BSE/Popbin documentation

Marc van der Sluys

February 15, 2025

Contents

BSE/Popbin is the binary-stellar-evolution/population-sythesis code developed by Jarrod Hurley et al.

(2000, 2002). The original Fortran code can be downloaded from his website. At Utrecht Universtity, we have been using this code since 2001. This document attempts to document the usage and internals of the "Utrecht branch" of BSE/Popbin.

Both the `bse` and `popbin` programs take `binary.in` as an input file when starting. The parameters of the initial binaries can be generated with a pseudo-random-number generator, or read from a list in `binaries.in`. Where `bse` generates `binary.dat` with detailed stellar-evolution tracks (small time steps), `popbin` generates `binaries.out`, with the main evolutionary phases of a population of binaries, one line per binary.

Specific searches can be stored in `DWDs*.out` (zero-age double white dwarfs) and `NSBHs*.csv` (binaries solely consisting of neutron stars and/or black holes).

1 Input file `binary.in`

1.1 Binary input list

Namelist containing input parameters for BSE/Popbin.

1.1.1 Initial conditions

`mass0` = **5.0 3.5** Masses of the primary and secondary in Mo

`Tphysf` = **13000.0** Desired final age in Myr

`tb` = **17.0** Orbital period in days

`Kstar` = **1 1** Stellar type: 0 or 1 on the ZAMS, use -Kstar to start in the desired evolved state Kstar

`z` = **0.02** Metallicity

`ecc` = **0.0** Orbital eccentricity

`Nbin` = **1000** Number of binaries to compute with Popbin (0 = use number in binaries.in)

`initial_binary_seed` = **1** Random-number seed for generation of initial binaries (>0). -1: use binaries.in file. Can be overruled on cli.

1.1.2 Input and output files

`binaries_in_file` = **'binaries.in'** Name of the input binaries file for Popbin (binaries.in)

`binaries_out_file` = **'binaries.out'** Name of the output binaries file for Popbin (binaries.out). Overruled if `initial_binary_seed` >0.

1.1.3 Time steps

`pts1` = **0.05** Determines timestep on MS (0.05)

`pts2` = **0.01** Determines timestep on GB, CHeB, AGB, HeGB (0.01)

`pts3` = **0.02** Determines timestep on HG, HeMS (0.02)

1.1.4 Stellar winds

`Neta` = **0.5** Reimers mass-loss coefficient (0.5)

`Bwind` = **0.0** Binary enhanced mass loss parameter (inactive for single)

`Hewind` = **1.0** Helium-star mass-loss factor (1.0)

`beta` = **0.125** Wind velocity factor: proportional to v_{wind}^2 (0.125) `xi` = **1.0** Wind accretion efficiency factor (1.0)

`acc2` = **1.5** Bondi-Hoyle wind accretion factor (1.5)

`wind_multiplier` = **1.0** Scale the default wind mass-loss rates (1.0)

1.1.5 Mass transfer, stability and conservation

`Qinv_fastmt` = **1.0** Increase the mass-transfer rate by this factor if $M_d > M_a$ (1.0)

`HGqcrit` = **4.0** Critical mass ratio for stable/unstable mass transfer in the Hertzsprung gap (4.0)

`use_zetas` = **T** Use zetas to determine whether mass transfer is stable (F)

`use_beta_MT` = **1** Mass-accretion efficiency to use:

1. default BSE (variable β_{MT}); `beta_mt_min/_max` ignored
2. fixed, random β_{MT}, chosen randomly for each binary from (`beta_mt_min`, `beta_mt_max`) (except when MT=thermal: variable and (0,1))
3. fixed, random β_{MT}, from (`beta_mt_min`, `beta_mt_max`), also when MT=thermal
4. 4-variable β_{MT}; `beta_mt_min/_max` ignored - how different from 1)?

