



Rate Estimation Program for Tidal Disruption Events Manual

Christian H. Hannah

March 3, 2025

This document serves as supplemental documentation for REPTiDE and should be consulted alongside the code paper Hannah et al. (2024b).

Contents

1 Installation	1
2 Usage Examples	1
3 Execution	2
3.1 Inputs	2
3.2 Outputs	3
3.3 Single vs Batch Rates	4
3.4 1-D Surface Brightness to 3-D Stellar Density	4
4 Attribution	4
5 Dependencies	5

1 Installation

To install REPTiDE, use the following:

```
pip install git+https://github.com/christianhannah/reptide.git
```

2 Usage Examples

When downloaded, an additional directory named “Examples” can be found within the package directory containing three example scripts:

- `reptide_sb_to_density_example.py`: This script demonstrates how to use REPTiDE to convert a 1D surface brightness profile into a 3D stellar mass density profile via Abel inversion (as in Hannah et al., 2024a).

- `reptide_single_example_analytic.py`: This script demonstrates a single TDE rate calculation for a density profile described by Equation 1.
- `reptide_single_example_discrete.py`: This script demonstrates a single TDE rate calculation for a discrete density profile (i.e. tabulated x and y data).

3 Execution

3.1 Inputs

The primary inputs required for this calculation are the central black hole mass (M_\bullet) and the radial 3D stellar density profile $\rho(r)$. There are two versions of the code that depend on the definition of the stellar density profile; this can be either “Discrete” or “Analytic.” In the discrete version, the density profile is supplied as a discrete, non-parametric set of data points. The analytic version uses a density profile that follows a power law with Sersic decay:

$$\rho(r) = \begin{cases} \rho_{5\text{pc}} \left(\frac{r}{5\text{ pc}}\right)^{-\gamma} & , r < r_d \\ I_{\text{eff}} \times \exp \left[-b_n * \left(\frac{r}{r_{\text{eff}}}\right)^{(1/n)-1} \right] & , r \geq r_d. \end{cases} \quad (1)$$

Here, $\rho_{5\text{pc}}$ is the stellar density at a radius of 5 pc, γ is the power law slope, and r_d is the radius where the exponential decay starts. For the Sersic decay component, I_{eff} is defined such that the density matches at r_d given the user-defined sersic index (n) and effective radius (r_{eff}).

Below are the input parameters related to the density profile based on the version.

Discrete:

- `rads`: Radii for the density measurements [m]
- `dens`: 3D stellar densities [kg/m^3]
- `sflag`: Boolean to indicate fixed inner density power law slope.
- `s`: inner power law density slope applied if `sflag` = True.

Analytic:

- `slope`: Power law slope of the density profile.
- `rho_5pc`: 3D stellar density at a radius of 5 pc [kg/m^3].
- `decay_start`: Radius to begin Sersic profile density decay [m].
- `reff`: Width of the Sersic profile decay [m].
- `n`: (optional; default=1) Sersic index for the decay.
- `smooth`: (optional; default=0.1) Parameter describing the smoothness of the broken power-law when applying a Bahcall-Wolf cusp.

The rest of the parameters for REPTiDE are the same regardless of version:

- `name`: Galaxy identifier - used for personal object identification.
- `M_BH`: Black hole mass [kg]
- `bw_cusp`: (optional; default=False) Boolean to indicate the application of a Bachall-Wolf cusp to the density profile at small radii.
- `bw_rad`: (optional; default=1e-300) Radius to begin the Bachall-Wolf cusp if `bw_cusp`=True.

- M_{min} : (optional; default=0.08) Minimum stellar mass to consider in the present-day mass function [M_\odot].
- M_{max} : (optional; default=1.0) Maximum stellar mass to consider in the present-day mass function [M_\odot].
- $quiet$: (optional; default=True) Boolean to toggle status printing during calculation.

