A plotting on primordial black holes observational constraints

Jesus P. D. F.[1], Sobrinho J. L. G.

June 2025

Grupo de Astronomia da Universidade da Madeira
Faculdade de Ciências Exatas e da Engenharia
Universidade da Madeira, Campus da Penteada, 9020-105 Funchal
astro@uma.pt
https://astro.web.uma.pt/Grupo/index.htm

Resumo

Primordial Black Holes (PBHs) might have formed at the initial stages of the Universe with masses spanning several orders of magnitude. Although to the present day we are not aware of any detection of PBHs there are a set of observational constraints limiting the possible number density allowed for them at different mass ranges. In this work we present a code (one version written in Wolfram Mathematica and another one in Python) which aims to implement known observational constraints on the number density of PBHs. The code is written in a simple and flexible way, allowing the introduction of new constraints as well as the update/refinement of existing ones.

Keywords: black hole physics - stars: black holes - cosmology: cosmic background radiation, dark matter, early Universe, observations, primordial nucleosynthesis - gamma-rays: diffuse background

1 Introduction

Black holes (BHs) are objects predicted by the laws of physics. In terms of mass, a BH can be classified as supermassive ($\gtrsim 10^6 M_\odot$), intermediate ($\sim 10^3 - 10^5 M_\odot$), stellar ($\sim 1 - 100 M_\odot$) or substellar ($< 1 M_\odot$). It is now well established that supermassive BHs reside in the centres of galaxies (e.g. Kormendy & Ho, 2013; van den Bosch & de Zeeuw, 2010; Nóbrega & Sobrinho, 2020) including our Galaxy with a $\approx 4.297 \times 10^6 M_\odot$ BH at its centre (Gravity Collaboration et al., 2023). A stellar mass BH is the likely remnant of a star with an initial mass greater than $\sim 40 - 60 M_\odot$. With $\approx 33 M_\odot$ Gaia BH3 serves as an example of a stellar mass BH candidate which was recently discovered (Gaia Collaboration et al., 2024). It has

[1] Jesus P. D. F. was supported by the **INGRESSA** 2025 programme, promoted by the **Secretaria Regional da Educação**, through the **Direção Regional de Juventude** (DRJ) - Região Autónoma da Madeira.

not yet been possible to demonstrate beyond reasonable doubt the existence of a single intermediate mass BH. Nevertheless, a few candidates have been identified so far (e.g. Sobrinho & Augusto, 2024).

We are aware of some processes that could lead to BH formation in the present day Universe. This includes BHs with masses from $\sim 3M_\odot$ up to $10^{12}M_\odot$ (e.g. Chisholm, Dodelson, & Kolb, 2003; Volonteri, 2010; Burrows, Wang, & Vartanyan, 2024). Although BHs with less than $3M_\odot$ could not have been formed through stellar evolution (e.g. Chisholm, Dodelson, & Kolb, 2003) they might have formed in the Early Universe due, for example, to the gravitational collapse of density fluctuations (e.g. Hawking, 1971; Carr, 1975; Novikov et al., 1979; Polarski & Dolgov, 2001). Such BHs are refered to as Primordial Black Holes (PBHs). Depending on the formation epoch and/or on the formation mechanism these PBHs could have masses ranging from the Planck Mass ($\sim 10^{-5}$ g) up to $\sim 10^{12}M_\odot$ (Carr et al., 2021b). It is not ruled out the possibility that at least some of the currently observed BHs or BH candidates consist in PBHs. For example, some of the binary BHs associated with gravitational wave events could be of primordial origin (e.g. Freitas & Sobrinho, 2021).

PBHs have been a source of intense study for more than 50 years, despite the fact that there is still no evidence for them. PBHs have been invoked to explain numerous cosmological features such as extragalactic and Galactic γ-ray backgrounds, the presence of antimatter in cosmic rays, the mechanism of short-period gamma-ray bursts, lensing effects, the origin of massive compact halo objects, the orgin of supermassive black holes in galactic nuclei or gravitational wave events. Generally speaking, there are other possible explanations for these features, so there is still no definitive evidence for PBHs. However, studying each of these (and other) effects allows one to place observational constraints on the number density of PBHs of a given mass (e.g. Carr et al., 2021a).

The aim of this report is to present a code written in Wolfram Mathematica as well as in Python that implements, in a very flexible way, some of the main observational constraints on PBHs. The report is organized as follows: after reviewing, in Section 2, some main functions that will be implemented in the code, in Section 3, we list the observational constraints on PBHs. In Section 4 we discuss some aspects concerning the implementation code. Finally, in Section 5 we present some conclusions.

2 The main functions

The majority of the PBHs formed at a particular epoch might have masses within the order of the horizon mass, M_H, at that epoch (e.g. Sobrinho et al., 2016):

$$M_H(t) \sim 10^{15} \left(\frac{t}{10^{-23}\ \mathrm{s}} \right)\ \mathrm{g} \tag{1}$$

We can also think about the age of the Universe t_H for which the horizon mass assumed some value M. From equation (1) we get:

$$t_H(t) \sim 10^{-23} \left(\frac{M}{10^{15}\ \mathrm{g}}\right)\ \mathrm{s} \tag{2}$$

For example, a PBH formed when the Universe was 10^{-27} s old would have a mass of $\approx 10^{11}$ g (comparable to the mass of a small asteroid) and a PBH formed when the Universe was 10^{-5} s old would have $\approx 0.5 M_\odot$.

As a PBH radiates due to the Hawking process, it loses mass or, in other words, it evaporates. The amount of time required for a PBH to change its mass from an initial value M_i to a final value M_f (with $M_f < M_i$) through the evaporation mechanism is given by (e.g. Sobrinho & Augusto, 2014):

$$t_{evap} \approx \frac{M_i^3 - M_f^3}{1.6 \times 10^{17} f_\chi(M)} \tag{3}$$

where

$$\begin{aligned} f_\chi(M) \approx 1.569 + 0.569 \Bigg[& \exp\left(\frac{-M}{4.53 \times 10^{14}}\right) + 6\exp\left(\frac{-M}{1.60 \times 10^{14}}\right) + \\ & + 3\exp\left(\frac{-M}{9.60 \times 10^{13}}\right) + 3\exp\left(\frac{-M}{2.56 \times 10^{13}}\right) + \\ & + \exp\left(\frac{-M}{2.68 \times 10^{13}}\right) + 3\exp\left(\frac{-M}{9.07 \times 10^{12}}\right) + \\ & + 3\exp\left(\frac{-M}{0.48 \times 10^{12}}\right)\Bigg] + 0.963\left[\exp\left(\frac{-M}{1.10 \times 10^{14}}\right)\right] \end{aligned} \tag{4}$$

is a non-dimensional function of the PBH mass M accounting for the contributions of the different species of particles being emitted by the PBH (Gibilisco, 1997).

The critical mass for which the lifetime of an evaporating PBH equals the present day age of the Universe is (Carr et al., 2021a):

$$M_* \approx 5.1 \times 10^{14}\ \mathrm{g} \tag{5}$$

This value corresponds to the horizon mass when the age of the Universe was (cf. equation 2):

$$t_H \approx 5.0 \times 10^{-24}\ \mathrm{s} \tag{6}$$

The probability that a fluctuation crossing the horizon at some instant t_k has of collapsing and forming a PBH can be written as (e.g. Sobrinho et al., 2016):

$$\beta(t_k) = \frac{1}{\sqrt{2\pi}\sigma(t_k)} \int_{\delta_c}^{\infty} \exp\left(-\frac{\delta^2}{2\sigma^2(tk)}\right) d\delta \tag{7}$$

where $\sigma^2(tk)$ is the mass variance at horizon crossing and δ_c is the threshold for PBH formation (see Sobrinho et al., 2016, and references therein for more details).

In the case of perturbations with δ only slightly larger than the critical value, δ_c, the PBH masses rather obey the scaling law (Niemeyer & Jedamzik, 1999):

$$M_{PBH} \propto M_H \left(\delta - \delta_c\right)^{\Gamma} \tag{8}$$

where $\Gamma \approx 0.36$ in the case of a radiation-dominated Universe. Assuming that for the majority of the PBHs formed at a given instant t we have $M_{PBH}(t) \approx M_H(t)$ we can relate directly the mass of a particular PBH with the instant of formation using equations (1) and (2). The fraction of the Universe going into PBHs (cf. equation 7) of a given mass M can be also written in the form (Carr et al., 2021a):

$$\beta(M) \approx 7.06 \times 10^{-18} \gamma^{-1/2} \left(\frac{h}{0.67}\right)^2 \left(\frac{g_{*i}}{106.75}\right)^{1/4} \Omega_{PBH} \left(\frac{M}{M_\odot}\right)^{1/2} \tag{9}$$

where γ is a numerical factor somewhat below unity which depends on the details of the gravitational collapse, h is the Hubble parameter, g_{*i} represents the effective number of degrees of freedom at the epoch of PBH formation, and Ω_{PBH} is the present day density parameter of the PBHs. It may be useful to define a new parameter (Carr et al., 2021a):

$$\beta'(M) = \gamma^{1/2} \left(\frac{h}{0.67}\right)^{-2} \left(\frac{g_{*i}}{106.75}\right)^{-1/4} \beta(M) \tag{10}$$

which is valid as long as the Universe does not deviate from the standard radiation-dominated behaviour.

The fraction $f(M)$ of the halo in PBHs can be expressed as (Carr et al., 2021a):

$$f(M) = 3.81 \times 10^8 \beta'(M) \left(\frac{M}{M_\odot}\right)^{-1/2} \tag{11}$$

assuming that the PBHs have a monochromatic mass function (i.e. they span a mass range of $\Delta M_{PBH} \sim M_{PBH}$) and cluster in galactic halos in the same way as other forms of CDM do (unless they are too large).

PBHs with initial masses smaller than M_* have evaporated completely and therefore cannot contribute to the DM halo. In this mass range the function $\beta(M)$ (or $\beta'(M)$) is more meaningfull than the $f(M)$ function. Nevertheless, one can still mathematically relate $f(M)$ to $\beta'(M)$ (and, consequently to $\beta(M)$) using equation (11) (see Carr et al., 2021a).

The fitting functions giving the effective number of degrees of freedom for the energy density $g_{*\rho}$ and the effective number of degrees of freedom for the entropy density g_{*s} in the high energy regime (120 MeV $\leq T \leq 10^{16}$ GeV) can be written as (Saikawa & Shirai, 2018):

$$g_{*\rho}(T) \approx \frac{\displaystyle\sum_{i=0}^{11} a_i t^i}{\displaystyle\sum_{i=0}^{11} b_i t^i}, \tag{12}$$

$$\frac{g_{*\rho}(T)}{g_{*s}(T)} \approx 1 + \frac{\displaystyle\sum_{i=0}^{11} c_i t^i}{\displaystyle\sum_{i=0}^{11} d_i t^i}, \tag{13}$$

where $t = \ln(T[\mathrm{GeV}])$ and the coefficients a_i, b_i, c_i and d_i are the ones shown in Table 1. For the low energy regime ($T < 120$ MeV), we have the fitting functions (Saikawa & Shirai, 2018):

$$\begin{aligned} g_{*\rho}(T) \approx{}& 2.030 + 1.353\ S_{fit}^{\frac{4}{3}}\left(\frac{m_e}{T}\right) + 3.495\ f_p\left(\frac{m_e}{T}\right) + 3.446\ f_p\left(\frac{m_\mu}{T}\right) \\ &+ 1.05\ b_p\left(\frac{m_{\pi^0}}{T}\right) + 2.08\ b_p\left(\frac{m_{\pi^\pm}}{T}\right) + 4.165\ b_p\left(\frac{m_1}{T}\right) + \\ &+ 30.55\ b_p\left(\frac{m_2}{T}\right) + 89.4\ b_p\left(\frac{m_3}{T}\right) + 8209\ b_p\left(\frac{m_4}{T}\right) \end{aligned} \tag{14}$$

$$\begin{aligned} g_{*s}(T) \approx{}& 2.008 + 1.923\ S_{\text{fit}}\left(\frac{m_e}{T}\right) + 3.442\ f_s\left(\frac{m_e}{T}\right) + 3.468\ f_s\left(\frac{m_\mu}{T}\right) + \\ &+ 1.034\ b_s\left(\frac{m_{\pi^0}}{T}\right) + 2.068\ b_s\left(\frac{m_{\pi^\pm}}{T}\right) + 4.16\ b_s\left(\frac{m_1}{T}\right) + \\ &+ 30.55\ b_s\left(\frac{m_2}{T}\right) + 90\ b_s\left(\frac{m_3}{T}\right) + 6209\ b_s\left(\frac{m_4}{T}\right) \end{aligned} \tag{15}$$

where the mass values are $m_e = 5.11 \times 10^{-6}$ GeV, $m_\mu = 0.1056$ GeV, $m_{\pi^0} = 0.135$ GeV, $m_{\pi^\pm} = 0.140$ GeV, $m_1 = 0.5$ GeV, $m_2 = 0.77$ GeV, $m_3 = 1.2$ GeV and $m_4 = 2$ GeV.

The functions f_p, b_p, f_s, b_s and S_{fit} used in the right-hand sides of equations (14) and (15) are given by the expressions:

$$f_p(x) = e^{-1.04855x}\left(1 + 1.03757x + 0.508630x^2 + 0.0893988x^3\right) \tag{16}$$

Tabela 1: Coefficients for the fitting functions for $g_{*\rho}$ (equation 12) and g_{*s} (equation 13) when 120 MeV $\leq T \leq 10^{16}$ GeV (Saikawa & Shirai, 2018).

i	a_i	b_i	c_i	d_i
0	1	1.43382E-02	1	7.07388E+01
1	1.11724E+00	1.37559E-02	6.07869E-01	9.18011E+01
2	3.12672E-01	2.92108E-03	-1.54485E-01	3.31892E+01
3	-4.68049E-02	-5.38533E-04	-2.24034E-01	-1.39779E+00
4	-2.65004E-02	-1.62496E-04	-2.82147E-02	-1.52558E+00
5	-1.19760E-03	-2.87906E-05	2.90620E-02	-1.97857E-02
6	1.82812E-04	-3.84278E-06	6.86778E-03	-1.60146E-01
7	1.36436E-04	2.78776E-06	-1.00005E-03	8.22615E-05
8	8.55051E-05	7.40342E-07	-1.69104E-04	2.02651E-02
9	1.22840E-05	1.17210E-07	1.06301E-05	-1.82134E-05
10	3.82259E-07	3.72499E-09	1.69528E-06	7.83943E-05
11	-6.87035E-09	-6.74107E-11	-9.33311E-08	7.13518E-05

$$b_p(x) = e^{-1.03149x}\left(1 + 1.03317x + 0.398264x^2 + 0.0648056x^3\right) \tag{17}$$

$$f_s(x) = e^{-1.04190x}\left(1 + 1.03400x + 0.456426x^2 + 0.0595248x^3\right) \tag{18}$$

$$b_s(x) = e^{-1.03365x}\left(1 + 1.03397x + 0.342548x^2 + 0.0506182x^3\right) \tag{19}$$

$$S_{\text{fit}}(x) = 1 + \frac{7}{4}e^{-1.0419x}\left(1 + 1.034x + 0.456426x^2 + 0.0595249x^3\right) \tag{20}$$

3 List of observational constraints

In order to establish a comprehensive list on PBH observational constraints we considered the review from Carr et al. (2021a) which provides a complete update on the subject. In total we have considered 55 observational constraints (see Table 2), all of them retrieved from Carr et al. (2021a) with the exception of the constraint related to the CMB dipole limit which was retrieved from Carr et al. (2021b).

It is important to note that these constraints have, as a general rule, different degrees of uncertainty associated with them. It may be relevant to consider whether a given constraint should be taken into account or not (see Carr et al., 2021a, for a discussion on this topic).

In order to accommodate this possibility we defined for each constraint in our code a boolean type variable which states if that particular constraint is enabled (1) or disabled (0). For simplicity, the names of these variables match the acronyms of the constraints (see Table 2). The default values (DV) assigned in each case are in accordance with Carr et al. (2021a).

Some of the constraints are given in the form of piecewise functions. For these cases we consider each of the branches as a specific constraint on its own. For example, from the Millilensing of Compact Radio Sources we have the constraint (Carr et al., 2021a):

$$f(M) < \begin{cases} \left(\dfrac{M}{2\times 10^4 M_\odot}\right)^{-2} & , \quad M < 10^5 M_\odot \\ 0.06 & , \quad 10^5 M_\odot < M < 10^8 M_\odot \\ \left(\dfrac{M}{4\times 10^8 M_\odot}\right)^{2} & , \quad M > 10^8 M_\odot \end{cases} \tag{21}$$

We separate this constraint into three, namely, CRSMLS1, CRSMLS2 and CRSMLS3 (cf. Table 2).

Tabela 2: Observational constraints on PBHs. For each constraint we establish an acronym which is also the name of the boolean type variable that specifies if the constraint is enabled or disabled when executing the code. The column labeled DV indicates if the corresponding constraint is enabled by default or not in agreement with the discussion presented in Carr et al. (2021a).