`beta_MT_min` = **0.0** Minimum mass-accretion efficiency to use (1.0 = conservative)

`beta_MT_max` = **1.0** Maximum mass-accretion efficiency to use (1.0 = conservative)

`epsnov` = **0.001** Fraction of accreted matter retained in nova eruptions (0.001)

`eddfac` = **1.0** Eddington limit factor for mass transfer (1.0)

1.1.6 Angular-momentum loss and tides

`MBfac` = **1.0** Scale factor for magnetic braking (1.0)

`gamma_rl` = **-1.0** Specific angular momentum for mass lost during RLOF: -2.d0: secondary wind; -1.d0 (primary wind; default); >0.d0: `gamma_rl` × orbital AM

`rlof_AMtransfer` = **T** Allow AM transfer between spin and orbit during mass transfer (T)

`Tflag` = **1** Activate tidal circularisation (1)

1.1.7 Common envelopes

`CEflag` = **0** CE prescription used: 0: default alpha-CE, 3: Webbink/de Kool CE (0)

`alpha_ce` = **1.0** Common-envelope efficiency parameter (1.0)

`lambda_be` = **0.5** Envelope binding-energy factor (for CE evolution; 0.5). If <0: use $|\lambda_{be}|$; if >=0: compute; if `LVK_Ebind_GB` is true, use $E_{int} + |\lambda_{be}| \times E_{recom}$ (0-1, or e.g. -0.5)

`LVK_Ebind_GB` = **F** Compute envelope binding energy on the GBs using Loveridge, van der Sluys & Kalogera (2011) (F)

`gamma_ee` = **-1.75** γ_{EE} parameter, for AM-balanced envelope ejection using γ prescription; used instead of some alpha CE's if `gamma_ee` >0 (-1.75)

`allow_dri` = **T** Allow Darwin-Riemann instabilities if the donor is a giant and $3\times J_{donor} > J_{orb}$ (T)

1.1.8 WDs, NSs, BHs and SNe

`IFflag` = **0** Use WD IFMR of HPE, 1995, MNRAS, 272, 800 (0)

`WDcool` = **1** WD-cooling model: 0: Mestel, 1: Hurley & Shara (1)

`NSflag` = **1** Take NS/BH mass from Belczynski et al. 2002, ApJ, 572, 407 (1)

`mxNS` = **3.0** Maximum NS mass; a "NS" with M>mxNS is a BH (recommendation: use 1.8 if NSflag=0; 3.0 if NSflag=1)

`BHflag` = **0** Allows velocity kick at BH formation (0)

`sigma` = **190.0** Dispersion in the Maxwellian for the SN kick speed in km/s (190)

`idum` = **-29768** Random-number seed used by the kick routine and beta_{mt}

1.2 Binary input evolved

Namelist used when one or both Kstars is <0 - start with an evolved binary.

`Tphys` = **0.0** Something with starting age ???

`aj` = **0.0 0.0** Something with starting age ??? (epoch = thpys - aj)

`mass` = **5.0 3.5** Initial masses in Mo

`Ospin` = **0.0 0.0** Initial spin. Set to 0.0 to auto-determine, set to >0.001 to start with small spin

2 Input file `binaries.in`

This file provides a list of initial (ZAMS) binaries to evolve with `popbin`. The lines can be read unformatted.

The first line is a single number `Nbin` which is the number of binaries in the list.

The following `Nbin` lines form a block with the initial-binary parameters with the following columns:

1. `m1`: initial mass of star 1 (Mo)
2. `m2`: initial mass of star 2 (Mo)
3. `Pi`: initial orbital period (days)
4. `ecc`: initial eccentricity (0-1)
5. `z`: metallicity
6. `Tmax`: maximum time to evolve binary for (Myr)
7. `beta_mt0`: the accretion efficiency (0-1)

3 Stellar types

The stellar types are known as `kw1,2` *or* `kstar1,2`. *Note: starts at 0!*

- 0. Low-mass MS star (deeply/fully convective)
- 1. Main-sequence star
- 2. Hertzsprung gap
- 3. First/Red Giant branch (RGB)
- 4. Core helium burning

- 5. First AGB
- 6. Second AGB
- 7. Naked helium MS
- 8. Naked helium HG
- 9. Naked helium GB
- 10. Helium WD
- 11. Carbon/oxygen WD
- 12. Oxygen/neon WD
- 13. Neutron star
- 14. Black hole
- 15. No remnant

4 Variables

4.1 BCM

In the variable `bcm(row, column)`, the row indicates the time-step number and the column the variable. In total, there are 50,000 rows and 34 columns. A model is saved every `dtp` time, hardcoded at `0` (every timestep). Variables 16–29 for star 2 correspond to 2–15 for star 1.

