LISA BPS Common Output format

Draft Proposal v. 0.1.

Luca Graziani[1] & Ruggero Valli[2],
and the LISA Synthetic UCB Catalogue Group
April 24, 2023.

1 Introduction

This document describes a common format for the output of binary population synthesis (BPS) codes agreed upon by the members of the LISA Synthetic UCB Catalogue Group. The goal of the format is to provide a common reference framework to describe the evolution of a single, isolated binary system or a population of isolated binaries all formed at assigned gas metallicity.

1.1 Motivation

To provide a common output format across BPSs participating the LISA collaboration in order to:

- Simplify data processing, especially for large datasets.
- Easily and safely compare different BPS predictions.
- Minimize errors due to unit conversion, or avoid misinterpretation or similar physical quantities.
- Provide a reliable agreement of scientific data interpretation.

1.2 Leading Principles

The proposed common format is designed to adhere to the following general principles:

Progressive: the common format is designed to be forward/backward-compatible with its extensions and additions. This means that the common format is aimed at being capable to change in time in order to comply with

[1]`www.phys.uniroma1.it`, `luca.graziani@uniroma1.it`
[2]`www.mpa-garching.mpg.de`, `ruvalli@mpa-garching.mpg.de`

the evolution of the BPS codes currently in the collaboration (and their future modifications), as well as to include new BPS codes.

Flexible: the common format can accommodate both specific and generic information, as well as missing data.

Unambiguous: the common format defines value, organization and labelling of all required data and its compliance as precisely as possible in order to minimize differences in its practical implementation.

Easy to implement: the proposed format, in its initial version, aims at being close to the data formats currently adopted by existing BPS.

To satisfy the previous requirements the format adopts a hierarchical organization of its values in stratified layers dubbed as L0 (common core variables), L1 and L2. Their definition will be provided below and could change and vary across progressive versions of common format. The present format is labelled as version 0.1. As a rule, a file containing data defined as compliant with the common format should be provided by specifying the reference format version and the implemented layer (e.g. "This file contains data in the LISA common format v0.1 at level L2").

2 Description of the format

The common format can be practically implemented in both hdf5 and ASCII csv file formats. In this preliminary draft we describe its csv implementation. A ASCII csv file conforming to the present format is required to respect the following rules :

- The file contains `3+NROWS` lines.
- Each line contains a sequence of fields separated by commas `','`.
- Quantities of type double, are represented in exponential notation with a number of decimal digits between 2 and 5 (e.g. 1.23e10 or 1.23456e10), with the exception of the constants in the header that can contain as many digits as necessary.
- The first 3 lines contain **the file Header**
- The following `NROWS` lines contain the **the Table of Values**.

2.1 The Header

The following variables will be used in the present document:

- `NHFIELDS` is an integer number greater or equal to the number of header fields defined in 1
- `NCOLUMNS` is an integer number greater or equal to the number of columns in level L0, L1 or L2 defined in 2, depending on the adopted standard level.

The Header is composed of 3 lines. The first 2 lines contains `NHFIELDS` field names or labels. In particular, the first line contains unique names (labels) of header fields common to the entire data-set stored in the file (e.g. version of the adopted standard, common level provided in the file, BPS code label, etc), while the second line contains their specific values, appearing in the same order as the labels they correspond to. The required header fields are described in table 1. The third line of the header contains the names (labels) of the variables describing each binary system evolution according to the specific implemented level (either defined in L0, L1 or L2, see Table 2). Their physical corresponding values are listed in columns with same order in the Table of Values.

An example of a Header compliant with the standard is the following:

```
version,level,bps,Z,NSYS,NROWS,Ggrav,clight,Msun,Rsun,Lsun,Zsun,AU
1.0,L0,ComBinE,0.01,1000,25678,6.67430e-8,2.99792458e10,1.98847e33
    ,6.957e10,3.828e33,0.0134,1.495978707e13
```

Forward-compatibility: the Header

To insure forward-compatibility, the only requirement is that **all the labels described in table 1 are present in the Header**. The order in which they appear is not fixed. Moreover, additional fields that are not described in table 1 are ignored. In this way, future extensions of the format that add new header fields will be compatible with parsers that are compliant with older versions.

TODO: Add table with the possible values of the field `bps`. TODO: Add explanation for the use of NSYS and NROWS

Why storing physical constants?

The physical constants present in the Header should match with the ones used internally by the BPS that produced the data. Knowing the values of the physical constants used helps avoiding conversion errors in post-processing.