Acronym	**Constraint**	**DV**
BBNDH	Big Bang Nucleosynthesis D/H	1
BBNY	Big Bang Nucleosynthesis Y	1
GE	Generation of entropy	0
CMBSD	CMB spectral distortions	1
CMBA	CMB small scale anisotropies	1
EGB1	Extragalactic gamma-ray background 1 (evaporated)	1
EGB2	Extragalactic gamma-ray background 2 (evaporated)	1
EGB3	Extragalactic gamma-ray background 3 (exploding)	1
EGB4	Extragalactic gamma-ray background 4 (evaporating)	1
EGAP	Extragalactic antiprotons	1
LSP	Emission of the Lightest Supersymmetric Particle LSP	1
DUP	Decay of unstable particles: e.g. gravitino, neutralino	0
	continues on next page	

Tabela 2 – continuation

Acronym	Constraint	DV
GRBFL	Femtolensing of GRBs	0
EROSMLS1	EROS microlensing of stars 1	1
EROSMLS2	EROS microlensing of stars 2	1
EROSMLS3	EROS microlensing of stars 3	1
OGLEMLS1	OGLE microlensing of stars 1	1
OGLEMLS2	OGLE microlensing of stars 2	1
OGLEMLS3	OGLE microlensing of stars 3	1
KeplerMLS	Kepler microlensing of stars	1
QMLS	Quasar microlensing	1
CRSMLS1	Millilensing of compact radio sources 1	1
CRSMLS2	Millilensing of compact radio sources 2	1
CRSMLS3	Millilensing of compact radio sources 3	1
WDNS	Capture of PBHs by white dwarfs and neutron stars	1
WBS1	Wide-binary stability 1	1
WBS2	Wide-binary stability 2	1
DGC1	Disruption of globular clusters 1	1
DGC2	Disruption of globular clusters 2	1
ERI1	Eridanus II - survival of star clusters 1	1
ERI2	Eridanus II - survival of star clusters 2	1
DHTS1	Disk heating and tidal streams 1	1
DHTS2	Disk heating and tidal streams 2	1
DFHO1	Dynamical friction on halo objects 1	1
DFHO2	Dynamical friction on halo objects 2	1
DFHO3	Dynamical friction on halo objects 3	1
DFHO4	Dynamical friction on halo objects 4	1
DDGC1	Disruption/distortion of galaxies in clusters 1	1
DDGC2	Disruption/distortion of galaxies in clusters 2	1
DDGC3	Disruption/distortion of galaxies in clusters 3	1
MWG1	MW type galaxies 1	1
MWG2	MW type galaxies 2	1
DG1	Dwarf type galaxies 1	1
DG2	Dwarf type galaxies 2	1
CG1	Clusters of galaxies 1	1
CG2	Clusters of galaxies 2	1
FBC1	First baryonic clouds 1	0
FBC2	First baryonic clouds 2	0
WMAPCMB1	WMAP-CMB anisotropies/distortions 1	0
WMAPCMB2	WMAP-CMB anisotropies/distortions 2	0
WMAPCMB3	WMAP-CMB anisotropies/distortions 3	0
FIRASCMB1	FIRAS-CMB anisotropies/distortions 1	0

continues on next page

Tabela 2 – continuation

Acronym	Constraint	DV
FIRASCMB2	FIRAS-CMB anisotropies/distortions 2	0
FIRASCMB3	FIRAS-CMB anisotropies/distortions 3	0
CMBDL	CMB dipole limit	1

3.1 Constraints on evaporated/evaporating PBHs

In Table 3 we show a list of 11 constraints on PBHs that have already evaporated completely or are in the final stage of their evaporation. In the following text and also in the caption of Table 3 we present a few remarks about these constraints (see Carr et al., 2021a, for further details).

High energy particles emitted by PBHs modify the standard Big Bang Nucleosynthesis (BBN) scenario in different ways. The corresponding PBH constraints depend on three parameters: the initial baryon-to-photon ratio, the PBH initial mass M or, equivalently, its lifetime τ, and the initial PBH number density normalized to the entropy density ($Y_{PBH} = n_{PBH}/s$) giving (Carr et al., 2021a):

$$\beta'(\tau) = 5.4 \times 10^{21} \left(\frac{\tau}{1 \text{ s}}\right)^{1/2} \tag{22}$$

The computational implementation of this function offers some chalanges. We decided to consider instead an approach using two line segments:

$$\beta'(M) = \begin{cases} 2 \times 10^{-19} & , \quad 10^{9} \text{ g} < M < 10^{10} \text{ g} \\ \\ 10^{-24} & , \quad 10^{10} \text{ g} < M < 2 \times 10^{13} \text{ g} \end{cases} \tag{23}$$

as shown in Figure 1. In terms of constraints these two segments are identified as Big Bang Nucleosynthesis Y (BBNY) and Big Bang Nucleosynthesis D/H (BBNDH) - cf. Table 2.

PBHs with an initial mass equal to M_* (cf. equation 5) are supposed to be at the final stages of their evaporation process, i.e., they should be exploding by now in the form of a γ-ray Burst (GRB). As a matter of fact the measured γ-ray background implies an upper limit of $\beta'(M) \sim 10^{-29}$ which is the strongest constraint on PBHs over all mass ranges known to date (Carr et al., 2021a). We represent this situation with the constraint EGB3 (cf. Tables 2 and 3). In implementing this constraint we consider the mass range $[M_*(1-\varepsilon), M_*(1+\varepsilon)]$ with $\varepsilon = 1.75\%$ (cf. Appendices A and B). Note that if necessary the value of ε can be easily changed in the code.

Tabela 3: Constraints on evaporated/evaporating PBHs. For each constraint it is indicated the respective acronym (see Table 2) and the expression giving the upper limit allowed for $\beta'(M)$, the mass range in which the constraint is valid and, finally, the corresponding time range. Numerical parameters apearing in some of the constraints: CMBA: $f_H \approx 0.1$; EGB: $\epsilon \approx 0.4$; EGAP: $r \approx 9 \times 10^{-4}$ and $f_* \approx 1.9$; LSP: $M_{LSP} = 32.3$ GeV (cf. Abreu et al. (2000)); DUP: $Y_{PBH} \approx 10^{-14}$ and $x_\Phi \approx 0.006$ (see text and Carr et al. (2021a) for more details).

Constraint	$\beta'(M)$ upper limit	Mass range	Time range
BBNY	2×10^{-19}	10^9 g - 10^{10} g	10^{-29} s - 10^{-28} s
BBNDH	10^{-24}	10^{10} g - 2.0×10^{13} g	10^{-28} s - $10^{-24.7}$ s
GE	$10^{-5}(\frac{M}{10^9\ \mathrm{g}})^{-1}$	$< 10^9$ g	$< 10^{-29}$ s
CMBSD	$10^{-16}(\frac{M}{10^{11}\ \mathrm{g}})^{-1}$	10^{11} g - 10^{13} g	10^{-27} s - 10^{-25} s
CMBA	$3 \times 10^{-30}(\frac{f_H}{0.0\ \mathrm{g}})^{-1}(\frac{M}{10^{13}})^{3.1}$	2.5×10^{13} g - 2.4×10^{14} g	$10^{-25.6}$ s - $10^{-24.6}$ s
EGB1	$5 \times 10^{-28} \left(\frac{M}{M_*}\right)^{-\frac{5}{2}-2\epsilon}$	$< M_*$	$< 5.0 \times 10^{-24}$ s
EGB2	$5 \times 10^{-26} \left(\frac{M}{M_*}\right)^{\frac{7}{2}+\epsilon}$	$> M_*$	$> 5.0 \times 10^{-24}$ s
EGB3	$\approx 10^{-29}$	$\approx M_*$	$\approx 5.0 \times 10^{-24}$ s
EGAP	$3 \times 10^{24} \left(\frac{r}{10^{-5}}\right) \left(\frac{f_\chi(M)}{f_*}\right) \left(\frac{M}{M_*}\right)^{-1/2}$	2×10^4 g - 4×10^{13} g	2×10^{-34} s - 4×10^{-25} s
LSP	$10^{-18} \left(\frac{M}{10^{11}\,\mathrm{g}}\right)^{-1/2} \left(\frac{M_{\mathrm{LSP}}}{100\,\mathrm{GeV}}\right)^{-1}$	$< 10^{11} \left(\frac{M_{\mathrm{LSP}}}{10\,\mathrm{GeV}}\right)^{-1}$ g	$< 10^{-27} \left(\frac{\mathrm{M_{LSP}}}{10GeV}\right)^{-1}$ s
DUP	$5 \times 10^{-19} \left(\frac{M}{10^9\ \mathrm{g}}\right)^{-1/2} \left(\frac{Y_{PHB}}{10^{-14}}\right) \left(\frac{x_\Phi}{0.006}\right)^{-1}$	$< 10^9$ g	$< 10^{-29}$ s

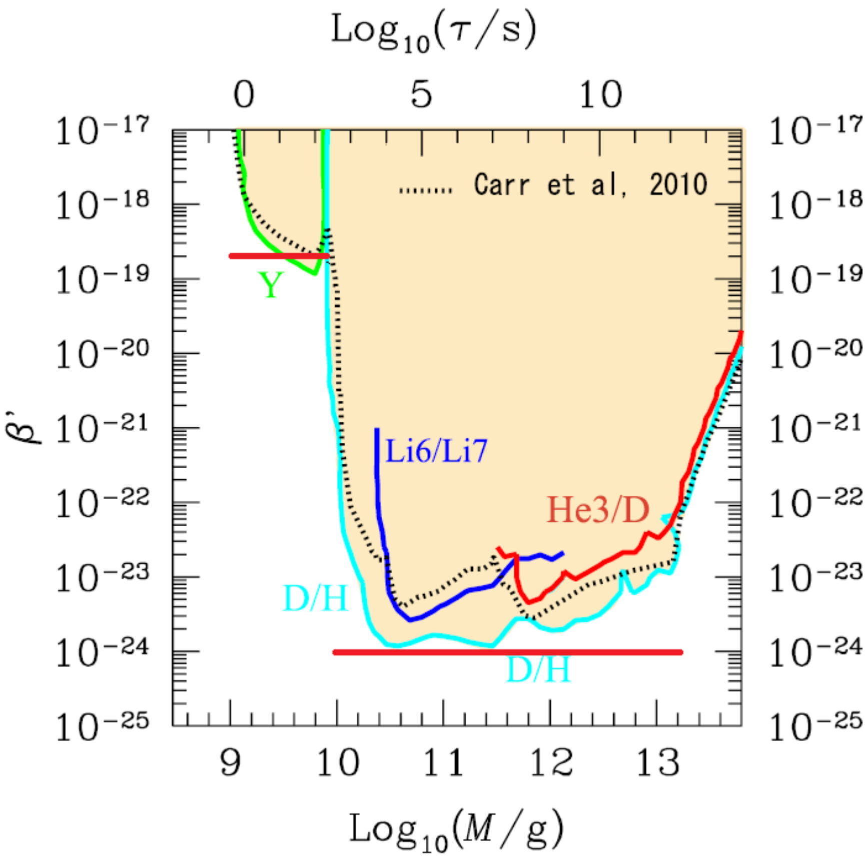

Figura 1: Upper bounds on $\beta'(M)$ required for the success of the Big Bang Nucleosynthesis (BBN) model. The two horizontal solid lines at $\beta'(M) \approx 2 \times 10^{-19}$ and $\beta'(M) \approx 2 \times 10^{-24}$ represent our approach for the Big Bang Nucleosynthesis Y (BBNY) and Big Bang Nucleosynthesis D/H (BBNDH) constraints (adapted from Carr et al., 2021a).

3.2 Constraints on non-evaporated PBHs

PBHs with initial masses $> M_*$ are too large to have evaporated completely by now. In this case the constraints on PBHs are usually expressed in terms of the function $f(M)$ which gives the fraction of the halo in PBHs, assuming that PBHs cluster in galactic halos in the same way as other forms of CDM do (unless they are too large) and that PBHs have a monochromatic mass function ($\Delta M_{PBH} \sim M_{PBH}$). The observational constraints on $f(M)$ derive mainly from:

- PBH evaporations
- gravitational lensing

- dynamical effects
- PBH accretion
- gravitational wave observations

Associated with these constraints we have different degrees of uncertainties which could be related with the observations and/or with the physics envolved (for a detailed discussion see Carr et al., 2021a). Many of the constraints depend on physical parameters and others are related to different cosmological and astrophysical assumptions. Some of the constraints are extended into the unphysical $f(M) > 1$ domain (see Carr et al., 2021a). This is because it is expected for some of these limits to eventually shift downward as the observational data improves but keeping the same mathematical form.

In Table 4 we show a list of 44 constraints on non-evaporated PBHs covering the mass range $M_* - \sim 10^{21} M_\odot$.

4 The code

4.1 The time/mass axis

We considered the observational constraints as a function of time, i.e., as a function of the age of the Universe, and also as a function of the horizon mass (see equation 1). In order to perform the numerical computation we assume a discrete logarithmic time scale, with the time interval between two consecutive instants given by:

$$\log_{10}\left(\frac{t_{i+1}}{1\ \mathrm{s}}\right) - \log_{10}\left(\frac{t_i}{1\ \mathrm{s}}\right) = \mathrm{stept} \Leftrightarrow t_{i+1} = t_i 10^{\ \mathrm{stept}} \tag{24}$$

The default value assumed for the parameter stept is 0.005. In that case we get:

$$t_{i+1} \approx 1.01158 t_i \tag{25}$$

The time axis extends from min1 into max1 with:

$$\mathrm{min1} = \log_{10}\left(\frac{t_{min}}{1\ \mathrm{s}}\right) \tag{26}$$

and

$$\mathrm{max1} = \log_{10}\left(\frac{t_{max}}{1\ \mathrm{s}}\right) \tag{27}$$

with t_{min} and t_{max} representing the two extremes of the time interval that we are interested in. The default values of min1 and max1 are, respectively, -30 and 17.

Tabela 4: Constraints on non-evaporated PBHs. For each constraint it is indicated the respective acronym (see Table 2) and the expression giving the upper limit allowed for $f(M)$, the mass range in which the constraint is valid and, finally, the corresponding time range. Numerical parameters appearing in some of the constraints: EGB4: $0.1 < \epsilon < 0.4$ (Carr et al., 2021a); DHTS2 and DHFO4: $M_{halo} \approx 10^{12} M_\odot$ (cf. Gardner, McDermott, & Yanny, 2021); DFHO1, DFHO2 and DFFHO3: r_c represents the Milky Way halo core radius normalized to 2 kpc (Carr et al., 2021a).

Constraint	$f(M)$ upper limit	Mass range	Time range
GRBFL	0.1	5×10^{16} g - 10^{19} g	5×10^{-22} s - 10^{-19} s
EGB4	$2 \times 10^{-8} \left(\frac{M}{M_*}\right)^{3+\epsilon}$	$> M_*$	$> 5 \times 10^{-24}$ s
EROSMLS1	1	$6 \times 10^{-8} M_\odot$ - $15 M_\odot$	1.2×10^{-12} s - 3.0×10^{-4} s
EROSMLS2	0.1	$10^{-6} M_\odot$ - $1 M_\odot$	2.0×10^{-11} s - 2.0×10^{-5} s
EROSMLS3	0.04	$10^{-3} M_\odot$ - $0.1 M_\odot$	2.0×10^{-8} s - 2.0×10^{-6} s
OGLEMLS1	1	$10^{-7} M_\odot$ - $1 M_\odot$	2.0×10^{-12} s - 2.0×10^{-5} s
OGLEMLS2	0.1	$10^{-6} M_\odot$ - $10^{-2} M_\odot$	2.0×10^{-11} s - 2.0×10^{-7} s
OGLEMLS3	0.01	$\approx 10^{-4} M_\odot$	$\approx 2.0 \times 10^{-9}$ s
KeplerMLS	0.3	$2 \times 10^{-9} M_\odot$ - $10^{-7} M_\odot$	4.0×10^{-14} s - 2.0×10^{-12} s
QMLS	1	$10^{-3} M_\odot$ - $60 M_\odot$	2.0×10^{-8} s - 1.2×10^{-3} s
CRSMLS1	$\left(\frac{M}{2\times 10^4 M_\odot}\right)^{-2}$	$< 10^5 M_\odot$	< 2.0 s
CRSMLS2	0.06	$10^5 M_\odot$ - $10^8 M_\odot$	2.0 s - 1989.0 s
CRSMLS3	$\left(\frac{M}{4\times 10^8 M_\odot}\right)^{2}$	$> 10^8 M_\odot$	> 1989.0 s
WDNS	$\left(\frac{M}{4.7\times 10^{24}\,\mathrm{g}}\right)\left(1 - \exp\left[-\left(\frac{M}{2.9\times 10^{23}\,\mathrm{g}}\right)\right]\right)^{-1}$	2.5×10^{18} g - 10^{25} g	2.5×10^{-20} s - 1.0×10^{-13} s

(continues on next page)

Table 4 (continued).