The columns are:

1. $t_{phys,f}$: current age (Myr)
2. kstar: stellar type for star 1 (int)
3. initial mass of star 1 (constant!)
4. current mass of star 1
5. ${}^{10}\log L$ for star 1
6. ${}^{10}\log R$ for star 1
7. ${}^{10}\log T_{\rm eff}$ for star 1
8. M_c: core mass of star 1
9. R_c: core radius of star 1
10. M_{env}: mass of convective envelope of star 1
11. R_{env}: radius of convective envelope of star 1
12. epoch of star 1 (gradually goes down (becomes negative!), then jumps up)
13. o_{spin} of star 1
14. $\Delta M_1 = \dfrac{-\Delta M_{\rm MT,1} - \Delta M_{\rm wind,1}}{\Delta t}$
15. R_1/R_{Rl1}
16. kstar: stellar type for star 2 (int)
17. initial mass of star 2 (constant!)
18. current mass of star 2

19. ${}^{10}\log L$ for star 2
20. ${}^{10}\log R$ for star 2
21. ${}^{10}\log T_{\rm eff}$ for star 2
22. $\rm M_c$: core mass of star 2
23. $\rm R_c$: core radius of star 2
24. $\rm M_{env}$: mass of convective envelope of star 2
25. $\rm R_{env}$: radius of convective envelope of star 2
26. epoch of star 2 (gradually goes down (becomes negative!), then jumps up)
27. $\rm o_{spin}$ of star 2
28. $\Delta M_2 \dfrac{+\Delta M_{\rm MT,2} - \Delta M_{\rm wind,2}}{\Delta t}$
29. $\rm R_2/R_{Rl2}$
30. $\rm t_b$: orbital period
31. sep: orbital separation
32. ecc: orbital eccentricity
33. Unused ???
34. Unused ???

4.2 BPP

In the variable `bpp(row, column)`*, the row indicates the evolutionary phase and the column the variable. In total, there are 80 rows and 17 columns. Data are stored at each change in evolutionary phase.*

The columns are:

1. $\rm t_{phys,f}$: final epoch of interest (time to evolve binary for?)
2. Mass of star 1
3. Mass of star 2
4. kstar1: stellar type for star 1
5. kstar2: stellar type for star 2
6. sep: orbital separation
7. ecc: orbital eccentricity
8. $\rm R_1/R_{Rl1}$
9. $\rm R_2/R_{Rl2}$
10. Integer to indicate binary state (e.g. coalesced, dissolved, . . . ; see below)
11. $\rm M_c$ of star 1
12. $\rm M_c$ of star 2
13. Luminosity of star 1
14. Luminosity of star 2
15. $T_{\rm eff1}$

16. T_{eff2}
17. Darwin-Riemann instability for this star? (0/1 = no/yes) — could have used 10?

4.2.1 Changes in evolutionary phase

The numbers stored in the variable `bpp(:,10)` *are used to indicate the cause of a change in evolutionary phase ("Δ kw").*

1. Initial binary
2. Type change — details unknown?
3. RLOF begins
4. RLOF ends
5. Contact binary
6. Coalescence
7. Common envelope
8. GNTAGE: determine the age of a giant (accreting WD turns into giant?)
9. Nothing remains
10. Age of universe reached
11. Binary disrupted by supernova or tides
12. Symbiotic phase begins
13. Symbiotic phase ends
14. Blue straggler forms

4.3 SCM

Array containing stellar parameters: `scm(50000,14)`

1. age
2. stellar type (kw)
3. initial mass
4. current mass
5. log10(L)
6. log10(r)
7. log10(Teff)
8. core mass
9. core radius
10. envelope mass
11. envelope radius
12. epoch
13. spin

4.4 SPP

Array containing stellar-evolution phases: `spp(20,3)`

1. age
2. stellar typpe (kw)
3. mass

5 Ouptut file `binary.dat`

Data for detailed stellar-evolution tracks from BSE.