Line 1 label	Description	Unit	Variable Type	Example
`version`	the adopted version of the format		string	1.0
`level`	the adopted standard level		string	L0
`bps`	the BPS used to generate the data		string	SEVN
`bpsVer`	the BPS version in use.		string	1.0.0
`contact`	contact person or email		string	pippo@pluto.com
`Z`	initial absolute metallicity		double	0.01
`NSYS`	number of systems in the table		int	1000
`NROWS`	number of rows in the table		int	25678
`Ggrav`	gravitational constant (cgs)	$cm^3\ g^{-1}\ s^{-2}$	double	6.67430e-8
`clight`	speed of light (cgs)	cm/s	double	2.99792458e10
`Msun`	mass of the Sun (cgs)	g	double	1.98847e33
`Rsun`	radius of the Sun (cgs)	cm	double	6.957e10
`Lsun`	luminosity of the Sun (cgs)	erg/s	double	3.828e33
`Zsun`	metallicity of the Sun		double	0.0134
`AU`	astronomical unit (cgs)	cm	double	1.495978707e13

Table 1: The fields required in the header. Note that different bps labels should indicate different codes or different code branches developed independently, otherwise the label version should mark different code releases. are there other fields that could be useful to have in the header? - RV

2.2 The Table of Values

From line 4 to line `NROWS+3`, the file contains a table of values corresponding, in the same order, to the columns listed in the line 3 of the header. Each row describes the state of a binary system at a given point in time, as a collection of values defining the physical status of the binary and its components. The states pertaining to the same systems must appear in contiguous rows and be ordered chronologically: if two consecutive lines pertain to the same system, the value of the `time` column must be strictly increasing with the line number. Table 2 describes the columns required by the format and their layered organization. The columns must be all present once the implemented level is established. The meaning of the levels is as following:

- Level 0 (L0) is referred to as Common Core Level, indicating the minimum set of physical variables describing a binary system physical status at fixed time (i.e. a binary state). All the BPS codes participating the collaboration share values of L0 so that their physical outcome can be safely compared once provided in the common format.

- Level 1 (L1) contains variables shared by the largest group of participating BPS codes. Not all codes are expected to provide L1 values so the comparison across codes is limited with respect to L0. L1 also serves to stimulate the development of missing features in BPS codes missing these qauntites.

- Level 2 (L2) groups physical values provided by some of the participating BPS without requirement of a common intersection. For specific applications or analysis they could be necessary.

Forward-compatibility: the Table of Values

To insure forward-compatibility, the only requirement is that **all the column described in table 2 are present in the Table of Values**. The labels are case sensitive and must be written exactly as in this document. The order in which they appear is not fixed. Moreover, additional columns that are not described in table 2 are ignored. Additional labels are thus not forbidden, while not accounted for by the standard. In this way, future extensions of the format that add new columns will be compatible with parsers that are compliant with older versions.

Undefined and missing values

Unless differently specified, when the value of a field is not defined, it is assigned the value `NaN`. When the value of a field is missing/unknown/not reported, the filed is left empty (i.e. two contiguous comma separators).
For instance, when the primary component becomes a black hole, its effective temperature is not defined anymore, and its field will be assigned the value `Teff1=NaN`. On the other hand, if the BPS code does not provide the effective temperature in the output, the component may or may not have an effective temperature, but we don't know its value. The field will then be left empty.

Label	Description	Unit	Variable Type
		L0	
ID	progressive system ID		long
UID	unique system ID		string
time	time	Myr	double
event	event type[a]		string
semiMajor	semi-major axis	$R_\odot$	double
eccentricity	orbital eccentricity		double
type1	type of object 1[b]		string
mass1	mass 1	$M_\odot$	double
radius1	radius 1	$R_\odot$	double
Teff1	effective temperature 1	K	double
massHecore1	He core mass 1	$M_\odot$	double
type2	type of object 2[b]		string
mass2	mass 2	$M_\odot$	double
radius2	radius 2	$R_\odot$	double
Teff2	effective temperature 2	K	double
massHeCore2	He core mass 2	$M_\odot$	double
		L1	
envBindEn	envelope binding energy	erg	double
massCOCore1	CO core mass 1	$M_\odot$	double
massCOCore2	CO core mass 2	$M_\odot$	double
radiusRL1	radius of the Roche Lobe 1	$R_\odot$	double
radiusRL2	radius of the Roche Lobe 1	$R_\odot$	double
period	orbital period	day	double
luminosity	bolometric luminosity	$L_\odot$	double
		L2	
RLOMass	mass accreted during RLO	$M_\odot$	double
jOrb	orbital angular momentum	$M_\odot\ {R_\odot}^2$/day	double
spin1	spin 1	$M_\odot\ {R_\odot}^2$/day	double
spin2	spin 2	$M_\odot\ {R_\odot}^2$/day	double
omega1	angular velocity 1	day^{-1}	double
omega2	angular velocity 2	day^{-1}	double
Hsup	H surface mass fraction		double
Hesup	He surface mass fraction		double
Csup	C surface mass fraction		double
Nsup	N surface mass fraction		double
Osup	O surface mass fraction		double

Table 2: The columns present in the table. [a]See section 3. [b]See section 4. some quantities need a better definition. Does massHeCore include also the mass of the CO? How is the boundary between core and envelope defined? Convection or chemical composition? Is the angular velocity measured at the surface? at the equator? Is the spin the absolute value or can it also be negative? - RV

ID and UID

There are two kinds of IDs. The *progressive system ID* (`ID`) is an integer number that is assigned progressively to the systems. The first system appearing in the file will have `ID` 1, the second will have `ID` 2 and so on until the last system with `ID` `NSYS`. The `ID` is an easy way to identify a system within the same file. For example, for a fast scanning of the systems, this value can be used to select ranges (e.g. 100-105) or to quickly find a specific system id to debug.
The *unique system ID* (`UID`) can instead be any string and can be used to identify systems across different files. The only requirement for the `UID` is that no two systems can have the same `UID` within the same file. It can be used to match the system described in the standard format with the analogous in the orginal file of the specific BPS output.