Constraint	$f(M)$ upper limit	Mass range	Time range
WBS1	$\left(\frac{M}{500\,M_\odot}\right)^{-1}$	$500\,M_\odot$ - $10^3\,M_\odot$	$9.9\times10^{-3}\,$s - $2.0\times10^{-2}\,$s
WBS2	0.5	$10^3\,M_\odot$ - $10^8\,M_\odot$	$2.0\times10^{-2}\,$s - $2.0\times10^{3}\,$s
DGC1	$\left(\frac{M}{3\times10^4 M_\odot}\right)^{-1}$	$3\times10^4 M_\odot$ - $10^6 M_\odot$	$6\times10^{-1}\,$s - $19.89\,$s
DGC2	0.03	$10^6 M_\odot$ - $10^{11} M_\odot$	$19.89\,$s - $2\times10^{6}\,$s
ERI1	$\left(\frac{M}{3.7\,M_\odot}\right)^{-1} / \left[1.1 - 0.1\ln\left(\frac{M}{M_\odot}\right)\right]$	$< 10^4\,M_\odot$	$< 2\times10^{-1}\,$s
ERI12	$\frac{M}{10^6\,M_\odot}$	$> 10^4\,M_\odot$	$> 2\times10^{-1}\,$s
DHTS1	$\left(\frac{M}{3\times10^6\,M_\odot}\right)^{-1}$	$< 3\times10^9\,M_\odot$	$< 6\times10^4\,$s
DHTS2	$\frac{M}{M_{\rm halo}}$	$> 3\times10^9\,M_\odot$	$> 6\times10^4\,$s
DFHO1	$\left(\frac{M}{2\times10^4\,M_\odot}\right)^{-10/7}\left(\frac{r_c}{2\,{\rm kpc}}\right)^2$	$< 5\times10^5\,M_\odot$	$< 9.9\,$s
DFHO2	$\left(\frac{M}{4\times10^4\,M_\odot}\right)^{-2}\left(\frac{r_c}{2\,{\rm kpc}}\right)^2$	$5\times10^5\,M_\odot$ - $2\times10^6\left(\frac{r_c}{2\,{\rm kpc}}\right)M_\odot$	$9.9\,$s - $4\times10^1\left(\frac{r_c}{2\,{\rm kpc}}\right)\,$s
DFHO3	$\left(\frac{M}{0.1\,M_\odot}\right)^{-1/2}$	$2\times10^6\left(\frac{r_c}{2\,{\rm kpc}}\right)M_\odot$ - $10^8 M_\odot$	$3.98\times10^1\left(\frac{r_c}{2\,{\rm kpc}}\right)\,$s - $1.99\times10^3\,$s
DFHO4	$\frac{M}{M_{\rm halo}}$	$> 10^8 M_\odot$	$> 2\times10^3\,$s
DDGC1	$\left(\frac{M}{10^{10}\,M_\odot}\right)^{-1}$	$10^{10}\,M_\odot$ - $10^{11}\,M_\odot$	$2.0\times10^5\,$s - $2.0\times10^6\,$s
DDGC2	0.1	$10^{11}\,M_\odot$ - $10^{13}\,M_\odot$	$2.0\times10^6\,$s - $2.0\times10^8\,$s
DDGC3	$\frac{M}{2\times10^{14}\,M_\odot}$	$> 10^{13}\,M_\odot$	$> 2.0\times10^8\,$s
MWG1	$\left(\frac{M}{10^6\,M_\odot}\right)^{-1}$	$10^6\,M_\odot$ - $10^9\,M_\odot$	$2.0\times10^1\,$s - $2.0\times10^4\,$s
MWG2	$\frac{M}{10^{12}\,M_\odot}$	$10^9\,M_\odot$ - $10^{12}\,M_\odot$	$2.0\times10^4\,$s - $2.0\times10^7\,$s

(continues on next page)

Table 4 (continued).

Constraint	$f(M)$ upper limit	Mass range	Time range
DG1	$\left(\frac{M}{10^5\,M_\odot}\right)^{-1}$	$10^5\,M_\odot$ - $3\times10^7\,M_\odot$	$2.0\,\mathrm{s}$ - $6.0\times10^2\,\mathrm{s}$
DG2	$\frac{M}{10^{10}\,M_\odot}$	$3\times10^7\,M_\odot$ - $10^{10}\,M_\odot$	$6.0\times10^2\,\mathrm{s}$ - $2.0\times10^5\,\mathrm{s}$
CG1	$\left(\frac{M}{10^7\,M_\odot}\right)^{-1}$	$10^7\,M_\odot$ - $3\times10^{10}\,M_\odot$	$2.0\times10^2\,\mathrm{s}$ - $6.0\times10^5\,\mathrm{s}$
CG2	$\frac{M}{10^{14}\,M_\odot}$	$3\times10^{10}\,M_\odot$ - $10^{14}\,M_\odot$	$6.0\times10^5\,\mathrm{s}$ - $2.0\times10^9\,\mathrm{s}$
FBC1	$\frac{10^3\,\mathrm{M}_\odot}{M}$	$10^3\,\mathrm{M}_\odot$ - $3\times10^4\,\mathrm{M}_\odot$	$2.0\times10^{-2}\,\mathrm{s}$ - $6.0\times10^{-1}\,\mathrm{s}$
FBC2	$\frac{M}{10^6\,\mathrm{M}_\odot}$	$3\times10^4\,\mathrm{M}_\odot$ - $10^6\,\mathrm{M}_\odot$	$6.0\times10^{-1}\,\mathrm{s}$ - $2.0\times10^1\,\mathrm{s}$
WMAPCMB1	$\left(\frac{M}{30\,M_\odot}\right)^{-2}$	$30\,M_\odot \quad 10^4\,M_\odot$	$6.0\times10^{-4} \quad 2.0\times10^{-1}\,\mathrm{s}$
WMAPCMB2	1×10^{-5}	$10^4\,M_\odot \quad 10^{11}\,M_\odot$	$2.0\times10^{-1} \quad 2.0\times10^6\,\mathrm{s}$
WMAPCMB3	$\frac{M}{10^{16}\,M_\odot}$	$>10^{11}\,M_\odot$	$>2.0\times10^6\,\mathrm{s}$
FIRASCMB1	$\left(\frac{M}{1\,\mathrm{M}_\odot}\right)^{-2}$	$1\,\mathrm{M}_\odot$ - $10^3\,\mathrm{M}_\odot$	$2\times10^{-5}\,\mathrm{s}$ - $2\times10^{-2}\,\mathrm{s}$
FIRASCMB2	0.015	$10^3\,\mathrm{M}_\odot$ - $10^{14}\,\mathrm{M}_\odot$	$2\times10^{-2}\,\mathrm{s}$ - $1.9\times10^9\,\mathrm{s}$
FIRASCMB3	$\frac{M}{10^{16}\,\mathrm{M}_\odot}$	$>10^{14}\,\mathrm{M}_\odot$	$>1.9\times10^9\,\mathrm{s}$
CMBDL	$10^{-1.5\log_{10}\left(\frac{M}{1\ \mathrm{g}}\right)+78.7}$	$10^{18.5}\,\mathrm{M}_\odot$ - $10^{21.5}\,\mathrm{M}_\odot$	$6.3\times10^{13}\,\mathrm{s}$ - $6.3\times10^{16}\,\mathrm{s}$

Within this range we can accommodate all the constraints from when the Universe was $\sim\ 10^{-30}$ s up to the present time.

In order to compute the observational constraints as a function of mass we always start by converting the corresponding instant of time into the corresponding horizon mass with the help of equation (1). For the default values of min1 and max1 the mass axis spans from 10^9 g ($5 \times 10^{-26} M_\odot$) to 10^{55} g ($5 \times 10^{21} M_\odot$).

By changing the values min1 and max1 we can restrict the study to a particular epoch. Additionally, by changing the stept value we can obtain a greater (or lesser) number of points for the final line of the graph. Keeping the default values we get 9401 points for the time axis.

4.2 Outputs: figures and data files

When executing the code (in Mathematica or Python) 20 outputs are generated: 12 figures + 6 data files + 2 text files, as follows:

1. **ActiveConstraints** (text):
   list of all the constraints that were enabled during execution.
2. **constraintsFigBTglobal** (figure):
   set of curves for all the enabled constraints in terms of $\beta(t)$.
3. **constraintsFigBLTglobal** (figure):
   set of curves for all the enabled constraints in terms of $\beta'(t)$.
4. **constraintsFigFTglobal** (figure):
   set of curves for all the enabled constraints in terms of $f(t)$.
5. **constraintsFigBMglobal** (figure):
   set of curves for all the enabled constraints in terms of $\beta(M)$.
6. **constraintsFigBLMglobal** (figure):
   set of curves for all the enabled constraints in terms of $\beta'(M)$.
7. **constraintsFigFMglobal** (figure):
   set of curves for all the enabled constraints in terms of $f(M)$.
8. **constraintsFigBT** (figure):
   upper limit on $\beta(t)$ combining all the enabled constraints.
9. **constraintsFigBLT** (figure):
   upper limit on $\beta'(t)$ combining all the enabled constraints.
10. **constraintsFigFT** (figure):
    upper limit on $f(t)$ combining all the enabled constraints.
11. **constraintsFigBM** (figure):
    upper limit on $\beta(M)$ combining all the enabled constraints.

12. **constraintsFigBLM** (figure):
    upper limit on $\beta'(M)$ combining all the enabled constraints.

13. **constraintsFigFM** (figure):
    upper limit on $f(M)$ combining all the enabled constraints.

14. **constraintsBT** (data):
    points for the curve $\beta(t)$ on graph **constraintsFigBT**.

15. **constraintsBLT** (data):
    points for the curve $\beta'(t)$ on graph **constraintsFigBLT**

16. **constraintsFT** (data):
    points for the curve $f(t)$ on graph **constraintsFigFT**

17. **constraintsBM** (data):
    points for the curve $\beta(M)$ on graph **constraintsFigBM**

18. **constraintsBLM** (data):
    points for the curve $\beta'(M)$ on graph **constraintsFigBLM**

19. **constraintsFM** (data):
    points for the curve $f(M)$ on graph **constraintsFigFM**

20. **RelevantConstraints** (text):
    sequence of constraints leading the curve $\beta(t)$ (or equivalently $\beta'(t)$ and $f(t)$) acrros the time axis.

In Figure 2 we present the ensemble of curves corresponding to the 55 constraints of Table 2 as a function of time as obtained with the Mathematica code and in Figure 3 the same curves obtained with the Mathematica code but now as function of mass. In Figure 4 we show the result of combining the constraints in order to get the curve giving the upper limits for the functions $\beta(t)$, $\beta'(t)$, $f(t)$ as obtained with the Mathematica code and in Figure 5 the same but now with β, β' and f as a function of mass.

In Figures 6 and 7 we show the Python version of, respectively, Figure 2 and Figure 3. In Figure 8 we show the Python version of Figure 4 and in Figure 9 the Python version of Figure 5.

In order to illustrate the contents of the data files we show in Table 5 the values for 30 points extracted from **constraintsBT** as generated with the Mathematica code. In Table 6 we show the output obtained with the Python code for the same points.

In Table 7 we show the contents of the **RelevantConstraints** text file when all the 55 constraints from Table 2 are enabled. This allows us to see which constraint is actually dominant or relevant for a given time sector.

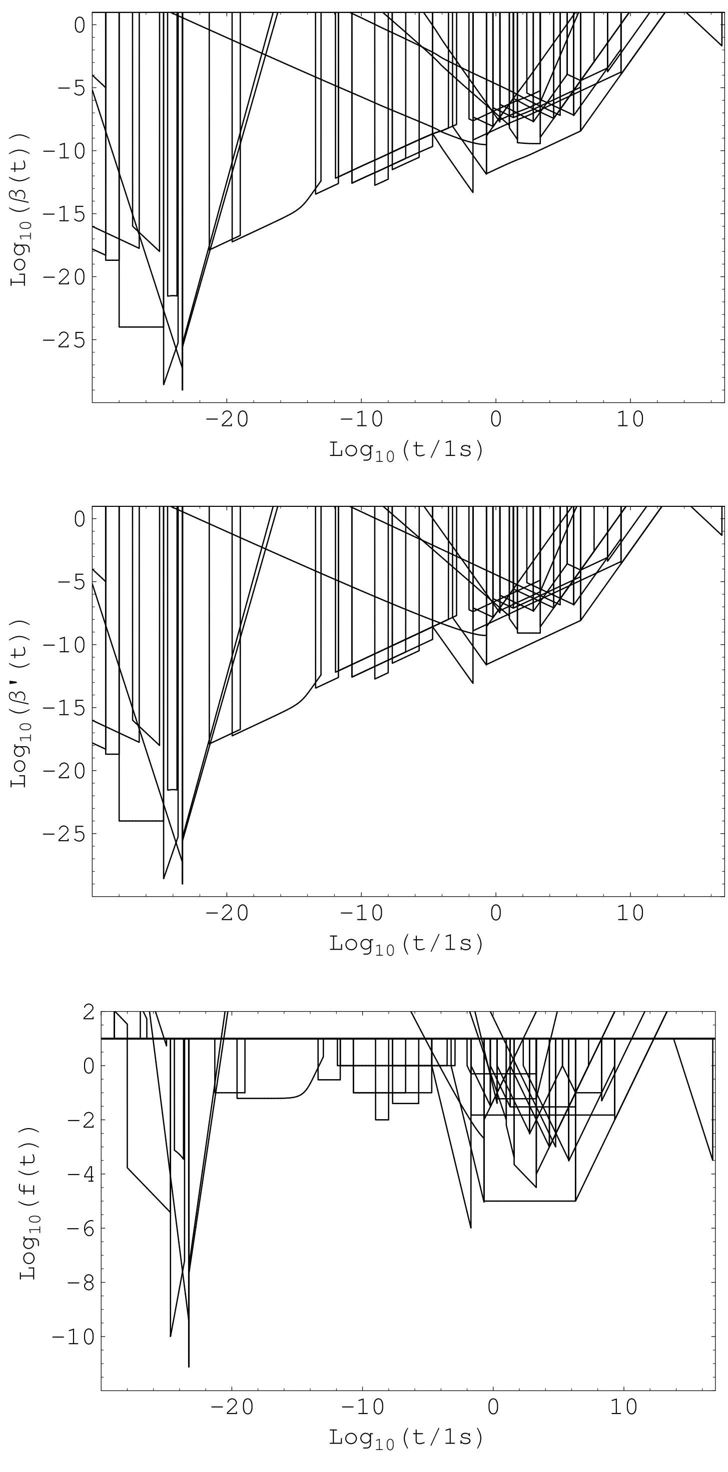

Figura 2: Constraints on (from top to bottom) $\beta(t)$, $\beta'(t)$ and $f(t)$ for a monochromatic PBH mass function. Each curve represents a particular constraint. All the 55 constraints presented on Table 2 were taken into account. All the graphs in this figure were obtained with the code written in Mathematica.

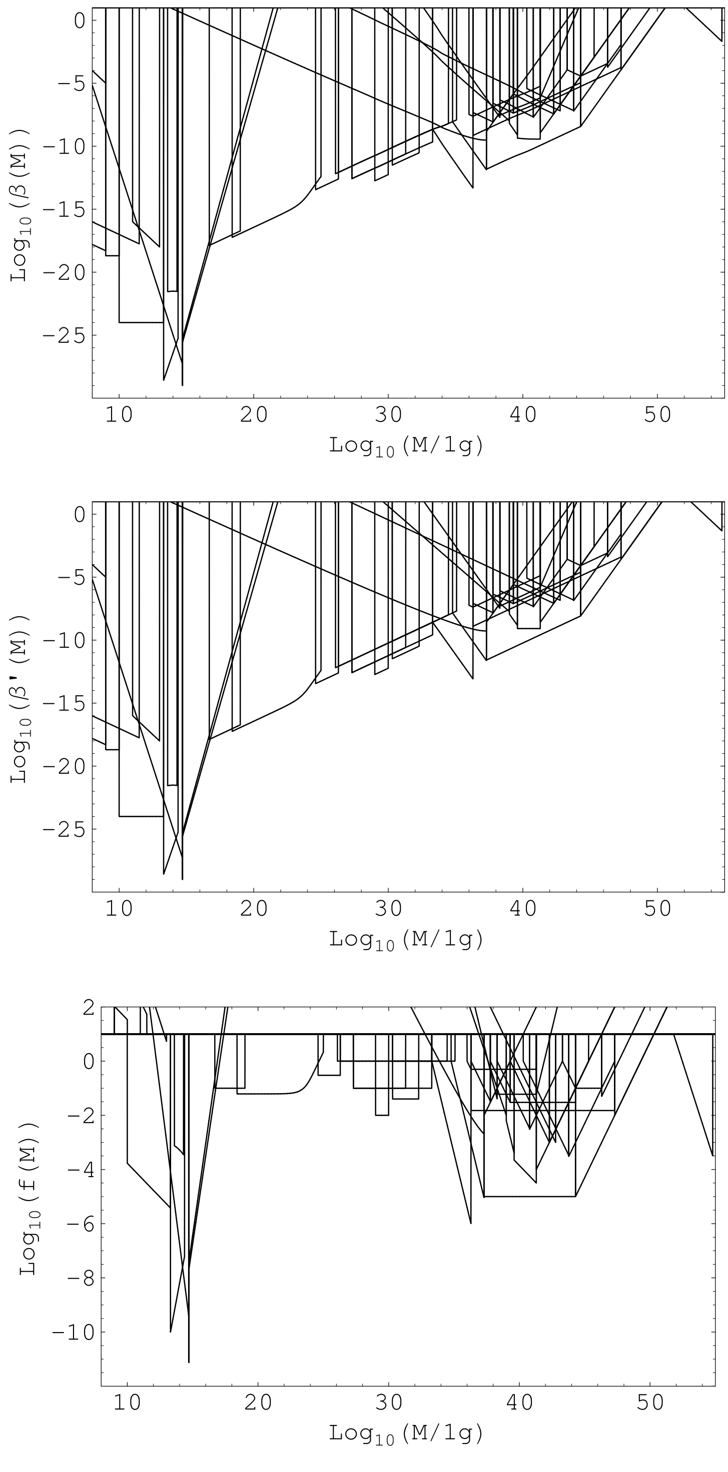

Figura 3: Constraints on (from top to bottom) $\beta(M)$, $\beta'(M)$ and $f(M)$ for a monochromatic PBH mass function. Each curve represents a particular constraint. All the 55 constraints presented on Table 2 were taken into account. All the graphs in this figure were obtained with the code written in Mathematica.