1. `age`: age (Myr)
2. `kw1`: evolutionary stage of star 1 (-)
3. `kw2`: evolutionary stage of star 2 (-)
4. `M1`: mass of star 1 (Mo?)
5. `M2`: mass of star 2 (Mo?)
6. `Mc1`: core mass of star 1 (Mo?)
7. `Mc2`: core mass of star 2 (Mo?)
8. `log`: R1 log radius of star 1 (Ro?)
9. `log`: R2 log radius of star 2 (Ro?)
10. `R1/Rrl1`: Roche-lobe-filling factor of star 1 (-)
11. `R2/Rlr2`: Roche-lobe-filling factor of star 2 (-)
12. `log`: L1 log luminosity of star 1 (Lo?)
13. `log`: L2 log luminosity of star 2 (Lo?)
14. `Ospin1`: spin frequency of star 1? (?)
15. `Ospin2`: spin frequency of star 2? (?)
16. `dM1/dt`: mass gain/loss of star 1? (Mo/yr?)
17. `dM2/dt`: mass gain/loss of star 2? (Mo/yr?)
18. `a_orb`: orbital separation (Ro?)
19. `ecc`: orbital excentricity (-)

6 Output file `binaries.out`

6.1 Current

See subroutine `save_popbin_output()`.

6.1.1 Initial (ZAMS) binary

1. `m1`:
2. `m2`:
3. `porb`:

Formatting: `F7.3,F8.3,ES12.4`

(4. $beta_{MT}$ added?)

6.1.2 Final binary

1. `kw1`: evolutionary stage of star 1 (-)
2. `kw2`: evolutionary stage of star 2 (-)
3. `age`: age (Myr)
4. `M1`: mass of star 1 (Mo?)
5. `M2`: mass of star 2 (Mo?)
6. `P_orb`: orbital period (d?)
7. `L1`: luminosity of star 1 (Lo?)
8. `L2`: luminosity of star 2 (Lo?)
9. `Teff1`: effective temperature of star 1 (K?)
10. `Teff2`: effective temperature of star 2 (K?)
11. `sum(count_mt)`: total number of mass-transfer phases (-)
12. `rli`: =1 if *1 fills RL; =2 for *2 (=3 for both?) (-)
13. `count_stable(1)`: Number of stable mass-transfer phases initiated by star 1 (-)
14. `count_ce(1)`: Number of common envelopes initiated by star 1 (-)
15. `count_stable(2)`: Number of stable mass-transfer phases initiated by star 2 (-)
16. `count_ce(2)`: Number of common envelopes initiated by star 2 (-)

Formatting: `I6,I3, ES12.4,F8.3,F8.3, ES12.4,ES12.4,ES12.4,ES12.4,ES12.4, I7,I2,I3,I2,I3,I2,`

6.1.3 Blocks for (intermediate) mass-transfer episodes

1. `rli`: =1 if *1 fills RL; =2 for *2 (=3 for both?)
2. `mttype`: 0: binary coalesces/is disrupted, 1: Stable MT, 2: CE, 3: MT (stable) never ends
3. `kw1`: at the start of RLOF
4. `kw2`: at the start of RLOF
5. `age`: at the start of RLOF
6. `m1`: at the start of RLOF
7. `m2`: at the start of RLOF
8. `porb`: at the start of RLOF
9. `kw1`: at the end of RLOF
10. `kw2`: at the end of RLOF
11. `age`: at the end of RLOF
12. `Mass`: of star 1 at the end of RLOF
13. `Mass`: of star 2 at the end of RLOF
14. `P_orb`: at the end of RLOF

Formatting: `I8,I3, I3,I3, ES12.4, F8.3,F8.3, ES12.4, I5,I3,ES12.4,F8.3,F8.3,ES12.4`

If mttype ≠ 1,2, the last known values are printed in the last six columns instead of the values for the end of RLOF.

6.2 Original

See subroutine `save_popbin_output_original()`.