3 Component types

The component type is a numerical label that describes the evolutionary state of one of the two components of the binary system. The description can be generic (e.g. star) or detailed (e.g. asymptotic giant branch star), and this is achieved by placing the labels in a tree-like structure, where parents are more generic than children and leafs are the most specific labels. The numerical value of the label mimics the structure of the tree. The leftmost digit indicates the most generic branch, up to the rightmost digits that indicates the most specific leaf. For example, the numerical label for a first giant branch star is 123, where 1 in the first digits indicates that it is a star, 2 in the second digit means that it has a hydrogen envelope and 3 that it is currently burning hydrogen in a shell.

Future extensions to the format will preferably follow the same labelling pattern.

The following numerical labels are supported by this version of the common format. Any label that is not in the following list shall be considered equivalent to -2 (unknown/missing).

- 1 - Star: this category contains all objects that are undergoing some form of nuclear fusion in their interior and can be classified as stars
    - 11 - Protostar: an object of stellar mass that has not reached the zero age main sequence.

    - 12 - Star with a hydrogen envelope
        - 121 - Main sequence (or core hydrogen burning)
        - 122 - Hertzsprung gap
        - 123 - First giant branch (or shell hydrogen burning)
        - 124 - Core helium burning
        - 125 - Asymptotic giant branch (or shell helium burning)
            - 1251 - Early asymptotic giant branch
            - 1252 - Thermally pulsing asymptotic giant branch
    - 13 - Helium star: star that is completely depleted of a hydrogen envelope
        - 131 - Helium main sequence
        - 132 - Helium Hertzsprung gap
        - 133 - Helium first giant branch
    - 14 - Carbon star: star that is completely depleted of both hydrogen and helium envelope
    - 15 - Chemically homogeneous star: star that maintains a negligible internal chemical gradient due to efficient chemical mixing processes.
- 2 - White dwarf
    - 21 - Helium white dwarf
    - 22 - Carbon-Oxygen white dwarf
    - 23 - Oxygen-Neon white dwarf
- 3 - Neutron star
- 4 - Black hole
- 5 - Planet
- 6 - Brown dwarf
- 7 - Thorn-Zytkow Object: is a possible result of the merger of a star and a neutron star. A star with an accreting neutron star in the core.
- 9 - Other: anything that does not fit in any previous category.
- -1 - Massless remnant: this object doesn't exist anymore. e.g. it has exploded or has been disrupted or has merged with the companion. This label serves as placeholder.

- -2 - Unknown: the component type of the object is not specified. This information is missing.

4 Events

The event is a numerical label that describes what happened in the system that triggered the output of the current line (or equivalently the output of an interesting state). The numbering system is similar to the one used for the component type.

Hereafter, the character $*$ stands for 0, 1, 2 or 3 with the meaning

- 0 - not specified
- 1 - component 1
- 2 - component 2
- 3 - both components

The following numerical labels for events are supported by this version of the common format. Any label that is not in the following list shall be considered equivalent to -2 (unknown/missing).

- 1$*$ - Component $*$ changes type
- 2$*$ - Component $*$ goes supernova
  - 2$*$1 - runaway thermonuclear explosion in degenerate matter (Ia supernova, both single and double degenerate scenario)
  - 2$*$2 - Core collapse supernova
  - 2$*$3 - Electron capture supernova
  - 2$*$4 - Pair instability supernova
  - 2$*$5 - Pulsational pair instability supernova
  - 2$*$6 - Failed supernova (direct collapse into a black hole)
- 3$*$ - Component $*$ overflows its Roche lobe
- 4$*$ - Component $*$ goes back into its Roche lobe (end of stable mass transfer or end of a contact phase)
- 5$*$ - Component $*$ engulfs the companion, triggering a common envelope (53 is the double common envelope).

- 6 - the two components merge
- 7 - the system initiates a contact phase
- 8 - terminating condition reached, evolution is stopped. Only the last line of the evolution of a system can have an event with 8 as first digit.
    - 81 - max time reached
    - 82 - both components are compact remnants
    - 83 - the binary system is dissociated
    - 84 - only one object is left (e.g. due to a merger or because the companion as been disrupted
    - 85 - nothing left (both components are massless remnants)
    - 89 - other: a terminating condition different from any previous one
- 9 - other: any event that does not fit in any previous category.
- -1 - no notable events happened in this time step.
- -2 - unknown: the event is not specified. This information is missing.

5 Standard extension capabilities

The format is designed to be forward-compatible with the following classes of extensions:

- addition of fields in the Header
- addition of new columns in the Table of Values
- definition of new component types and event types