3.2 Outputs

In addition to the version-dependent input parameters mentioned above, REPTiDE will return the following values:

- $name$: Galaxy identifier.
- M_{BH} : Black hole mass [kg].
- TDE_rate_solar : TDE rate for monochromatic population of $1 M_\odot$ stars.
- TDE_rate_full : TDE rate for a realistic Kroupa present-day mass function (PDMF).
- orb_ens : Array of orbital energies used for the calculations [m^2/s^2].
- DF : Distribution function over orbital energy.
- q : Dimensionless diffusivity parameter over orbital energy.
- $masses$: Array of stellar masses used in the PDMF [kg].
- $LC_contributions_per_mass$: TDE rate per stellar mass in the PDMF.
- $LC_flux_per_mass$: Loss-cone flux curves for each stellar mass in the PDMF.
- ψ_rads : Radii over which the gravitational potential is computed [m].
- ψ_tot : Total potential [m^2/s^2].
- ψ_bh : Potential from the black hole [m^2/s^2].
- ψ_enc : Potential from the enclosed stellar mass [m^2/s^2].
- ψ_ext : Potential from the external mass [m^2/s^2].
- $DF_integrals$: Integrals as a function of orbital energy used to compute the distribution function.
- $mu_integrals$: Integrals as a function of orbital energy used to compute dimensionless diffusivity parameters.
- $periods$: Orbital periods as a function of orbital energy.
- mu : Angular momentum diffusion coefficients as a function of orbital energy.
- R_LC : Squared ratio of the loss-cone angular momentum to the angular momentum of a circular orbit as a function of orbital energy.
- bw_cusp : Boolean parameter indicating if a Bachall-Wolf cusp was applied.
- bw_rad : Radius at which the Bachall-Wolf cusp was applied if $bw_cusp = True$
- EHS : Boolean parameter used to indicate the inclusion of event horizon TDE rate suppression.

3.3 Single vs Batch Rates

After defining the necessary input data arrays (outlined in Section 3.1), the user must use one of the following functions to build the input data table:

- `in_table = _reptide_.create_discrete_input_table(args)`
- `in_table = _reptide_.create_analytic_input_table(args)`

These functions return an AstroPy table object that is passed directly to REPTiDE’s primary function, “`_reptide_.run_reptide(in_table, analytic, n_energies, EHS)`,” which performs the rate calculation. Here, *analytic* is a boolean parameter indicating the version of REPTiDE to use: *analytic*=True uses the analytic version and vice versa, *n_energies* specifies the number of orbital energies used in the calculation (optional; default=1000), and *EHS* specifies whether or not to include event horizon suppression of the TDE rate in calculating the “full” TDE rate (optional; default=False).

If the user would like to compute TDE rates for a batch of galaxies utilizing multiprocessing, there is a wrapper script imported with the module titled “`reptide_batch_run.py`.” In this case, the user must define their input parameter arrays at the beginning of this file under the “User Defined Section.” In addition to the standard inputs discussed in Section 3.1, the user must also define the *analytic*, *n_energies*, and *EHS* parameters discussed above, the number of cores desired for multiprocessing (*cpu_count*; defaults to the maximum number available), and the path/filename for the output “.fits” file (*out_fn*).

3.4 1-D Surface Brightness to 3-D Stellar Density

REPTiDE is also equipped with a function designed to convert a supplied 1-D surface brightness profile into a 3-D stellar mass density profile under the assumption of spherical symmetry; “`_reptide_.SB_to_Density([inputs])`”. The details of this procedure are outlined in Hannah et al. (2024a), while the inputs and outputs are described below.

Inputs:

- *rads*: Radii corresponding to the surface brightness measurements [arcsec].
- *SBs*: Surface brightness measurements [L_{\odot}/pc^2].
- *M_L*: Mass-to-light ratio for the system.
- *distance*: Distance to the system in pc.
- *ax_ratio*: (optional; default=1) Axial ratio for the system (b/a).
- *inc*: (default=90°) Inclination angle for the system [deg].

Outputs:

- *rad_pc*: Radii in pc.
- *densities*: 3-D stellar mass densities [M_{\odot}/pc^3]
- *gausses*: Individual Gaussians used for the MGE fit to the SB profile.
- *summed_gauss*: Summed Gaussian fit to the SB profile.

4 Attribution

If you use this software in your work, please cite Hannah et al. (2024b).

5 Dependencies

1. NumPy (v1.26.4)
2. SciPy (v1.13.0)
3. Astropy (v5.3.4)
4. tqdm (v4.66.2)
5. mgefit (v5.0.15)

The exact version of each package may not be required, but these are the versions the program was developed with. Note that pip should install these requirements automatically.