1. `tmax`: time `(F10.1)`
2. `kw`: type of star 1 `(I3)`
3. `kw2`: type of star 2 `(I3)`
4. `mx`: final mass of star 1 `(F8.3)`
5. `mx2`: final mass of star 2 `(F8.3)`
6. `eccx`: final orbital eccentricity `(F8.3)`
7. `tbx`: final orbital period `(ES14.6)`

7 Output file `DWDs.out`

Output file containing specific info on zero-age double white dwarfs. May be called `DWDs_123456.out`, where the number is the random-number seed.

- TODO: Change to csv

7.1 Columns

1. `M_1,i`: ZAMS mass of star 1 ($M_\odot$) `(F8.4)`
2. `M_2,i`: ZAMS mass of star 2 ($M_\odot$) `(F9.4)`
3. `P_orb,i`: Initial orbital period (d) `(ES12.4)`
4. `beta_mt,0`: Mass-transfer conservation factor `(F8.4)`
5. `kw1`: Type of white dwarf 1 `(I6)`
6. `kw2`: Type of white dwarf 2 `(I3)`
7. `max(twd1,twd2)`: Time since ZAMS when DWD was formed `(ES12.4)`
8. `M_WD1`: `Mass of WD 1 ~(F8.3)`
9. `M_WD2`: Mass of WD 2 `(F8.3)`
10. `P_orb,DWD`: DWD orbital period `(ES13.4)`
11. `L_WD1`: Luminosity of WD 1 ($L_\odot$) `(ES13.4)`
12. `L_WD2`: Luminosity of WD 2 ($L_\odot$) `(ES12.4)`
13. `t_WD1`: Cooling age of WD 1 (years?) `(ES13.4)`
14. `t_WD2`: Cooling age of WD 1 (years?) `(ES12.4)`
15. `gamma_EE,eff`: Effective $\gamma_{\rm EE}$ during main mass-transfer episode `(ES13.4)`
16. `sum(count_MTs)`: Total number of mass-transfer episodes `(I6)`
17. `0`: Unused `(I2)`
18. `count_stable_MTs(1)`: Number of mass-transfer episodes with star 1 as the donor `(I3)`

19. `count_CEs(1)`: Number of common envelopes with star 1 as the donor `(I2)`
20. `count_stable_MTs(2)`: Number of mass-transfer episodes with star 2 as the donor `(I3)`
21. `count_CEs(2)`: Number of common envelopes with star 2 as the donor `(I2)`

8 PlotBSE

`PlotBSE` is a Fortran program to analyse the output in `binaries.out` and `DWDs.out`. It is currently being replaced with `bse-postprocess`, written in Python. The columns below describe the contents of the arrays `dat1()` and `idat()`, not `dati()`!

8.1 ZAMS

- 1: `M1`:
- 2: `M2`:
- 3: `P`:
- 4: `beta_MT`:
- 8: `q1i`:
- 9: `q2i`:

8.2 Evolution

- 11: `kw1f`:
- 12: `kw2f`:
- 13: `# MT phases`:
- 14: `Rli`:
- 15: `# st.  MT *1`:
- 16: `# CEs *1`:
- 17: `# st.  MT *2`:
- 18: `# CEs *2`:
- 19: `1st MT = st/CE`:
- 20: `norml/dCE/revers.`:

8.3 Final

- 21: `age`:
- 22: `M1`:
- 23: `M2`:
- 24: `Porb`:
- 25: `L1`:
- 26: `L2`:
- 27: `Teff1 (der)`:
- 28: `Teff2 (der)`:

- 29: t_WD1:
- 30: t_WD2:

8.4 Final, derived

- 31: Mtot:
- 32: Ltot:
- 33: L2/L1:
- 34: qL:
- 35: q1:
- 36: q2:
- 37: gamma_EE:
- 38: Teff,tot:
- 39: Delta-tau:
- 40: a_orb:
- 41: age_WD1:
- 42: age_WD2:
- 43: Teff_WD1_CM:
- 44: Teff_WD2_CM:
- 45: Teff1_BSE/CM:
- 46: Teff2_BSE/CM:
- 47: Dl spectrum?:
- 48: Vmag: