfb_L09)

If we use the general form of L09 [3], then the fallback rate can be calculated as:

$$\dot{M}_{fb} = \frac{4\pi}{3} \frac{R_*}{t_{min}} \left(\frac{t}{t_{min}}\right)^{-5/3} \int\limits_{x(t)}^{R_*} \rho(R) R dR \tag{3}$$

where $x(t) = R_*(t/t_{min})^{-2/3}$. To calculate this equation, one more information about the structure of the star. Here we use a polytropic stellar model. The first step is to replace $\rho(R) - R$ parameters with $\theta(\xi) - \xi$ dimensionless parameters. ξ is the dimensionless radius:

$$R = \xi \cdot \left(\frac{4\pi G \rho_c^{1-1/n}}{K(n+1)}\right)^{-1/2}, \tag{4}$$

where ρ_c and K are constants and n is the polytropic index. ρ can be calculated from:

$$\rho = \rho_c \theta^n \tag{5}$$

We need to know the $\theta(\xi)$ values, so we solve the Lane-Emden equation with a second order Runge-Kutta method:

$$\frac{1}{\xi^2} \frac{d}{d\xi} \left(\xi^2 \frac{d\theta}{d\xi}\right) + \theta^n = 0 \tag{6}$$

The boundary conditions are:

$$\xi = 0: \qquad \theta = 1 \qquad \frac{d\theta}{d\xi} = 0 \tag{7}$$

$$\xi = \xi_{max}: \qquad \theta = 0 \tag{8}$$

If we rewrite Equation 3. with ξ and θ we get:

$$\dot{M}_{fb} = \frac{4\pi}{3} \frac{R_*}{t_{min}} \left(\frac{t}{t_{min}}\right)^{-5/3} \rho_c C^2 \int\limits_{x'(t)}^{\xi_{max}} \theta^n(\xi) \xi d\xi \tag{9}$$

where

$$C = \left(\frac{4\pi \rho_c^{1-1/n}}{K(n+1)}\right)^{-1/2} \tag{10}$$

and $x'(t) = \xi_{max}(t/t_{min})^{-2/3}$. The value of the integral will be the same for any given polytropic index, thus, it can be calculated in advance and write out into an output file. Later, to get the current value of the integral at a given t time and parameters, we can easily look it up from the table using linear interpolation. We also need to specify the ρ_c and K values. If we know the mass (M_*) and the radius (R_*) of the disrupted star, then:

$$R_* = \left(\frac{K(n+1)}{4\pi G \rho_c^{1-1/n}} \right)^{1/2} \xi_{max} \tag{11}$$

$$M_* = 4\pi\rho_c \left(\frac{K(n+1)}{4\pi G \rho_c^{1-1/n}} \right)^{3/2} \int\limits_0^{\xi_{max}} \theta^n \xi^2 d\xi \tag{12}$$

From Equation 11:

$$K = \frac{R_*^2}{\xi_{max}^2} \frac{1}{(n+1)} 4\pi G \rho_c^{1-1/n} \tag{13}$$

and from Equation 12 and 13:

$$\rho_c = \frac{M_*}{4\pi} \frac{\xi_{max}^3}{R_*} \left(\int\limits_0^{\xi_{max}} \theta^n \xi^2 d\xi \right)^{-1} \tag{14}$$

If we apply the polytropic indices for different types of stars (n=3 for MS stars and n=3/2 for WD stars) we are able to calculate the expressions above. The current version of TiDE contains this two indices, the tables (share/table_n3.dat and share/table_n3per2.dat) were produced with an assistant code which works as above.