References

Hannah, C. H., Seth, A. C., Stone, N. C., & van Velzen, S. 2024a, , 168, 137,
doi: <http://doi.org/10.3847/1538-3881/ad630a>

Hannah, C. H., Stone, N. C., Seth, A. C., & van Velzen, S. 2024b.
<https://arxiv.org/abs/2412.19935>
<https://arxiv.org/abs/2412.19935>

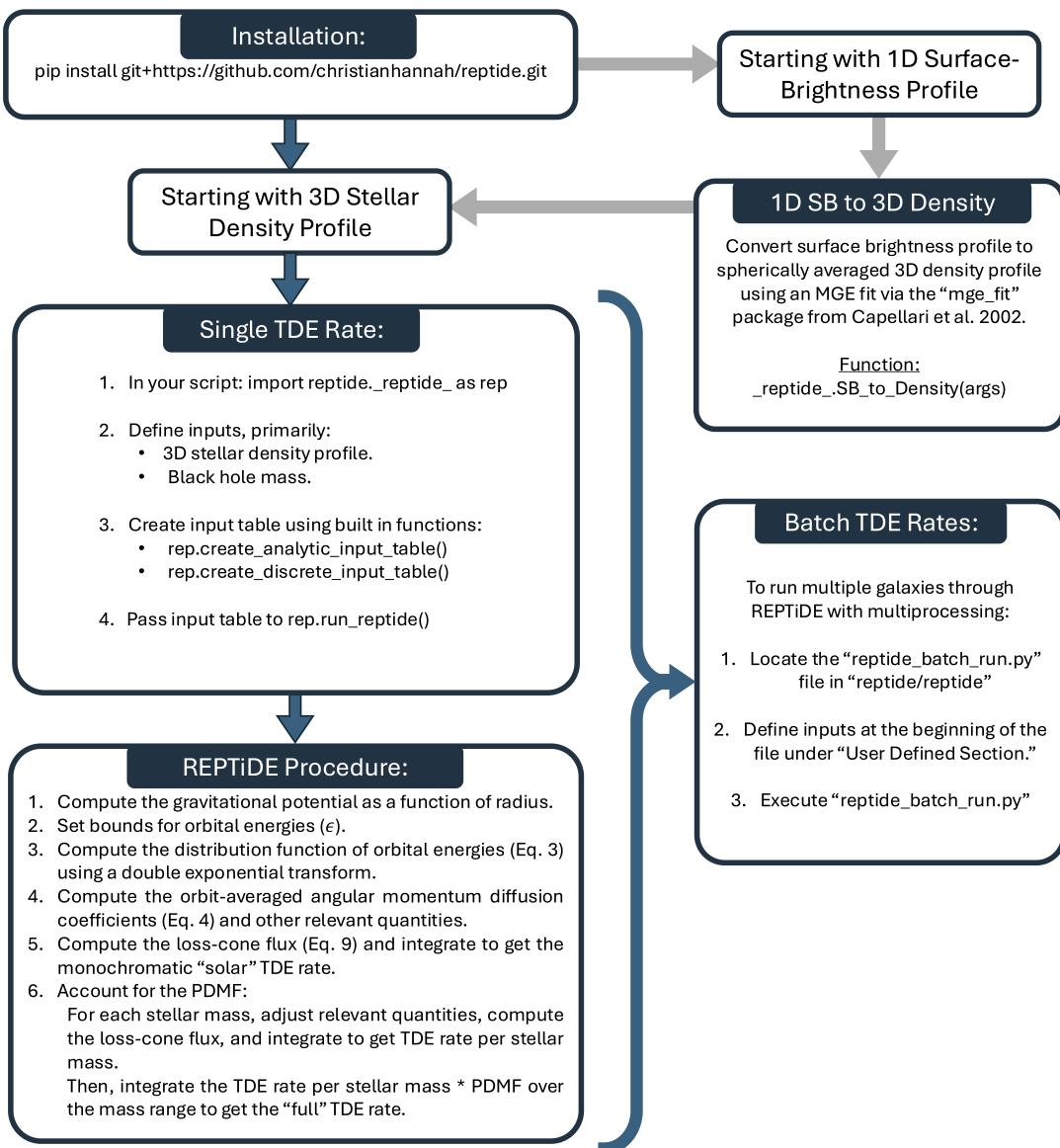


Figure 1: This figure serves to graphically explain the use of REPTiDE.