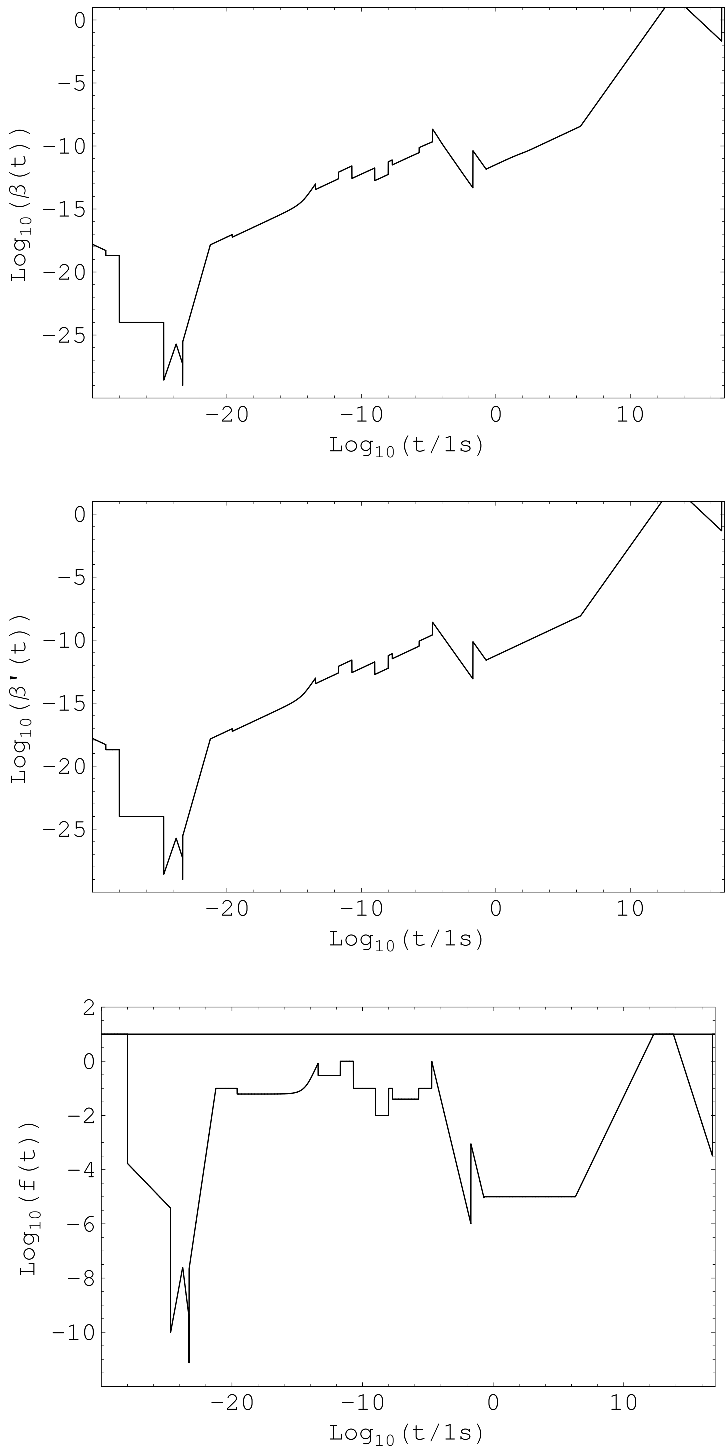

Figura 4: Combined constraints on (from top to bottom) $\beta(t)$, $\beta'(t)$ and $f(t)$ for a monochromatic PBH mass function. For each instant it was considered the constraint (among the 55 presented in Table 2) which gives the lowest value. All the graphs in this figure were obtained with the code written in Mathematica.

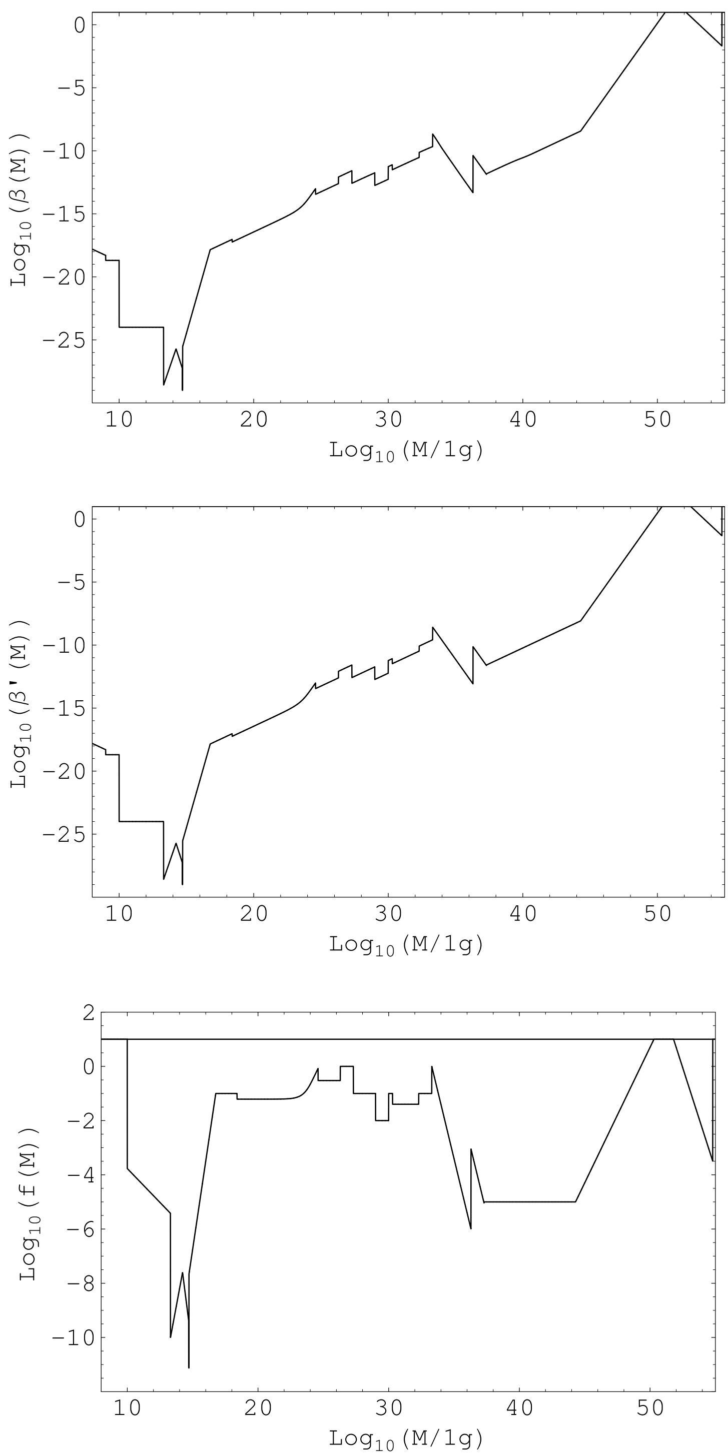

Figura 5: Combined constraints on (from top to bottom) $\beta(M)$, $\beta'(M)$ and $f(M)$ for a monochromatic PBH mass function. For each mass it was considered the constraint (among the 55 presented in Table 2) which gives the lowest value. All the graphs in this figure were obtained with the code written in Mathematica.

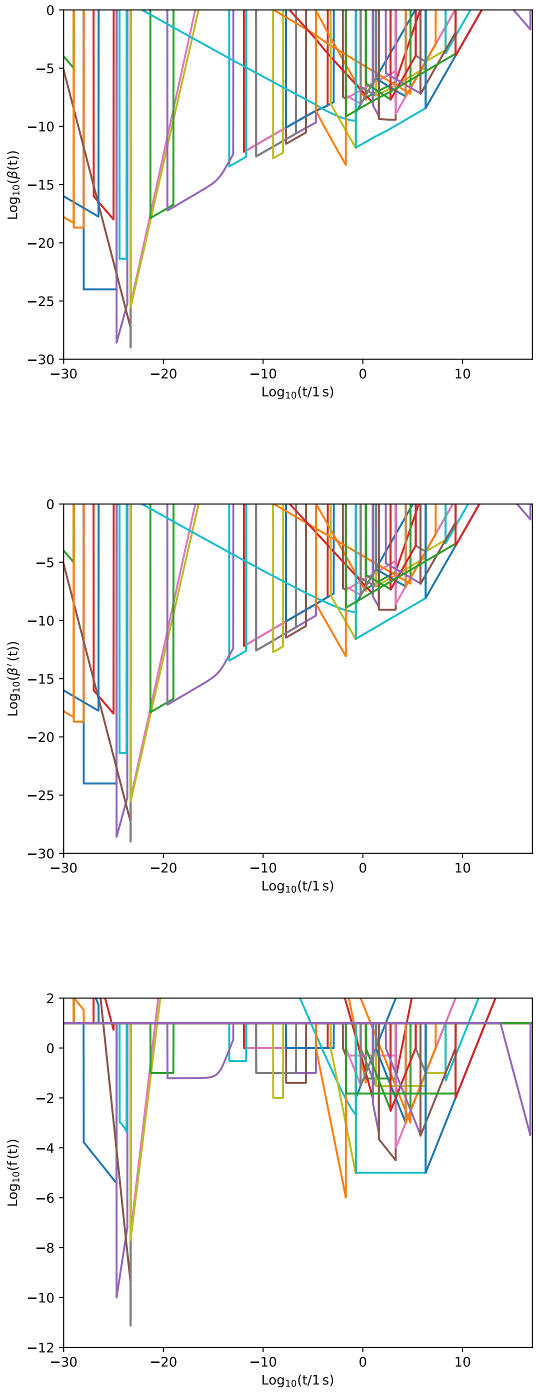

Figura 6: Constraints on (from top to bottom) $\beta(t)$, $\beta'(t)$ and $f(t)$ for a monochromatic PBH mass function. Each curve represents a particular constraint. All the 55 constraints presented on Table 2 were taken into account. All the graphs in this figure were obtained with the code written in Python.

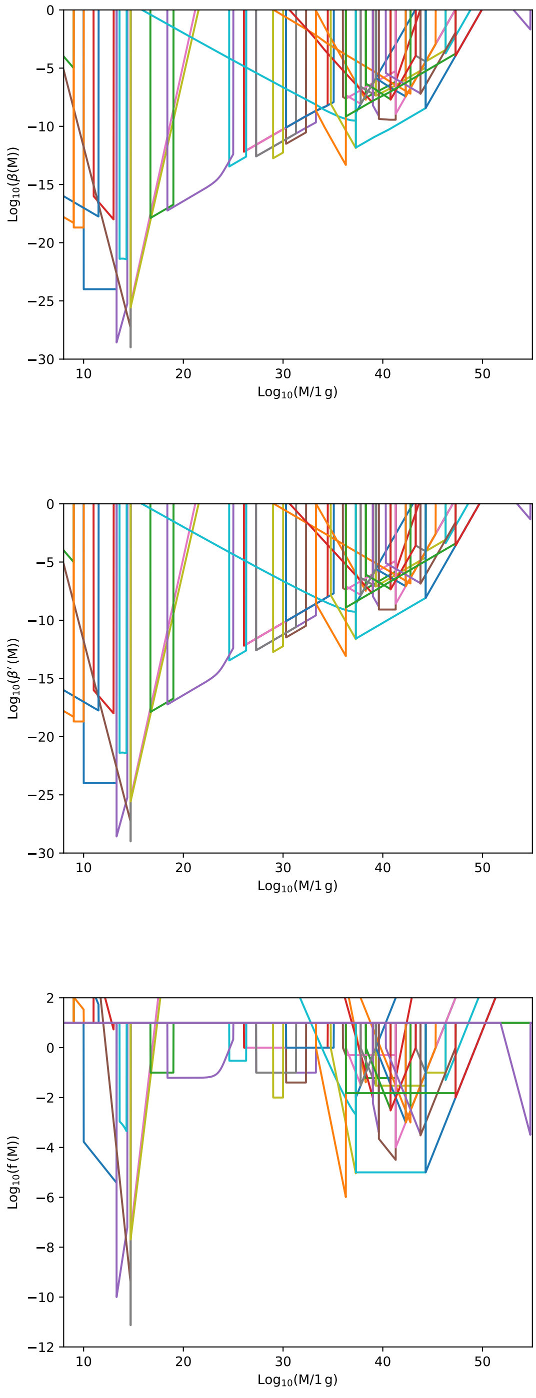

Figura 7: Constraints on (from top to bottom) $\beta(M)$, $\beta'(M)$ and $f(M)$ for a monochromatic PBH mass function. Each curve represents a particular constraint. All the 55 constraints presented on Table 2 were taken into account. All the graphs in this figure were obtained with the code written in Python.

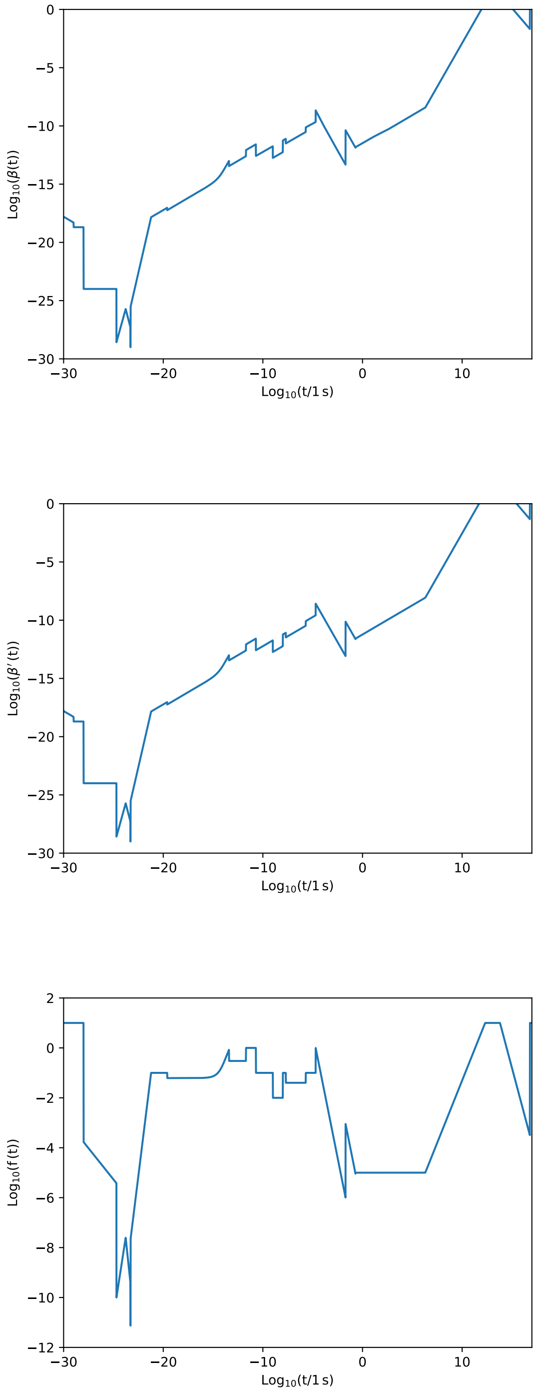

Figura 8: Combined constraints on (from top to bottom) $\beta(t)$, $\beta'(t)$ and $f(t)$ for a monochromatic PBH mass function. For each instant it was considered the constraint (among the 55 presented in Table 2) which gives the lowest value. All the graphs in this figure were obtained with the code written in Python.

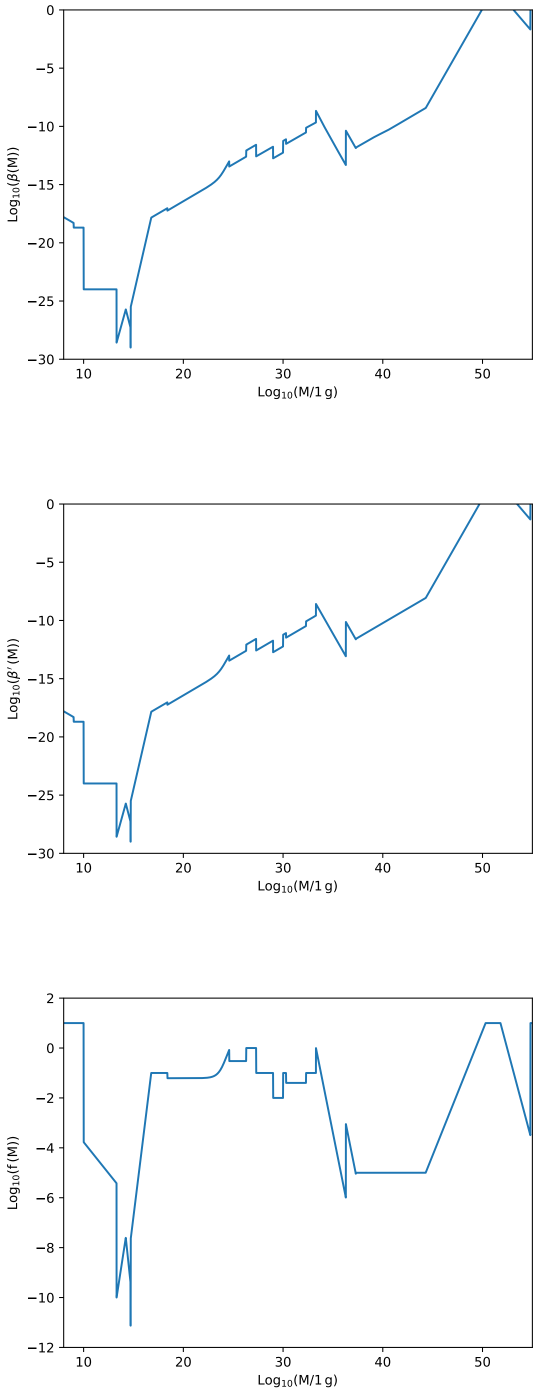

Figura 9: Combined constraints on (from top to bottom) $\beta(M)$, $\beta'(M)$ and $f(M)$ for a monochromatic PBH mass function. For each mass it was considered the constraint (among the 55 presented in Table 2) which gives the lowest value. All the graphs in this figure were obtained with the code written in Python.

Tabela 5: A sample of the values generated with the Mathematica code for the constraints on $\beta(t)$. This sample includes the point for which $\beta(t) \sim 10^{-29}$ (cf. Section 3.1).

$\log_{10}(t/1\ \mathrm{s})$	$\log_{10}(\beta(t))$
-23.400	-26.942842513263173
-23.395	-26.959342713131356
-23.390	-26.975842913132784
-23.385	-26.992343113267633
-23.380	-27.008843313536076
-23.375	-27.025343513938267
-23.370	-27.041843714474389
-23.365	-27.058343915144622
-23.360	-27.074844115949116
-23.355	-27.091344316888058
-23.350	-27.107844517961613
-23.345	-27.124344719169969
-23.340	-27.140844920513274
-23.335	-27.157345121991707
-23.330	-27.173845323605462
-23.325	-27.190345525354676
-23.320	-27.206845727239543
-23.315	-27.223345929260230
-23.310	-27.239846131416918
-23.305	-27.256346333709764
-23.300	-28.996798121598161
-23.295	-28.996798324163862
-23.290	-28.996798526866250
-23.285	-28.996798729705493
-23.280	-25.525614376573920
-23.275	-25.506864579687388
-23.270	-25.488114782938251
-23.265	-25.469364986326660
-23.260	-25.450615189852797
-23.255	-25.431865393516823

Tabela 6: A sample of the values generated with the Python code for the constraints on $\beta(t)$. This sample includes the point for which $\beta(t) \sim 10^{-29}$ (cf. Section 3.1).

$\log_{10}(t/1\text{ s})$	$\log_{10}(\beta(t))$
-23.400	-26.942842513263173
-23.395	-26.959342713131360
-23.390	-26.975842913132787
-23.385	-26.992343113267633
-23.380	-27.008843313536079
-23.375	-27.025343513938271
-23.370	-27.041843714474393
-23.365	-27.058343915144626
-23.360	-27.074844115949123
-23.355	-27.091344316888062
-23.350	-27.107844517961617
-23.345	-27.124344719169972
-23.340	-27.140844920513278
-23.335	-27.157345121991710
-23.330	-27.173845323605462
-23.325	-27.190345525354680
-23.320	-27.206845727239546
-23.315	-27.223345929260230
-23.310	-27.239846131416922
-23.305	-27.256346333709768
-23.300	-28.996798121598164
-23.295	-28.996798324163866
-23.290	-28.996798526866250
-23.285	-28.996798729705493
-23.280	-25.525614376573923
-23.275	-25.506864579687392
-23.270	-25.488114782938251
-23.265	-25.469364986326664
-23.260	-25.450615189852801
-23.255	-25.431865393516823

Tabela 7: The contents of the **RelevantConstraints** output text file when all the 55 constraints from Table 2 are enabled.

$\log_{10}(t_i/1\text{ s})$	$\log_{10}(t_f/1\text{ s})$	Constraint
-30.0	-29.0	Decay of unstable particles : e.g. gravitino, neutralino
-29.0	-28.0	Big Bang Nucleosynthesis Y
-28.0	-24.695	Big Bang Nucleosynthesis D/H
-24.695	-23.77	CMB small scale anisotropies
-23.77	-23.3	Extragalactic gamma - ray background 1 (evaporated)
-23.3	-23.28	Extragalactic gamma - ray background 3 (exploding)
-23.28	-21.23	Extragalactic gamma-ray background 4 (evaporating)
-21.23	-19.6	Femtolensing of GRBs
-19.6	-13.4	Capture of PBHs by white dwarfs and neutron stars
-13.4	-11.7	Kepler microlensing of stars
-11.7	-10.7	OGLE microlensing of stars 1
-10.7	-7.7	OGLE microlensing of stars 2
-7.7	-5.7	EROS microlensing of stars 3
-5.7	-4.7	EROS microlensing of stars 2
-4.7	-1.7	FIRAS - CMB anisotropies/distortions 1
-1.7	-0.7	WMAP - CMB anisotropies/distortions 1
-0.7	6.3	WMAP - CMB anisotropies/distortions 2
6.3	9.3	WMAP - CMB anisotropies/distortions 3
9.3	12.59	FIRAS - CMB anisotropies/distortions 3
12.59	14.12	Empty
14.12	16.8	CMB dipole limit
16.8	17	Empty

4.3 How to add a new constraint

Suppose that we want to introduce a new constraint into the system. Let us consider, as an example, the constraint given by the expression:

$$f(M) < \left(\frac{M}{10^5 M_\odot}\right)^{\phi-1} \tag{28}$$

valid for the mass range 10^{19} g - $10^3 M_\odot$ and with $\phi \approx \sqrt{2}$. Let's name this constraint **Yet Another Constraint** and associate it with the acronym **YAC**. The acronym will also serve as the name of the boolean variable that defines whether the constraint is enabled or disabled.

4.3.1 Wolfram Mathematica

To implement this constraint in the Wolfram Mathematica language program we start by writing a block of code like the one shown in Figure 10. Basically, we can take a block of code associated with one of the constraints already implemented and make the necessary changes. The changes to be made are those highlighted in blue in Figure 10. More specifically we must:

- in the **If** condition insert the name of the boolean type variable (which matches the constraint acronym).
- assign the name of the constraint to the **cName** variable.
- implement the mathematical expression that defines the constraint (in this example given by equation 28) in the body of the so-called **constraint** function.
- insert the value of the lowest mass (in grams) as the argument of the **th** function in the evaluation of **tmin** (in the absence of a lower limit for the mass just write **tmin = min1**).
- insert the value of the highest mass (in grams) as the argument of the **th** function in the evaluation of **tmax** (in the absence of an upper limit for the mass just write **tmax = max1**)
- any additional parameters can be added to the code (e.g. **phi = Sqrt[2];**).

The block shown on Figure 10 should be inserted into Section [**6 - List of Constraints**] of the main code (the order with respect to the other constraints already implemented is not relevant). Additionaly in Section [**2 - Enable / Disable Constraints**] we must insert the new boolean type variable **YAC** assigning it with the value **0** or **1**.

Notice that the program is prepared to work with constraints given in terms of the function $f(M)$ (or equivalently $f(t)$, in which case it is computed as $f(\mathrm{th}(M))$). If the constraint is given in terms of the function $\beta(M)$ or $\beta'(M)$ (that can happen if we were talking about a mass range for which all PBHs have already evaporated) then we must convert it to the $f(M)$ form according to equations (10) or (11). This was, for example, the case of the **EGB1** constraint (cf. Table 3).

In order to test the new constraint in isolation first we must disable all the other constraints (Section [**2 - Enable / Disable Constraints**]), enabling only the new constraint (**YAC = 1**). Running the code we get the 12 Figures, six data files and two text files (cf. Section 4.2) accounting for the new constraint. As an example we show in Figure 11 the $\beta(M)$ and $\beta(t)$ curves obtained with Mathematica.

4.3.2 Python

To implement this constraint in the Python code, we proceed analogously to the approach taken in Mathematica, while adapting the syntax and structure to the Python environment. We take an existing code block for a previously implemented constraint and make the necessary modifications (Figure 12). In detail, the following steps should be followed:

```
(* YAC - Yet Another Constraint *)
(* ref: someone S. et al., 2025, Astro, 99, 01 [eq.xx] *)
(* MASS RANGE: 10^19 g < m < 10^3 ms *)
If[YAC == 1,
  phi = Sqrt[2];
  cName = "Yet Another Constraint";
  constraint = Function[m, (m / (10^5 * ms)) ^ (phi - 1)];
  tmin = Log[10, th[10^19]] ; tmax = Log[10, th[10^3 * ms]] ;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

Figura 10: The code block to implement the hypothetical new constraint YAC (Yet Another Constraint) in the Mathematica version of the program. The parts that should be changed, in comparison to the code blocks used for other constraints already implemented, are those marked in blue (see text for more details).

- In Mathematica, the constraint is conditionally activated with **If[YAC = 1, ...]**. In Python, this becomes **if constraints["YAC"] == 1:**, using the dictionary constraints as a central control mechanism for toggling constraints.
- assign the name of the constraint to the **cName** variable.
- implement the mathematical expression that defines the new constraint (in this example given by equation 28) in the body of the so-called **constraint** function using `def` and the convenient Python mathematical operators.
- insert the value of the lowest mass (in grams) as the argument of the **th** function in the evaluation of **tmin** (in the absence of a lower limit for the mass just write **tmin = min1**).
- insert the value of the highest mass (in grams) as the argument of the **th** function in the evaluation of **tmax** (in the absence of an upper limit for the mass just write **tmax = max1**)
- The declaration `phi = Sqrt[2]` becomes `phi = sqrt(2)` in Python, corresponding to $\phi = \sqrt{2}$, assuming `sqrt` is already imported from the appropriate module (e.g., `math` or `numpy`).
- In the Python version, the result of the constraint evaluation is added to the global list using `Global.append(FB(tmin, tmax, YAC))`. This replaces the Mathematica instruction `Global Append[Global, tabFC]`, integrating the constraint output into the system in a syntactically appropriate manner.

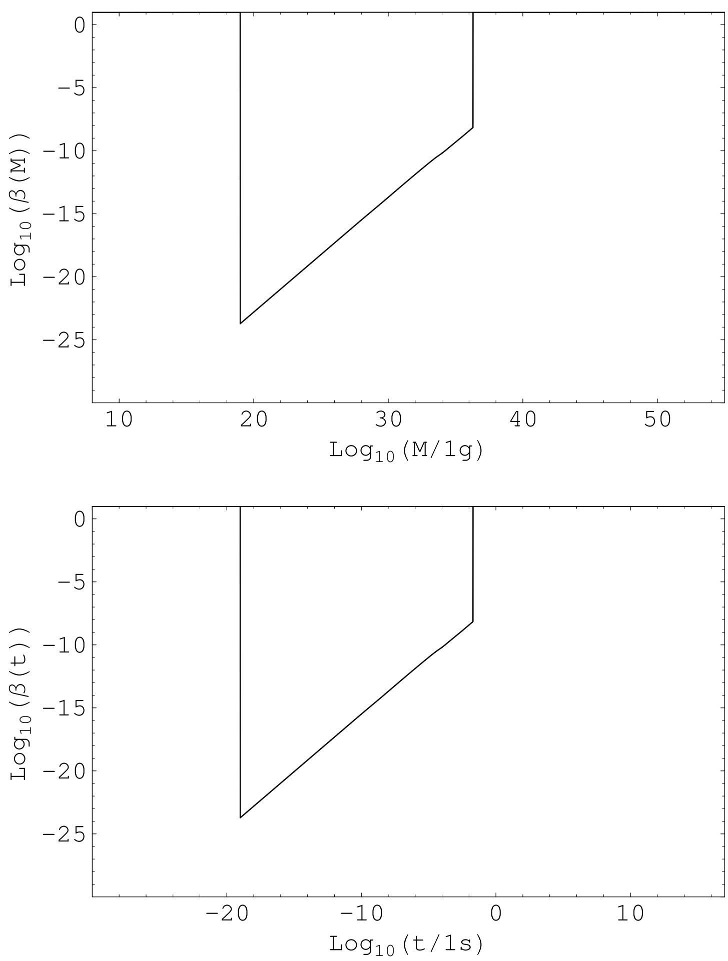

Figura 11: The functions $\beta(M)$ (top) and $\beta(t)$ (bottom) for the new hypothetical constraint YAC (cf. equation 28) obtained with the Mathematica code.

The new code block should be placed in section [**6 – List of Constraints**] of the main Python script. Furthermore, the constraint must be registered in section [**2 – Enable / Disable Constraints**] by adding the corresponding entry to the `constraints` dictionary. For example:

```
constraints = {
"BBNDH": 1, # Big Bang Nucleosynthesis D/H
"BBNY": 1, # Big Bang Nucleosynthesis Y_p
... other constraints
"YAC": 1, # Yet Another Constraint
}
```

This enables the constraint by default; setting its value to `0` would disable it. As

```
# YAC: Yet Another Constraint
# ref: someone S. et al., 2025, Astro, 99, 01 [eq.xx]
# MASS RANGE: 10^19 g < m < 10^3 ms
if constraints["YAC"] == 1:
    phi=sqrt(2)
    cName = "Yet Another Constraint"
    def YAC(m):
        return (m/(10**5*ms))**(phi-1)
    tmin=log10(th(10**19))
    tmax=log10(th(10**3*ms))
    Global.append(FB(tmin,tmax,YAC))
    ActiveConstraints.append(cName)
```

Figura 12: The code block to implement the hypothetical new constraint YAC (Yet Another Constraint) in the Python version of the program. The parts that should be changed, in comparison to the code blocks used for other constraints already implemented, are those marked in blue (see text for more details).

previously noted, the position in which the new constraint is added within the dictionary is irrelevant to the programs execution.

4.4 How to remove an existing constraint

In order to exclude temporarily an existing constraint from the computation, just disable the corresponding boolean variable. In order to remove it permanently, just delete the corresponding block code.

5 Conclusions

PBHs might have formed in the early Universe due, for example, to the gravitational collapse of density fluctuations. These PBHs could have masses as low as the Planck Mass ($\sim 10^{-5}$ g) or as high as $\sim 10^{12} M_\odot$ or even higher. To the present day we are not aware of the detection of a single PBH. It is not ruled out the possible primordial origin of at least some of the observed BHs or BH candidates (all of them with masses within the stellar range or higher). To date, we are not aware of the detection of any BH candidate with substellar mass. Nevertheless, there is a set of observational constraints limiting the possible number density allowed for PBHs at different mass ranges.

In this work we have considered the constraints on PBHs with masses ranging from 10^9 g ($5 \times 10^{-26} M_\odot$) up to 10^{55} g ($5 \times 10^{21} M_\odot$). We have developed a code capable of plotting the observational constraints on the number density of PBHs within the mentioned mass range. More specifically, we have written a version of the code in Wolfram Mathematica and another one in Python. The code, which has a similar structure in both versions, is written in a simple and flexible way, allowing the introduction of new constraints as well as the eventual update/refinement of the existing ones.

In terms of future work, we intend to use this code (or the output it provides) in combination with other applications used to determine the numerical density of PBHs formed in different scenarios. We also intend to add new constraints or improve the implementation of existing ones whenever necessary.

A The implementation code in Wolfram Mathematica

Here we present the code implemented in the Wolfram Mathematica language (Wolfram Mathematica, 2005). The code is divided into 11 sections as follows:

1. **Initial technical settings**: set current directory.
2. **Enable/Disable Constraints**: list of boolean type variables indicating whether each constraint is enabled (1) or disabled (0) - see Section 3
3. **Main parameters and functions**: parameters, physical constants and functions that are common to several constraints.
4. **Degrees of freedom g(T)**: functions related to the existence of different particle species (see Section 2).
5. **Functions beta, betaprime and f**: the sector where the functions β, β' and f are implemented, together with the function **FB** which controls all the calculations and data store for further use.
6. **List of Constraints**: this sections is subdivided in several blocks, one for each constraint. The standard structure of these blocks is detailed in Section 4.3 (where we explained how to add a new constraint).
7. **Frames for the different figures**: six frames for the six types of graphs that will be returned as outputs [$\beta(t)$, $\beta'(t)$, $f(t)$, $\beta(M)$, $\beta'(M)$, $f(M)$].
8. **Generating the global figures**: the sector where de data for the global graphs (see Figures 2, 3, 6 and 7) is selected.
9. **Display all figures**: display all graphs on screen.
10. **Generate figures with the minimum values only**: the sector where de data for the combined graphs (see Figures 4, 5, 8 and 9) is determined.
11. **SAVE FILES and FIGURES**: saving to the current directory the 20 outputs mentioned in Section 4.2.