To determine the moment of the peak accretion rate, we need to get the derivative of Eq. 3 against time, and find the value when it is equal to zero. For that we use another variable, $y = t/t_{min}$, and solve the above equation for y. Then we will be able to determine the position of the peak relative to t_{min}. We introduce the function $F(x)$ as:

$$F(x) = - \int\limits_{\xi_{max}}^{x} \theta^n \xi d\xi \Rightarrow \frac{dF}{dx} = - \frac{d}{dx} \int\limits_{\xi_{max}}^{x} \theta^n \xi d\xi = -\theta^n(x) x \tag{15}$$

$$\frac{dF(x)}{dy} = \frac{dF}{dx}\frac{dx}{dy} \tag{16}$$

Using these and realizing that the maximum must not be at $x = \xi_{max}$ position (and from it, $F(x) \neq 0$), we get:

$$-\frac{5}{2} y^{4/3} + \xi_{max}^2 \frac{\theta^n(x)}{F(x)} = 0 \tag{17}$$

If:

$$G(y) = \xi_{max}^2 \frac{\theta^n(x)}{F(x)} \tag{18}$$

(where $x = \xi_{max} y^{-2/3}$), then the values of G(y) can found via linear interpolation. The sought y position can be found with a Newthon-Raphson method. Based on this, we find:

$$t_{peak} = 5.77 \cdot t_{min} \qquad (n = 3) \tag{19}$$

and

$$t_{peak} = 2.66 \cdot t_{min} \qquad (n = 3/2) \tag{20}$$

4.2.4 Classical, constant n_∞ (default value: -5/3)

At the first two accretion rate model (classical and constant ρ L09 case) we can specify n_∞ parameter too. It is possible to set the value of n_∞ as a constant, if one writes its value after the Mdotfb flag. Examples: -Mdotfb_classic -2.1; -Mdotfb_L09_const -2.1

4.2.5 n_∞ from GR13

Guillochon & Ramirez-Ruiz (2013) (therefore: GR13) [1] presents n_∞ as a function of β. From their paper $D_\gamma = n_\infty$, and the values are:

$$D_{5/3} = \frac{-0.93653 + 11.109\beta - 38.161\beta^2 + 50.418\beta^3 - 22.965\beta^4}{1 - 8.6394\beta + 26.012\beta^2 - 32.383\beta^3 + 14.350\beta^4} \tag{21}$$

$$D_{4/3} = \frac{-2.7332 + 6.9465\beta - 3.2743\beta^2 - 0.84659\beta^3 + 0.56254\beta^4}{1 - 2.3585\beta + 0.47593\beta^2 + 0.96280\beta^3 - 0.37996\beta^4} \tag{22}$$

To use this calculation one may write GR13 after Mdotfb flag.
Example: -Mdotfb_classic GR13

4.3 The peak of the accretion rate (only use if accretion rate model is classical or L09 with constant ρ)

The peak of the accretion rate ($\dot{M}_p$) is directly appear on the fallback accretion rate if you use classical or constant L09 model. In total L09 case (Section 4.2.3) the peak is just a property of the accretion rate. If you use total L09 accretion rate model, you must not set the peak accretion rate model, or the written out $\dot{M}_p$ value may will not represent the real value of it!

4.3.1 Classical result (default)

The classical result of the $\dot{M}_p$ peak of the accretion rate is:

$$\dot{M}_p = \frac{1}{3}\frac{M_*}{t_{min}} \tag{23}$$

Use the flag: -Mdotp_classic

4.3.2 L09 constant ρ

One may also calculate $\dot{M}_p$ from the L09 paper [3], and assume that the density profile, $\rho(x)$, is a constant. In this case $\dot{M}_p$ will be:

$$\dot{M}_p = \frac{1}{2}\frac{M_*}{t_{min}} \tag{24}$$

Warning! If you use -Mdotfb_L09_const accretion rate model, the peak of the accretion rate will not set automatically to this version. To select this, use the flag: -Mdotp_L09_const

4.3.3 GR13 model

GR13 [1] computed many hydrodynamical models and provided the following equation:

$$\dot{M}_p = A_\gamma M_6^{-1/2} m_*^2\, x_*^{-3/2}\ \mathrm{M}_\odot/\mathrm{yr}, \tag{25}$$

where γ is the polytropic exponent and A_γ is:

$$A_{5/3} = \exp\left[\frac{10 - 17\beta + 6.0\beta^2}{1 - 0.47\beta - 4.5\beta^2}\right] \tag{26}$$

$$A_{4/3} = \exp\left[\frac{27 - 28\beta + 3.9\beta^2}{1 - 3.3\beta - 1.4\beta^2}\right] \tag{27}$$

For this model use the -Mdotp_GR13 flag.

4.4 t_{min} time

4.4.1 β independent, classical result (default)

The classical result of t_{min} time is:

$$t_{min} = \frac{\pi}{\sqrt{2}} \left(\frac{r_t}{R_*} \right)^{3/2} \sqrt{\frac{r_t^3}{GM}} \tag{28}$$

The flag of this: -tmin_classic

4.4.2 β independent, corrected version

TiDE also contain a special case of the previous, initial result, we called corrected t_{min}. The explanation is the following: many studies showed, that $\beta = 1$ is not enough for full disruption of the star. If β reach a limit value (called β_d), then the total desintegration will happen. We define $r_t' = r_t/\beta_d$ and our new t_{min} definition is the following:

$$t_{min} = \frac{\pi}{\sqrt{2}} \left(\frac{r_t'}{R_*} \right)^{3/2} \sqrt{\frac{r_t'^3}{GM}} = \frac{\pi}{\sqrt{2}} \left(\frac{(r_t/\beta_d)}{R_*} \right)^{3/2} \sqrt{\frac{(r_t/\beta_d)^3}{GM}} \tag{29}$$

If $\beta_d = 1$ it restores the classical result. For this version of t_{min} we do not need any new flag, just change β_d value with -beta_limit flag.

4.4.3 β dependent

In many previous papers the classical result appeared with r_p instead of r_t. In this case, t_{min} time will be dependent on β:

$$t_{min} = \frac{\pi}{\sqrt{2}} \left(\frac{r_p}{R_*} \right)^{3/2} \sqrt{\frac{r_p^3}{GM}} \tag{30}$$

The flag is: -tmin_with_rp

4.4.4 t_{min} from GR13 t_{peak} values (only if accretion rate model is L09)

If we use the L09 $\dot{M}_{fb}$ model, we know that the time of the peak of the accretion rate depends linearly on t_{min}: the constant multiplier, C, is a function of the polytropic index. From this,

$$t_{min} = t_{peak}/C \tag{31}$$

. Alternatively, one can use t_{peak} from GR13:

$$t_{peak} = B_\gamma \cdot M_6^{1/2} \cdot m_*^{-1} \cdot x_*^{3/2} yr \tag{32}$$

where B_γ can be calculated from:

$$B_{4/3} = \frac{-0.38670 + 0.57291\sqrt{\beta} - 0.31231\beta}{1 - 1.2744\sqrt{\beta} - 0.90053\beta} \tag{33}$$

$$B_{5/3} = \frac{-0.30908 + 1.1804\sqrt{\beta} - 1.1764\beta}{1 + 1.3089\sqrt{\beta} - 4.1940\beta} \tag{34}$$

The flag is: -tmin_from_GR13, but it will result valid values when the accretion rate model is L09.

4.5 f_{out} models

4.5.1 Constant, time independent (default)

You can use a constant, time independent f_{out} parameter. One can set its value in the parameter file (fout value) or with -fout flag.

4.5.2 Time dependent f_{out} (-bi_fout)

One may also use a time dependent f_{out} value. In this case the code calculates the f_{out} parameter from the following equation:

$$f_{out} = \frac{2}{\pi}\arctan\left[\frac{1}{7.5}\left(\frac{\dot{M}_{fb}}{\dot{M}_{Edd}} - 1\right)\right]. \tag{35}$$

After $\dot{M}_{fb}$ is less than $\dot{M}_{Edd}$ this go negative, which are physically not valid. To prevent this, TiDE minimize time dependent f_{out} value as 0.1. To apply this time dependent parameter, use -bi_fout flag.