```
(* OBSERVATIONAL CONSTRAINTS
  ON PRIMORDIAL BLACK HOLES *)
(* Faculdade de Ciências Exatas e da Engenharia da Universidade da Madeira *)
(* Grupo de Astronomia da Universidade da Madeira - GAUMa *)
(* Sobrinho J.L.G. & Jesus P.D.F., June 2025 *)
(* Version for: Wolfram Mathematica 5.0 *)

(*
 Table of Contents:;
 01-Initial technical settings;
 02-Enable/Disable Constraints;
 03-Main parameters and functions;
 04-Degrees of freedom g (T);
 05-Functions beta,betaprime and f;
 06-List of Constraints;
 07-Frames for the different figures;
 08-Generating the global figures;
 09-Display global figures;
 10-Generate figures with the minimum values only;
 11-SAVE FILES and FIGURES;
 *)

(* 1 - Initial technical settings *)

(* current directory *)
SetDirectory["C:\Users\Unknown\Folder"];

(* 2 - Enable / Disable Constraints *)

BBNDH = 1; (*Big Bang Nucleosynthesis D/H*)
BBNY = 1; (*Big Bang Nucleosynthesis Y*)
GE = 1; (*Generation of entropy*)
CMBSD = 1; (*CMB spectral distortions*)
CMBA = 1; (*CMB small scale anisotropies*)
EGB1 = 1; (*Extragalactic gamma-ray background 1 (evaporated)*)
EGB2 = 1; (*Extragalactic gamma-ray background 2 (evaporated)*)
EGB3 = 1; (*Extragalactic gamma-ray background 3 (exploding)*)
EGB4 = 1; (*Extragalactic gamma-ray background 4 (evaporating)*)
EGAP = 1; (*Extragalactic antiprotons*)
LSP = 1; (*Emission of the Lightest Supersymmetric Particle LSP*)
DUP = 1; (*Decay of unstable particles:e.g.gravitino,neutralino*)
GRBFL = 1; (*Femtolensing of GRBs*)
EROSMLS1 = 1; (*EROS microlensing of stars 1*)
EROSMLS2 = 1; (*EROS microlensing of stars 2*)
EROSMLS3 = 1; (*EROS microlensing of stars 3*)
OGLEMLS1 = 1; (*OGLE microlensing of stars 1*)
OGLEMLS2 = 1; (*OGLE microlensing of stars 2*)
OGLEMLS3 = 1; (*OGLE microlensing of stars 3*)
KeplerMLS = 1; (*Kepler microlensing of stars*)
QMLS = 1; (*Quasar microlensing*)
CRSMLS1 = 1; (*Millilensing of compact radio sources 1*)
CRSMLS2 = 1; (*Millilensing of compact radio sources 2*)
```

```
CRSMLS3 = 1; (*Millilensing of compact radio sources 3*)
WDNS = 1; (*Capture of PBHs by white dwarfs and neutron stars*)
WBS1 = 1; (*Wide-binary stability 1*)
WBS2 = 1; (*Wide-binary stability 2*)
DGC1 = 1; (*Disruption of globular clusters 1*)
DGC2 = 1; (*Disruption of globular clusters 2*)
ERI1 = 1; (*Eridanus II-survival of star clusters 1*)
ERI2 = 1; (*Eridanus II-survival of star clusters 2*)
DHTS1 = 1; (*Disk heating and tidal streams 1*)
DHTS2 = 1; (*Disk heating and tidal streams 2*)
DFHO1 = 1; (*Dynamical friction on halo objects 1*)
DFHO2 = 1; (*Dynamical friction on halo objects 2*)
DFHO3 = 1; (*Dynamical friction on halo objects 3*)
DFHO4 = 1; (*Dynamical friction on halo objects 4*)
DDGC1 = 1; (*Disruption/distortion of galaxies in clusters 1*)
DDGC2 = 1; (*Disruption/distortion of galaxies in clusters 2*)
DDGC3 = 1; (*Disruption/distortion of galaxies in clusters 3*)
MWG1 = 1; (*MW type galaxies 1*)
MWG2 = 1; (*MW type galaxies 2*)
DG1 = 1; (*Dwarf type galaxies 1*)
DG2 = 1; (*Dwarf type galaxies 2*)
CG1 = 1; (*Clusters of galaxies 1*)
CG2 = 1; (*Clusters of galaxies 2*)
FBC1 = 1; (*First baryonic clouds 1*)
FBC2 = 1; (*First baryonic clouds 2*)
WMAPCMB1 = 1; (*WMAP-CMB anisotropies/distortions 1*)
WMAPCMB2 = 1; (*WMAP-CMB anisotropies/distortions 2*)
WMAPCMB3 = 1; (*WMAP-CMB anisotropies/distortions 3*)
FIRASCMB1 = 1; (*FIRAS-CMB anisotropies/distortions 1*)
FIRASCMB2 = 1; (*FIRAS-CMB anisotropies/distortions 2*)
FIRASCMB3 = 1; (*FIRAS-CMB anisotropies/distortions 3*)
CMBDL = 1; (*CMB dipole limit*)

(* constraints default values *)
defaultValue = 1;
defaultName = "empty";
```

(* 3 - Main parameters and functions *)

```
(* Sun mass *)
ms = 1.989 * 10^33 ; (* g *)

(* MW galaxy halo mass *)
(* ref: Gardner S.,McDermott S.D.,Yanny B.,2021,PrPNP,121,103904. *)
mhalo = 10^12 * ms; (* g *)

(* Mass for which the PBH lifetime equals the age of the Universe *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.16] *)
mx = 5.1 * 10^14; (* g *)

(* Horizon mass mh (g) at time t (s) *)
(* ref: e.g. Sobrinho J.L.G., Augusto P.,
 Gonçalves A.L., 2016, MNRAS, 463, 2348 [eq.02] *)
mh = Function[t, 10^15 * (t / (10^-23))]; (* g *)
```

```
(* Instant of time th (s) for which the horizon mass is m (g) *)
th = Function[m, m / (10^38)]; (* s *)

(* Converts time in seconds to temperature in GeV *)
Ttime = Function[ti, 0.00121 / Sqrt[ti]]; (* GeV *)

(* logarithmic step of time for the X-axis *)
stept = 0.005;

(* time interval considered on a logarithmic scale *)
min1 = -30; (* log (th/1 s) minimum value *)
max1 = 17; (* log (th/1 s) maximum value *)

(*  X-axis - log (time/1 s) *)
logTime = Table[i, {i, min1, max1, stept}];

(*  X-axis - log (mass/1 g) *)
logMass = Table[Log[10, mh[10^logTime[[i]]]], {i, 1, Length[logTime]}];
```

(* 4 - Degrees of freedom g (T) *)

(* T in GeV *)

```
(* Parameters for the numerical
   approximation of g (T) when 120 MeV < T < 10^16 GeV *)
(* ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 [Table 1] *)
a = {1, 1.11724 * 10^0, 3.12672 * 10^-1, -4.68049 * 10^-2,
   -2.65004 * 10^-2, -1.19760 * 10^-3, 1.82812 * 10^-4, 1.36436 * 10^-4,
   8.55051 * 10^-5, 1.22840 * 10^-5, 3.82259 * 10^-7, -6.87035 * 10^-9};
b = {1.43382 * 10^-2, 1.37559 * 10^-2, 2.92108 * 10^-3, -5.38533 * 10^-4,
   -1.62496 * 10^-4, -2.87906 * 10^-5, -3.84278 * 10^-6, 2.78776 * 10^-6,
   7.40342 * 10^-7, 1.17210 * 10^-7, 3.72499 * 10^-9, -6.74107 * 10^-11};
c = {1, 6.07869 * 10^-1, -1.54485 * 10^-1, -2.24034 * 10^-1,
   -2.82147 * 10^-2, 2.90620 * 10^-2, 6.86778 * 10^-3, -1.00005 * 10^-3,
   -1.69104 * 10^-4, 1.06301 * 10^-5, 1.69528 * 10^-6, -9.33311 * 10^-8};
d = {7.07388 * 10^1, 9.18011 * 10^1, 3.31892 * 10^1, -1.39779 * 10^0,
   -1.52558 * 10^0, -1.97857 * 10^-2, -1.60146 * 10^-1, 8.22615 * 10^-5,
   2.02651 * 10^-2, -1.82134 * 10^-5, 7.83943 * 10^-5, 7.13518 * 10^-5};

(* Degrees of freedom of the energy density *)
(* High energy domain  - 120 MeV < T < 10^16 GeV *)
(* ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 [eq. C.1] *)
grH[T_] := Sum[a[[i + 1]] * (Log[E, T]) ^i, {i, 0, 11}] /
   Sum[b[[i + 1]] * (Log[E, T]) ^i, {i, 0, 11}];

(* Degrees of freedom of the entropy density *)
(* High energy domain  - 120 MeV < T < 10^16 GeV *)
(* ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 [eq. C.2] *)
gsH[T_] := grH[T] / (1 + (Sum[c[[i + 1]] * (Log[E, T]) ^i, {i, 0, 11}] /
       Sum[d[[i + 1]] * (Log[E, T]) ^i, {i, 0, 11}]));
```

```
(* Fitting functions for the Low energy domain -  T < 120 MeV *)
(* ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 [eq. C.5 to C.9] *)
fp[x_] := E^(-1.04855 x) (1 + 1.03757 x + 0.508630 x^2 + 0.0893988 x^3);
bp[x_] := E^(-1.03149 x) (1 + 1.03317 x + 0.398264 x^2 + 0.0648056 x^3);
fs[x_] := E^(-1.04190 x) (1 + 1.03400 x + 0.456426 x^2 + 0.0595248 x^3);
bs[x_] := E^(-1.03365 x) (1 + 1.03397 x + 0.342548 x^2 + 0.0506182 x^3);
Sfit[x_] := 1 + 7 / 4 * E^(-1.0419 x) (1 + 1.034 x + 0.456426 x^2 + 0.0595249 x^3)
(* constants *)
me = 511 * 10^-6;
mu = 0.1056;
mpi0 = 0.135;
mpipm = 0.140;
m1 = 0.5; m2 = 0.77; m3 = 1.2; m4 = 2;

(* Degrees of freedom of the energy density *)
(* Low energy domain  - T < 120 MeV *)
(* ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 [eq. C.3] *)
grL[T_] := 2.030 + 1.353 * (Sfit[me / T]) ^ (4 / 3) +
   3.495 * fp[me / T] + 3.446 * fp[mu / T] + 1.05 * bp[mpi0 / T] + 2.08 * bp[mpipm / T] +
   4.165 * bp[m1 / T] + 30.55 * bp[m2 / T] + 80.4 * bp[m3 / T] + 8209 * bp[m4 / T];

(* Degrees of freedom of the entropy density *)
(* Low energy domain  - T < 120 MeV *)
(* ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 [eq. C.4] *)
gsL[T_] := 2.008 + 1.923 * Sfit[me / T] + 3.442 * fs[me / T] +
   3.468 * fs[mu / T] + 1.034 * bs[mpi0 / T] + 2.068 * bs[mpipm / T] +
   4.16 * bs[m1 / T] + 30.55 * bs[m2 / T] + 90 * bs[m3 / T] + 6209 * bs[m4 / T];

(* Global degrees of freedom High/Low energy domain *)
gr[T_] := If[T < 0.12, grL[T], grH[T]];
gs[T_] := If[T < 0.12, gsL[T], gsH[T]];


(* Contributions of the different
  species of particles to the PBH evaporation process *)
(* ref: Gibilisco M., 1997, IJMPA, 12, 4167 [eq. 2.3] *)
fx[m_] := 1.569 +
   0.569 * (E^(-m / (4.53 * 10^14)) + 6 E^(-m / (1.60 * 10^14)) + 3 E^(-m / (9.6 * 10^13)) +
      3 E^(-m / (2.56 * 10^13)) + E^(-m / (2.68 * 10^13)) + 3 E^(-m / (9.07 * 10^12)) +
      3 E^(-m / (0.48 * 10^12))) + 0.963 * E^(-m / (1.10 * 10^14));
```

(* 5 - Functions beta, betaprime and f *)

```
(* fraction f (M) of the halo in PBHs as a funtion of betaprime (M) *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.57] *)
f = Function[{mi, betalinhai}, 3.81 * 10^8 * betalinhai * 1 / Sqrt[mi / ms]];

(* betaprime (M) as a function of f (M) *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.57] *)
betaLinha = Function[{mi, fi}, fi / (3.81 * 10^8) * Sqrt[mi / ms]];

(* numerical factor somewhat below unity *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.4,5] *)
```

```
gamma = 1;

(* Hubble parameter present day value: H0=100 h km/s/Mpc *)
(* ref: Particle Data Group - Cosmological Parameters *)
(* Revised September 2021 by O.Lahav and A.R.Liddle *)
h = 0.674;

(* beta (t) as a function of betaprime (t) *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.8] *)
beta = Function[{betalinhai, ti},
  1 / Sqrt[gamma] * (gs[Ttime[ti]] / 106.75) ^ (1 / 4) * (h / 0.67) ^2 * betalinhai];
```

(* Compute: f, betaprime and beta *)

```
(* INPUT' s: *)
(* min0 = log (tmin/1 s) for this constraint *)
(* max0 = log (tmax/1 s) for this constraint *)
(* f0 = function which defines the constraint *)

(* OUTPUT: *)
(* list with lists of the form: *)
(* {log (t/1 s),log (m/1 g),log (f),log (betaLinha),log (beta),constraint name} *)
(* For points outside the constraints application domain it is returned: *)
(* {log (t/1 s),log (m/1 s),defaultValue,defaultValue,defaultValue,defaultName} *)

FB = Function[{min0, max0, f0},

   i = 1;
   res = {};
   While[i <= Length[logTime],

     If[logTime[[i]] < min0 || logTime[[i]] > max0,

      res = Append[res,
        {
         logTime[[i]],
         logMass[[i]],
         defaultValue,
         defaultValue,
         defaultValue,
         defaultName
        }],

      res = Append[res,
        {
         logTime[[i]],
         logMass[[i]],
         taux = 10^logTime[[i]];
         Log[10, f0[mh[taux]]],
         Log[10, betaLinha[mh[taux], f0[mh[taux]]]],
         Log[10, beta[betaLinha[mh[taux], f0[mh[taux]]], taux]] ,
         cName
        }
```

```
      ]];
     i = i + 1;
    ]; res
   ];
```

```
(* 6 - List of Constraints *)

Global = {}; (* list to save the results of all the constraints *)
ActiveConstraints = {};
(* list to store the names of all considered constraints *)
```

```
(* BBNDH - Big Bang Nucleosynthesis D/H *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [Figs. 4 and 5] *)
(* MASS RANGE D/H:   10^10 g < m < 10^13.3 g *)
(* within this mass range we have betaprime = 10^-24 (approx.) *)
If[BBNDH == 1,
 cName = "Big Bang Nucleosynthesis D/H";
 constraint = Function[m, 3.81 * 10^8 * 10^-24 * (m / ms) ^ (-1 / 2)];
 tmin = Log[10, th[10^10]]; tmax = Log[10, th[10^13.3]];
 tabFC = FB[tmin, tmax, constraint];
 Global = Append[Global, tabFC];
 ActiveConstraints = Append[ActiveConstraints, cName]
];
```

```
(* BBNY - Big Bang Nucleosynthesis Y *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [Figs. 4 and 5] *)
(* MASS RANGE Y: 10^9 g < m < 10^10 g *)
(* within this mass range we have betaprime = 2*10^-19 (approx.) *)
If[BBNY == 1,
  cName = "Big Bang Nucleosynthesis Y";
  constraint = Function[m, 3.81 * 10^8 * 2 * 10^-19 * (m / ms) ^ (-1 / 2)];
  tmin = Log[10, th[10^9]]; tmax = Log[10, th[10^10]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* GE - Generation of entropy *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.24] *)
(* MASS RANGE:  m < 10^9 g  - não tem limite inferior *)
If[GE == 1,
  cName = "Generation of entropy";
  constraint = Function[m, 3.81 * 10^8 * 10^-5 * (m / (10^9)) ^ (-1) * (m / ms) ^ (-1 / 2)];
  tmin = min1; tmax = Log[10, th[10^9]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* CMBSD - CMB spectral distortions *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.25] *)
(* MASS RANGE:  10^11 g < m < 10^13 g *)
```

```
If[CMBSD == 1,
  cName = "CMB spectral distortions";
  constraint = Function[m, 3.81 * 10^8 * 10^-16 * (m / (10^11)) ^ (-1) * (m / ms) ^ (-1 / 2)];
  tmin = Log[10, th[10^11]]; tmax = Log[10, th[10^13]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* CMBA - CMB small scale anisotropies *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [Eq.28] *)
(* MASS RANGE:  2.0*10^13 g < m < 2.4*10^14 g *)
If[CMBA == 1,
  cName = "CMB small scale anisotropies";
  fh = 0.1;
  constraint = Function[m,
    3.81 * 10^8 * 3 * 10^-30 * (fh / 0.1) ^ (-1) * (m / (10^13)) ^3.1 * (m / ms) ^ (-1 / 2)];
  tmin = Log[10, th[2 * 10^13]]; tmax = Log[10, th[2.4 * 10^14]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* EGB1 - Extragalactic gamma-ray background 1 (evaporated) *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [Eq.32] *)
(* MASS RANGE:  m < mx g *)
If[EGB1 == 1,
  cName = "Extragalactic gamma-ray background 1 (evaporated)";
  epsilon = 0.4;
  constraint =
   Function[m, 3.81 * 10^8 * 5 * 10^-28 * (m / mx) ^ (-5 / 2 - 2 * epsilon) * (m / ms) ^ (-1 / 2)];
  tmin = min1; tmax = Log[10, th[mx]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* EGB2 - Extragalactic gamma-ray background 2 (evaporated) *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [Eq.33] *)
(* MASS RANGE:  m > mx g *)
If[EGB2 == 1,
  cName = "Extragalactic gamma-ray background 2 (evaporated)";
  epsilon = 0.4;
  constraint =
   Function[m, 3.81 * 10^8 * 5 * 10^-26 * (m / mx) ^ (7 / 2 + epsilon) * (m / ms) ^ (-1 / 2)];
  tmin = Log[10, th[mx]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* EGB3 - Extragalactic gamma-
  ray background 3 (exploding) *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [Fig.18 - betaprime~10^-29] *)
(* MASS RANGE:  aprox mx *)
If[EGB3 == 1,
  cName = "Extragalactic gamma-ray background 3 (exploding)";
  EGB3width = 1.75 / 100; (* % *)
  constraint = Function[m, 3.81 * 10^8 * 10^-29 * (m / ms) ^ (-1 / 2)];
  tmin = Log[10, th[mx - mx * EGB3width]]; tmax = Log[10, th[mx + mx * EGB3width]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* EGB4 - Extragalactic gamma-
  ray background 4 (evaporating) *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.62] *)
(* MASS RANGE: m > mx  *)
If[EGB4 == 1,
  cName = "Extragalactic gamma-ray background 4 (evaporating)";
  epsilon = 0.25; (* 0.1 < epsilon < 0.4 *)
  constraint = Function[m, 2 * 10^(-8) * (m / (mx)) ^ (3 + epsilon)];
  tmin = Log[10, th[mx]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* EGAP - Extragalactic antiprotons *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.48] *)
(* MASS RANGE:  4*10^13 g < m < 2*10^14 g *)
If[EGAP == 1,
  cName = "Extragalactic antiprotons";
  fx0 = 1.9;
  r0 = 9 * 10^-4;
  constraint = Function[m,
    3.81 * 10^8 * 3 * 10^-24 * (r0 / 10^-5) (fx[m] / fx0) * (m / mx) ^ (1 / 2) * (m / ms) ^ (-1 / 2)];
  tmin = Log[10, th[4 * 10^13]]; tmax = Log[10, th[2 * 10^14]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* LSP - Emission of the Lightest
   Supersymmetric Particle LSP *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.49] *)
(* MASS RANGE: m < 10^11*(mLSP/100)^(-1) g *)
If[LSP == 1,
  cName = "Emission of the Lightest Supersymmetric Particle LSP";
  mLSP = 32.3; (* GeV *)(* ref: Abreu P.et al., 2000, PhLB, 489, 38 [Fig.3] *)
  constraint = Function[m,
    3.81 * 10^8 * 10^-18 * (m / 10^11) ^ (-1 / 2) * (mLSP / 100) ^ (-1) * (m / ms) ^ (-1 / 2)];
  tmin = min1; tmax = Log[10, th[10^11 * (mLSP / 100) ^ (-1)]];
```