4.6 Photospheric, trapping and color radius

4.6.1 Photospheric radius (default) (-rph)

The definition of the photospheric radius is

$$\kappa\rho(r_{\rm ph})r_{\rm ph} = 1 \tag{36}$$

In this case the luminosity of the wind component will calculate with r_{ph}. For more details about the calculation of the luminosity see Section 2.2 and 3.5 at [2].

4.6.2 (basic) Trapping radius (-rtr)

The definition of the trapping radius is the distance where the photon diffusion time is equivalent with the dynamical time. The basic criterion that at this distance

$$\kappa\rho(r_{\rm tr})r_{\rm tr} = \frac{c}{v_{\rm wind}}. \tag{37}$$

In this case TiDE use $r_{\rm tr}$ to calculate the wind luminosity.

4.6.3 (full) Trapping radius (-rtr_full)

The full solution of $r_{\rm tr}$ is

$$\frac{r_{\rm tr}}{r_{\rm L}} = 1 + \frac{\kappa_{\rm es} f_{\rm out} \dot{M}_{\rm fb}}{4\pi r_{\rm L} c}\frac{(r_{\rm w} - r_{\rm tr})^2}{r_{\rm w}^2}, \tag{38}$$

where $r_{\rm w} = r_{\rm L} + (t - t_{\rm min})v_{\rm w}$ is the outer boundary of the wind. This is a quadratic equation, and because of physical reasons, we will need the solution with the minus sign.

4.6.4 Color radius (-rc)

The color radius is the location where the last absorption of the photons occurs. The value of $r_{\rm c}$ can be calculated using the following criterion:

$$\tau_{\rm eff}(r_{\rm c}) \equiv \int_{r_{\rm c}}^{\infty} \kappa_{\rm eff}\rho dr = 1, \tag{39}$$

(more details see Section 3.5 at [2]). In this case TiDE check the relation between r_c and $r_{\rm tr}$ (Equation 38) every time step. If $r_c > r_{\rm tr}$, it will use r_c for wind luminosity calculations, else $r_{\rm tr}$.

5 Error and warning massages

There is a list of the possible error and warning massages and the explanation of their reasons.

5.1 ERROR massages

- Unknown politrop parameter: ... : the current version of TiDE could use two different politrop: 4per3 or 5per3
- t_start is less than tmin : because of physical reasons, the light curve (and diffused light curve too) could start at t_{min} time. If it would be less than it, it reasons many problems with the resulted luminosities (eg. nan or inf luminosities). If you would like to start your light curve from t_{min} time, you sholdn't set tstart value, it will automatically start from t_{min}.
- Unknown ninf calculation: ... : the list of possible ninf methods are in Section 4.2
- There is no type star with id: ... : Possible star ids are: MS; ms; WD; wd
- Couldn't open parameters.txt file : parameters.txt must exist (also if it is empty) at the same directory when you run TiDE.
- Error with parameter file line: ... :
- Error with argument: ... : if you do not use a number after your flag
- Unknown argument: ... : your flag is not exist
- ... cannot be opened : if couldn't open the table files. The problem may with TIDE_PATH environment variable.
- TIDE_PATH environment variable not found : this error massage could appear only if -Mdotfb_L09 was applied. This environment variable must exist, see Section 2.2

5.2 WARNING massages

- S is nan : This warning massage could appear if you use diffusion model. The case of it is a numerical problem. The equation of diffusion is:

$$L(t^*) = \frac{1}{t_{diff}} e^{-\frac{t^*}{t_{diff}}} \int_0^{t^*} e^{\frac{t'}{t_{diff}}} L_{inp}(t') dt' \tag{40}$$

  The problem is with the exponential multipliers: one of it is too large, one is too small if the time is too large. Mathematically it is not problem, but numerically it will result nan during the integration method. With L'Hôpital's rule can show, that if time goes to infinity, the diffused luminosity will goes to $L_{inp}(t)$. To resolve this numerical problem, if TiDE found the first nan value, it will use the model without diffusion to calculate the luminosities.