```
  tabFC = FB[tmin, tmax, constraint];
   Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* DUP - decay of unstable
  particles: e.g. gravitino, neutralino *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.50] *)
(* MASS RANGE:   m < 10^9 g *)
If[DUP == 1,
  cName = "Decay of unstable particles:e.g.gravitino,neutralino";
  YPBH = 10^-14;
  xphi = 0.006;
  constraint = Function[m, 3.81 * 10^8 * 5 * 10^-19 *
     (m / 10^9) ^ (-1 / 2) * (YPBH / 10^-14) * (xphi / 0.006) ^ (-1) * (m / ms) ^ (-1 / 2)];
  tmin = min1; tmax = Log[10, th[10^9]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* GRBFL - Femtolensing of GRBs *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.63] *)
(* MASS RANGE: 5*10^16 g < m < 10^19 g *)
If[GRBFL == 1,
  cName = "Femtolensing of GRBs";
  constraint = Function[m, 0.1];
  tmin = Log[10, th[5 * 10^16]]; tmax = Log[10, th[10^19]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* EROSMLS1 - EROS microlensing of stars 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.65] *)
(* MASS RANGE: 6*10^-8 ms < m < 15 ms *)
If[EROSMLS1 == 1,
  cName = "EROS microlensing of stars 1";
  constraint = Function[m, 1];
  tmin = Log[10, th[ms * 6 * 10^-8]]; tmax = Log[10, th[ms * 15]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* EROSMLS2 - EROS microlensing of stars 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.65] *)
(* MASS RANGE: 10^-6 ms < m < 1 ms *)
If[EROSMLS2 == 1,
  cName = "EROS microlensing of stars 2";
  constraint = Function[m, 0.1];
  tmin = Log[10, th[ms * 10^-6]]; tmax = Log[10, th[ms]];
  tabFC = FB[tmin, tmax, constraint];
```

```
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* EROSMLS3 - EROS microlensing of stars 3 *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.65] *)
(* MASS RANGE: 10^-3 ms < m < 0.1 ms *)
If[EROSMLS3 == 1,
  cName = "EROS microlensing of stars 3";
  constraint = Function[m, 0.04];
  tmin = Log[10, th[ms * 10^-3]]; tmax = Log[10, th[0.1 * ms]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* OGLEMLS1 - OGLE microlensing of stars 1 *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.66] *)
(* MASS RANGE: 10^-7 ms < m < 1 ms *)
If[OGLEMLS1 == 1,
  cName = "OGLE microlensing of stars 1";
  constraint = Function[m, 1];
  tmin = Log[10, th[ms * 10^-7]]; tmax = Log[10, th[ms]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* OGLEMLS2 - OGLE microlensing of stars 2 *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.66] *)
(* MASS RANGE: 10^-6 ms < m < 10^-2 ms *)
If[OGLEMLS2 == 1,
  cName = "OGLE microlensing of stars 2";
  constraint = Function[m, 0.1];
  tmin = Log[10, th[ms * 10^-6]]; tmax = Log[10, th[ms * 10^-2]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* OGLEMLS3 - OGLE microlensing of stars 3 *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.66] *)
(* MASS RANGE: m ~ 10^-4 ms *)
If[OGLEMLS3 == 1,
  cName = "OGLE microlensing of stars 3";
  constraint = Function[m, 0.01];
  tmin = Log[10, th[ms * 5 * 10^-5]]; tmax = Log[10, th[ms * 5 * 10^-4]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* KeplerMLS - Kepler microlensing of stars *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.67] *)
(* MASS RANGE: 2*10^-9 ms < m < 10^-7 ms *)
If[KeplerMLS == 1,
  cName = "Kepler microlensing of stars";
  constraint = Function[m, 0.3];
  tmin = Log[10, th[ms * 2.0 * 10^-9 ]] ; tmax = 1.0 * Log[10, th[ms * 10^-7]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* QMLS - Quasar microlensing *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.68] *)
(* MASS RANGE: 10^-3 ms < m < 60 ms *)
If[QMLS == 1,
  cName = "Quasar microlensing";
  constraint = Function[m, 1];
  tmin = Log[10, th[ms * 10^-3 ]] ; tmax = 1.0 * Log[10, th[ms * 60]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* CRSMLS1 - Millilensing of compact radio sources 1 *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.69] *)
(* MASS RANGE: m < 10^5 ms *)
If[CRSMLS1 == 1,
  cName = "Millilensing of compact radio sources 1";
  constraint = Function[m, (m / (2 * 10^4 * ms)) ^ (-2)];
  tmin = min1 ; tmax = Log[10, th[ms * 10^5]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* CRSMLS2 - Millilensing of compact radio sources 2 *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.69] *)
(* MASS RANGE: 10^5 ms < m < 10^8 ms *)
If[CRSMLS2 == 1,
  cName = "Millilensing of compact radio sources 2";
  constraint = Function[m, 0.06];
  tmin = Log[10, th[ms * 10^5]] ; tmax = Log[10, th[ms * 10^8]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

(* CRSMLS3 - Millilensing of compact radio sources 3 *)

```
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.69] *)
(* MASS RANGE: m> 10^8 ms *)
```

```
If[CRSMLS3 == 1,
  cName = "Millilensing of compact radio sources 3";
  constraint = Function[m, (m / (4 * 10^8 * ms))^2];
  tmin = Log[10, th[ms * 10^8]] ; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* WDNS - Capture of PBHs by
   white dwarfs and neutron stars *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.74] *)
(* MASS RANGE: 2.5*10^18 g < m < 10^25 g *)
If[WDNS == 1,
  cName = "Capture of PBHs by white dwarfs and neutron stars";
  constraint = Function[m, m / (4.7 * 10^24) * (1 - E^(-m / (2.9 * 10^23)))^(-1)];
  tmin = Log[10, th[2.5 * 10^18]]; tmax = Log[10, th[10^25]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* WBS1 - Wide-Binary stability 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.75] *)
(* MASS RANGE: 500 ms < m < 10^3 ms *)
If[WBS1 == 1,
  cName = "Wide-binary stability 1";
  constraint = Function[m, 500 * ms / m];
  tmin = Log[10, th[ms * 500]] ; tmax = Log[10, th[ms * 10^3]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* WBS2 - Wide-Binary stability 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.75] *)
(* MASS RANGE: 10^3 ms < m < 10^8 ms *)
If[WBS2 == 1,
  cName = "Wide-binary stability 2";
  constraint = Function[m, 0.5];
  tmin = Log[10, th[ms * 10^3]] ; tmax = Log[10, th[ms * 10^8]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* DGC1 - Disruption of globular clusters 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.76] *)
(* MASS RANGE: 3*10^4 ms < m < 10^6 ms *)
If[DGC1 == 1,
  cName = "Disruption of globular clusters 1";
  constraint = Function[m, 3 * 10^4 * ms / m];
  tmin = Log[10, th[ms * 3 * 10^4]] ; tmax = Log[10, th[ms * 10^6]];
```

```
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DGC2 - Disruption of globular clusters 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.76] *)
(* MASS RANGE: 10^6 ms < m < 10^11 ms *)
If[DGC2 == 1,
  cName = "Disruption of globular clusters 2";
  constraint = Function[m, 0.03];
  tmin = Log[10, th[ms * 10^6]] ; tmax = Log[10, th[ms * 10^11]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* ERI1 - Eridanus II-
  survival of star clusters 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.77] *)
(* MASS RANGE: m < 10^4 ms *)
If[ERI1 == 1,
  cName = "Eridanus II-survival of star clusters 1";
  constraint = Function[m, (m / (3.7 * ms)) ^ (-1) / (1.1 - 0.1 * Log[E, m / ms])];
  tmin = min1; tmax = Log[10, th[ms * 10^4]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* ERI2 - Eridanus II-
  survival of star clusters 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.77] *)
(* MASS RANGE: m > 10^4 ms *)
If[ERI2 == 1,
  cName = "Eridanus II-survival of star clusters 2";
  constraint = Function[m, m / (10^6 * ms)];
  tmin = Log[10, th[ms * 10^4]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DHTS1 - Disk heating and tidal streams 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.78] *)
(* MASS RANGE: m < 3*10^9 ms *)
If[DHTS1 == 1,
  cName = "Disk heating and tidal streams 1";
  constraint = Function[m, 3 * 10^6 * ms / m];
  tmin = min1; tmax = Log[10, th[ms * 3 * 10^9]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
```

```
];

(* DHTS2 - Disk heating and tidal streams 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.78] *)
(* MASS RANGE: m > 3*10^9 ms *)
If[DHTS2 == 1,
  cName = "Disk heating and tidal streams 2";
  constraint = Function[m, m / mhalo];
  tmin = Log[10, th[ms * 3 * 10 ^ 9]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DFHO1 -
  Dynamical friction on halo objects 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79] *)
(* MASS RANGE: m < 5*10^5 ms *)
If[DFHO1 == 1,
  cName = "Dynamical friction on halo objects 1";
  constraint = Function[m, (m / (2 * 10 ^ 4 * ms)) ^ (-10 / 7)];
  tmin = min1; tmax = Log[10, th[ms * 5 * 10 ^ 5]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DFHO2 -
  Dynamical friction on halo objects 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79] *)
(* MASS RANGE: 5*10^5 ms < m < 2*10^6 ms *)
If[DFHO2 == 1,
  cName = "Dynamical friction on halo objects 2";
  constraint = Function[m, (m / (4 * 10 ^ 4 * ms)) ^ (-2)];
  tmin = Log[10, th[ms * 5 * 10 ^ 5]]; tmax = Log[10, th[ms * 2 * 10 ^ 6]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DFHO3 -
  Dynamical friction on halo objects 3 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79] *)
(* MASS RANGE: 2*10^6 ms < m < 10^8 ms *)
If[DFHO3 == 1,
  cName = "Dynamical friction on halo objects 3";
  constraint = Function[m, (m / (0.1 * ms)) ^ (-1 / 2)];
  tmin = Log[10, th[ms * 2 * 10 ^ 6]]; tmax = Log[10, th[ms * 10 ^ 8]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* DFHO4 -
 Dynamical friction on halo objects 4 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79] *)
(* MASS RANGE: m > 10^8 ms *)
If[DFHO4 == 1,
  cName = "Dynamical friction on halo objects 4";
  constraint = Function[m, m / mhalo];
  tmin = Log[10, th[ms * 10^8]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DDGC1 - Disruption/distortion
   of galaxies in clusters 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.80] *)
(* MASS RANGE: 10^10 ms < m < 10^11 ms *)
If[DDGC1 == 1,
  cName = "Disruption/distortion of galaxies in clusters 1";
  constraint = Function[m, 10^10 * ms / m];
  tmin = Log[10, th[ms * 10^10]]; tmax = Log[10, th[ms * 10^11]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DDGC2 - Disruption/distortion
   of galaxies in clusters 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.80] *)
(* MASS RANGE: 10^11 ms < m < 10^13 ms *)
If[DDGC2 == 1,
  cName = "Disruption/distortion of galaxies in clusters 2";
  constraint = Function[m, 0.1];
  tmin = Log[10, th[ms * 10^11]]; tmax = Log[10, th[ms * 10^13]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DDGC3 - Disruption/distortion
   of galaxies in clusters 3 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.80] *)
(* MASS RANGE: m > 10^13 ms *)
If[DDGC3 == 1,
  cName = "Disruption/distortion of galaxies in clusters 3";
  constraint = Function[m, m / (2 * 10^14 * ms)];
  tmin = Log[10, th[ms * 10^13]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];
```

```
(* MW1 - MW type galaxies 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.88] *)
(* MASS RANGE: 10^6 ms < m < 10^9 ms *)
If[MWG1 == 1,
  cName = "MW type galaxies 1";
  constraint = Function[m, 10^6 * ms / m];
  tmin = Log[10, th[ms * 10^6]]; tmax = Log[10, th[ms * 10^9]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* MW2 - MW type galaxies 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.88] *)
(* MASS RANGE: 10^9 ms < m < 10^12 ms *)
If[MWG2 == 1,
  cName = "MW type galaxies 2";
  constraint = Function[m, m / (10^12 * ms)];
  tmin = Log[10, th[ms * 10^9]]; tmax = Log[10, th[ms * 10^12]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DG1 - Dwarf type galaxies 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.89] *)
(* MASS RANGE: 10^5 ms < m < 3*10^7 ms *)
If[DG1 == 1,
  cName = "Dwarf type galaxies 1";
  constraint = Function[m, 10^5 * ms / m];
  tmin = Log[10, th[ms * 10^5]]; tmax = Log[10, th[ms * 3 * 10^7]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* DG2 - Dwarf type galaxies 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.89] *)
(* MASS RANGE: 3*10^7 ms < m < 10^10 ms *)
If[DG2 == 1,
  cName = "Dwarf type galaxies 2";
  constraint = Function[m, m / (10^10 * ms)];
  tmin = Log[10, th[ms * 3 * 10^7]]; tmax = Log[10, th[ms * 10^10]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* CG1 - Clusters of galaxies 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.90] *)
(* MASS RANGE: 10^7 ms < m < 3*10^10 ms *)
If[CG1 == 1,
  cName = "Clusters of galaxies 1";
```

```
  constraint = Function[m, 10^7 * ms / m];
  tmin = Log[10, th[ms * 10^7]]; tmax = Log[10, th[ms * 3 * 10^10]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* CG2 - Clusters of galaxies 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.90] *)
(* MASS RANGE: 3*10^10 ms < m < 10^14 ms *)
If[CG2 == 1,
  cName = "Clusters of galaxies 2";
  constraint = Function[m, m / (10^14 * ms)];
  tmin = Log[10, th[ms * 3 * 10^10]]; tmax = Log[10, th[ms * 10^14]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* FBC1 - First baryonic clouds 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.91] *)
(* MASS RANGE: 10^3 ms < m < 3*10^4 ms *)
If[FBC1 == 1,
  cName = "First baryonic clouds 1";
  constraint = Function[m, 10^3 * ms / m];
  tmin = Log[10, th[ms * 10^3]]; tmax = Log[10, th[ms * 3 * 10^4]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* FBC2 - First baryonic clouds 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.91] *)
(* MASS RANGE: 3*10^4 ms < m < 10^6 ms *)
If[FBC2 == 1,
  cName = "First baryonic clouds 2";
  constraint = Function[m, m / (10^6 * ms)];
  tmin = Log[10, th[ms * 3 * 10^4]]; tmax = Log[10, th[ms * 10^6]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* WMAPCMB1 - WMAP-
  CMB anisotropies/distortions 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.92] *)
(* MASS RANGE: 30 ms < m < 10^4 ms *)
If[WMAPCMB1 == 1,
  cName = "WMAP-CMB anisotropies/distortions 1";
  constraint = Function[m, (m / (30 * ms)) ^ (-2)];
  tmin = Log[10, th[ms * 30]]; tmax = Log[10, th[ms * 10^4]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
```

```
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* WMAPCMB2 - WMAP-
  CMB anisotropies/distortions 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.92] *)
(* MASS RANGE: 10^4 ms < m < 10^11 ms *)
If[WMAPCMB2 == 1,
  cName = "WMAP-CMB anisotropies/distortions 2";
  constraint = Function[m, 10^-5];
  tmin = Log[10, th[ms * 10^4]]; tmax = Log[10, th[ms * 10^11]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* WMAPCMB3 - WMAP-
  CMB anisotropies/distortions 3 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.92] *)
(* MASS RANGE:  m > 10^11 ms *)
If[WMAPCMB3 == 1,
  cName = "WMAP-CMB anisotropies/distortions 3";
  constraint = Function[m, m / (10^16 * ms)];
  tmin = Log[10, th[ms * 10^11]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* FIRASCMB1 - FIRAS-
  CMB anisotropies/distortions 1 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.93] *)
(* MASS RANGE: 1 ms < m < 10^3 ms *)
If[FIRASCMB1 == 1,
  cName = "FIRAS-CMB anisotropies/distortions 1";
  constraint = Function[m, (m / ms) ^ (-2)];
  tmin = Log[10, th[ms * 1]]; tmax = Log[10, th[ms * 10^3]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* FIRASCMB2 - FIRAS-
  CMB anisotropies/distortions 2 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.93] *)
(* MASS RANGE: 10^3 ms < m < 10^14 ms *)
If[FIRASCMB2 == 1,
  cName = "FIRAS-CMB anisotropies/distortions 2";
  constraint = Function[m, 0.015];
  tmin = Log[10, th[ms * 10^3]]; tmax = Log[10, th[ms * 10^14]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
```

```
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* FIRASCMB3 - FIRAS-
  CMB anisotropies/distortions 3 *)
(* ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq .93] *)
(* MASS RANGE:  m > 10^14 ms *)
If[FIRASCMB3 == 1,
  cName = "FIRAS-CMB anisotropies/distortions 3";
  constraint = Function[m, m / (10^16 * ms)];
  tmin = Log[10, th[ms * 10^14]]; tmax = max1;
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* CMBDL - CMB dipole limit *)
(* ref: Carr B., Kuhnel F., Visinelli L., 2021, MNRAS, 501, 2029 [Fig .1] *)
(* MASS RANGE: 10^18.5 ms < m < 10^21.5 ms *)
If[CMBDL == 1,
  cName = "CMB dipole limit";
  constraint = Function[m, 10^(-1.5 * Log[10, m] + 78.7)];
  tmin = Log[10, th[ms * 10^18.5]] ; tmax = Log[10, th[ms * 10^21.5]];
  tabFC = FB[tmin, tmax, constraint];
  Global = Append[Global, tabFC];
  ActiveConstraints = Append[ActiveConstraints, cName]
 ];

(* 7 - Frames for the different figures *)

(* frame for f (t) *)
fig0t = Plot[1, {i, min1, max1},
   PlotRange → {{min1, max1}, {-12, 2}}, DisplayFunction → Identity,
   Frame → True, Axes → None, FrameLabel → {"Log10 (t/1s)", "Log10 (f(t))"}];

(* frame for f (M) *)
Mmin1 = Log[10, mh[10^min1]]; Mmax1 = Log[10, mh[10^max1]];
fig0M = Plot[1, {i, Mmin1, Mmax1},
   PlotRange → {{Mmin1, Mmax1}, {-12, 2}}, DisplayFunction → Identity,
   Frame → True, Axes → None, FrameLabel → {"Log10 (M/1g)", "Log10 (f(M))"}];

(* frame for betaprime (M) *)
Mmin1 = Log[10, mh[10^min1]]; Mmax1 = Log[10, mh[10^max1]];
fig0betaLinhaM = Plot[1, {i, Mmin1, Mmax1},
   PlotRange → {{Mmin1, Mmax1}, {-30, 1}}, DisplayFunction → Identity,
   Frame → True, Axes → None, FrameLabel → {"Log10 (M/1g)", "Log10 (β' (M))"}];

(* frame for betaprime (t) *)
fig0betaLinhat = Plot[1, {i, min1, max1},
   PlotRange → {{min1, max1}, {-30, 1}}, DisplayFunction → Identity,
   Frame → True, Axes → None, FrameLabel → {"Log10 (t/1s)", "Log10 (β' (t))"}];

(* frame for beta (M) *)
Mmin1 = Log[10, mh[10^min1]]; Mmax1 = Log[10, mh[10^max1]];
```

```
fig0betaM = Plot[1, {i, Mmin1, Mmax1},
   PlotRange → {{Mmin1, Mmax1}, {-30, 1}}, DisplayFunction → Identity,
   Frame → True, Axes → None, FrameLabel → {"Log10 (M/1g)", "Log10 (β(M))"}];

(* frame for beta (t) *)
fig0betat = Plot[1, {i, min1, max1},
   PlotRange → {{min1, max1}, {-30, 1}}, DisplayFunction → Identity,
   Frame → True, Axes → None, FrameLabel → {"Log10 (t/1s)", "Log10 (β(t))"}];
```