6 Some examples

6.1 The Parameters file

An example for the parameter file (parameters.txt) is showed in Figure 1. Each line contains one parameter and its value, the separation character must be a tabulator. The order of the lines is irrelevant. This file can be empty, but must exist at the same directory where you run the `TiDE` code.

6.2 Creating a monochromatic light curve

To create a monochromatic light curve you must use -lc flag. An example of this can be seen in Figure 2. In this case the parameter file was like in Figure 1. During this process, the program will write out some information about the actual run: the chosen parameter values (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 will also output some values that can be calculate from this fiducial set of parameters (R_S $[m]$: the Schwarzschild 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) can be seen in Figure 3. The unit of λ (lambda) parameter is nm.

```
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. Each line contains one parameter, the separator character must be a tabulator.

```
TiDE_PUBLIC$  TiDE -lc
Parameters of the black hole:
        Mass: 1 10^6 M_sun      Schwarzschild radius: 2.94961e+09 m
Parameters of the star:
        Mass: 1 M_sun   radius: 1 R_sun
        Used mass-radius relation: Main sequence
        Politrop: 4per3 n: 3
        K: 3.84109e+09  rho_c: 76201.3
Used constant fout parameter: 0.1
fv = 1  eta = 0.1       beta = 1
d = 0   i = 0   nu = 6.3e+14 Hz N = 1000
Used classical tmin calculation with rt distance. tmin = 41.0076 day
tstart = 41.0076 day    tend = 1041.01 day      dt = 0.5 day
rin = 8.84884e+09 m     rout = 1.39268e+11 m
Mdotfb calculated as classical power function.
Mdot_peak = 1.87136e+23 kg/s
vwind = 1.34164e+08 m/s
```

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

```
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)
41.0076 6.97788e+26     6.85077e+26     1.27109e+25
41.5076 6.77243e+26     6.64533e+26     1.27101e+25
42.0076 6.57541e+26     6.44832e+26     1.27093e+25
42.5076 6.38637e+26     6.25929e+26     1.27085e+25
43.0076 6.20492e+26     6.07784e+26     1.27077e+25
43.5076 6.03066e+26     5.90359e+26     1.27069e+25
```

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

```
TiDE_PUBLIC$  TiDE -diffusion -tdiff 2.4
TiDE_PUBLIC$  TiDE -diffusion -tdiff 2.4 -tend 3000
WARNING: S is nan!
TiDE_PUBLIC$  head -n 20 lcurve_diffusion.dat
#tdiff  2.4
#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)
41.1076 2.83884e+25
41.6076 1.51498e+26
42.1076 2.47706e+26
42.6076 3.22223e+26
43.1076 3.79274e+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.

6.3 Creating a 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. You also could see, that 'S is nan!' warning massage is appear if t_{end} is enough large. The output file (lcurve_diffusion.dat) almost the same as previous case.

References

[1] James Guillochon and Enrico Ramirez-Ruiz. Hydrodynamical Simulations to Determine the Feeding Rate of Black Holes by the Tidal Disruption of Stars: The Importance of the Impact Parameter and Stellar Structure. , 767(1):25, April 2013.

[2] Zsófia V. Kovács-Stermeczky and József Vinkó. Comparison of different tidal disruption event light curve models with tide, a new modular open source code, 2023.

[3] G. Lodato, A. R. King, and J. E. Pringle. Stellar disruption by a supermassive black hole: is the light curve really proportional to t$^{-5/3}$? , 392(1):332–340, January 2009.

[4] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007.