(* 8 - Generating the global figures *)

```
figft = {};
figfM = {};
figblt = {};
figblM = {};
figbt = {};
figbM = {};

comp = Length[logTime];

i = 1; (* pointing to constraint #1 *)
While[i ≤ Length[Global],

 w = Table[{Global[[i, j, 1]], Global[[i, j, 3]]}, {j, 1, comp}];
 fig = ListPlot[w, PlotJoined → True, DisplayFunction → Identity];
 figft = Append[figft, fig];

 w = Table[{Global[[i, j, 2]], Global[[i, j, 3]]}, {j, 1, comp}];
 fig = ListPlot[w, PlotJoined → True, DisplayFunction → Identity];
 figfM = Append[figfM, fig];

 w = Table[{Global[[i, j, 1]], Global[[i, j, 4]]}, {j, 1, comp}];
 fig = ListPlot[w, PlotJoined → True, DisplayFunction → Identity];
 figblt = Append[figblt, fig];

 w = Table[{Global[[i, j, 2]], Global[[i, j, 4]]}, {j, 1, comp}];
 fig = ListPlot[w, PlotJoined → True, DisplayFunction → Identity];
 figblM = Append[figblM, fig];

 w = Table[{Global[[i, j, 1]], Global[[i, j, 5]]}, {j, 1, comp}];
 fig = ListPlot[w, PlotJoined → True, DisplayFunction → Identity];
 figbt = Append[figbt, fig];

 w = Table[{Global[[i, j, 2]], Global[[i, j, 5]]}, {j, 1, comp}];
 fig = ListPlot[w, PlotJoined → True, DisplayFunction → Identity];
 figbM = Append[figbM, fig];

 i = i + 1
];
```

(* 9 - Display global figures *)

```
Print["f as a function of time:"];
figFT2025global = Show[fig0t, figft, DisplayFunction → $DisplayFunction]

Print["f as a function of mass:"];
figFM2025global = Show[fig0M, figfM, DisplayFunction → $DisplayFunction]

Print["beta prime as a function of time:"];
figBLT2025global = Show[fig0betaLinhat, figblt, DisplayFunction → $DisplayFunction]

Print["beta prime as a function of mass:"];
figBLM2025global = Show[fig0betaLinhaM, figblM, DisplayFunction → $DisplayFunction]

Print["beta as a function of time:"];
figBT2025global = Show[fig0betat, figbt, DisplayFunction → $DisplayFunction]

Print["beta as a function of mass:"];
figBM2025global = Show[fig0betaM, figbM, DisplayFunction → $DisplayFunction]
```

(* 10 - Generate figures with the minimum values only *)

```
i = 1; (* pointing to the first instant (or mass) *)

minft = {};
minfM = {};
minblt = {};
minblM = {};
minbt = {};
minbM = {};
minCName = {};

While[i ≤ comp,

  minft = Append[minft,
    {logTime[[i]], Min[Table[Global[[j, i, 3]], {j, 1, Length[Global]}]]}];
  minfM = Append[minfM, {logMass[[i]],
     Min[Table[Global[[j, i, 3]], {j, 1, Length[Global]}]]}];
  minblt = Append[minblt, {logTime[[i]],
     Min[Table[Global[[j, i, 4]], {j, 1, Length[Global]}]]}];
  minblM = Append[minblM, {logMass[[i]],
     Min[Table[Global[[j, i, 4]], {j, 1, Length[Global]}]]}];
  minbt = Append[minbt, {logTime[[i]],
     Min[Table[Global[[j, i, 5]], {j, 1, Length[Global]}]]}];
  minbM = Append[minbM, {logMass[[i]],
     Min[Table[Global[[j, i, 5]], {j, 1, Length[Global]}]]}];

   minCName = Append[minCName,

    {logTime[[i]],

     wx1 = Table[Global[[j, i, 5]], {j, 1, Length[Global]}];
     wx1Min = Min[wx1];
```

```
      j = 1; resMIN = "-";
      While[j <= Length[wx1],
       If[Global[[j, i, 5]] == wx1Min, resMIN = Global[[j, i, 6]]];
       j = j + 1];
      resMIN
     }
    ];

  i = i + 1
 ];

figft = ListPlot[minft, PlotJoined → True, DisplayFunction → Identity];
figfM = ListPlot[minfM, PlotJoined → True, DisplayFunction → Identity];
figblt = ListPlot[minblt, PlotJoined → True, DisplayFunction → Identity];
figblM = ListPlot[minblM, PlotJoined → True, DisplayFunction → Identity];
figbt = ListPlot[minbt, PlotJoined → True, DisplayFunction → Identity];
figbM = ListPlot[minbM, PlotJoined → True, DisplayFunction → Identity];

Print["f as a function of time:"];
figFT2025 = Show[fig0t, figft, DisplayFunction → $DisplayFunction]

Print["f as a function of mass:"];
figFM2025 = Show[fig0M, figfM, DisplayFunction → $DisplayFunction]

Print["beta prime as a function of time:"];
figBLT2025 = Show[fig0betaLinhat, figblt, DisplayFunction → $DisplayFunction]

Print["beta prime as a function of mass:"];
figBLM2025 = Show[fig0betaLinhaM, figblM, DisplayFunction → $DisplayFunction]

Print["beta as a function of time:"];
figBT2025 = Show[fig0betat, figbt, DisplayFunction → $DisplayFunction]

Print["beta as a function of mass:"];
figBM2025 = Show[fig0betaM, figbM, DisplayFunction → $DisplayFunction]

(* selecting the ranges where a given constraint applies *)
j = 2; minCNameReduced = {minCName[[1]]};
While[j <= Length[minCName],
  If[minCName[[j, 2]] != minCName[[j - 1, 2]],
   minCNameReduced = Append[minCNameReduced, minCName[[j]]]];
  j = j + 1];
minCNameReduced;

Print["Active Constraints:"];
MatrixForm[ActiveConstraints]
Print["Relevant Constraints:"];
MatrixForm[minCNameReduced]
```

(* 11 - SAVE DATA FILES and FIGURES *)

```
(* Saving data *)
Save["math-ActiveConstraints.m", ActiveConstraints];
Save["math-RelevantConstraints.m", minCNameReduced];
```

```
Save["math-constraintsFT.m", minft];
Save["math-constraintsFM.m", minfM];
Save["math-constraintsBLT.m", minblt];
Save["math-constraintsBLM.m", minblM];
Save["math-constraintsBT.m", minbt];
Save["math-constraintsBM.m", minbM];

(* Saving figures *)
Display["math-constraintsFigFTglobal.eps", figFT2025global, "EPS"];
Display["math-constraintsFigFMglobal.eps", figFM2025global, "EPS"];
Display["math-constraintsFigBLTglobal.eps", figBLT2025global, "EPS"];
Display["math-constraintsFigBLMglobal.eps", figBLM2025global, "EPS"];
Display["math-constraintsFigBTglobal.eps", figBT2025global, "EPS"];
Display["math-constraintsFigBMglobal.eps", figBM2025global, "EPS"];

Display["math-constraintsFigFT.eps", figFT2025, "EPS"];
Display["math-constraintsFigFM.eps", figFM2025, "EPS"];
Display["math-constraintsFigBLT.eps", figBLT2025, "EPS"];
Display["math-constraintsFigBLM.eps", figBLM2025, "EPS"];
Display["math-constraintsFigBT.eps", figBT2025, "EPS"];
Display["math-constraintsFigBM.eps", figBM2025, "EPS"]
```

B The implementation code in Python

Here we present the code implemented in the Python language (Python Software Foundation, 2023), using the program Thonny (Thonny IDE, 2023). The code is divided into 11 sections as follows:

1. **Initial technical settings**: set current directory.
2. **Enable/Disable Constraints**: list of boolean type variables indicating whether each constraint is enabled (1) or disabled (0) - see Section 3.
3. **Main parameters and functions**: parameters, physical constants and functions that are common to several constraints.
4. **Degrees of freedom** $g(T)$: functions related to the existence of different particle species (see Section 2).
5. **Functions beta, betaprime and f**: the sector where the functions β, β' and f are implemented, together with the function **FB** which controls all the calculations and data store for further use.
6. **List of Constraints**: this section is subdivided in several blocks, one for each constraint. The standard structure of these blocks is detailed in Section 4.3 (where we explained how to add a new constraint).
7. **Frames for the different figures**: six frames for the six types of graphs that will be returned as outputs [$\beta(t)$, $\beta'(t)$, $f(t)$, $\beta(M)$, $\beta'(M)$, $f(M)$].
8. **Generating the global figures**: the sector where the data for the global graphs (see Figures 2, 3, 6 and 7) is selected.
9. **Generate figures with the minimum values only**: the sector where the data for the combined graphs (see Figures 4, 5, 8 and 9) is determined.
10. **Display global figures**: display the global graphs on screen.
11. **SAVE FILES and FIGURES**: saving to the current directory the 20 outputs mentioned in Section 4.2.

```
#OBSERVATIONAL CONSTRAINTS ON PRIMORDIAL BLACK HOLES

#Faculdade de Ci ncias Exatas e da Engenharia da Universidade da Madeira *
#Grupo de Astronomia da Universidade da Madeira - GAUMa *
#Sobrinho J.L.G. & Jesus P.D.F., June 2025 *
#Version for: Python 3.10.11

#Table of Contents
# 01 - Initial technical settings
# 02 - Enable/Disable Constraints
# 03 - Main parameters and functions
# 04 - Degrees of freedom g (T)
# 05 - Functions beta, betaprime and f
# 06 - List of Constraints
# 07 - Frames for the different figures
# 08 - Generating the global figures
# 09 - Generate figures with the minimum values only
# 10 - Display all figures
# 11 - SAVE FILES and FIGURES


# 01 - Initial technical settings
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt, log10, exp
import os

# 02 - Enable/Disable Constraints
constraints = {
"BBNDH": 1,    # Big Bang Nucleosynthesis D/H
"BBNY": 1,     # Big Bang Nucleosynthesis Y
"GE": 1,       # Generation of entropy
"CMBSD": 1,    # CMB spectral distortions
"CMBA": 1,     # CMB small scale anisotropies
"EGB1": 1,     # Extragalactic gamma-ray background 1 (evaporated)
"EGB2": 1,     # Extragalactic gamma-ray background 2 (evaporated)
"EGB3": 1,     # Extragalactic gamma-ray background 3 (exploding)
"EGB4": 1,     # Extragalactic gamma-ray background 4 (
```

```
    evaporating)
"EGAP": 1,      # Extragalactic antiprotons
"LSP": 1,       # Emission of the Lightest Supersymmetric Particle LSP
"DUP": 1,       # Decay of unstable particles : e.g. gravitino, neutralino
"GRBFL": 1,     # Femtolensing of GRBs
"EROSMLS1": 1, #EROS microlensing of stars 1
"EROSMLS2": 1, #EROS microlensing of stars 2
"EROSMLS3": 1, #EROS microlensing of stars 3
"OGLEMLS1": 1, #OGLE microlensing of stars 1
"OGLEMLS2": 1, #OGLE microlensing of stars 2
"OGLEMLS3": 1, #OGLE microlensing of stars 3
"KeplerMLS": 1,#Kepler microlensing of stars
"QMLS": 1,      #Quasar microlensing
"CRSMLS1": 1,   # Millilensing of compact radio sources 1
"CRSMLS2": 1,   # Millilensing of compact radio sources 2
"CRSMLS3": 1,   # Millilensing of compact radio sources 3
"WDNS": 1,      # Capture of PBHs by white dwarfs and neutron stars
"WBS1": 1,      # Wide-binary stability 1
"WBS2": 1,      # Wide-binary stability 2
"DGC1": 1,      # Disruption of globular clusters 1
"DGC2": 1,      # Disruption of globular clusters 2
"ERI1": 1,      # Eridanus II - survival of star clusters 1
"ERI2": 1,      # Eridanus II - survival of star clusters 2
"DHTS1": 1,     # Disk heating and tidal streams 1
"DHTS2": 1,     # Disk heating and tidal streams 2
"DFHO1": 1,     # Dynamical friction on halo objects 1
"DFHO2": 1,     # Dynamical friction on halo objects 2
"DFHO3": 1,     # Dynamical friction on halo objects 3
"DFHO4": 1,     # Dynamical friction on halo objects 4
"DDGC1": 1,     # Disruption/distortion of galaxies in clusters 1
"DDGC2": 1,     # Disruption/distortion of galaxies in clusters 2
"DDGC3": 1,     # Disruption/distortion of galaxies in clusters 3
"MW1": 1,      # MW type galaxies 1
"MW2": 1,      # MW type galaxies 2
"DG1": 1,       # Dwarf type galaxies 1
```

```
"DG2": 1,       # Dwarf type galaxies 2
"CG1": 1,       # Clusters of galaxies 1
"CG2": 1,       # Clusters of galaxies 2
"FBC1": 1,      # First baryonic clouds 1
"FBC2": 1,      # First baryonic clouds 2
"WMAPCMB1": 1, # WMAP - CMB anisotropies/distortions 1
"WMAPCMB2": 1, # WMAP - CMB anisotropies/distortions 2
"WMAPCMB3": 1, # WMAP - CMB anisotropies/distortions 3
"FIRASCMB1": 1,# FIRAS - CMB anisotropies/distortions 1
"FIRASCMB2": 1,# FIRAS - CMB anisotropies/distortions 2
"FIRASCMB3": 1,# FIRAS - CMB anisotropies/distortions 3
"CMBDL": 1,     # CMB dipole limit
}

constraintsnames = [
    # Big Bang Nucleosynthesis
    "BBNDH", "BBNY",

    # Generation of Entropy
    "GE",

    # CMB spectral distortions and anisotropies
    "CMBSD", "CMBA",

    # Extragalactic gamma-ray background
    "EGB1", "EGB2", "EGB3", "EGB4",

    # Extragalactic antiprotons
    "EGAP",

    # Lightest Supersymmetric Particle emission
    "LSP",

    # Decay of unstable particles
    "DUP",

    # Femtolensing of GRBs
    "GRBFL",

    # Microlensing constraints
    "EROSMLS1", "EROSMLS2", "EROSMLS3",
    "OGLEMLS1", "OGLEMLS2", "OGLEMLS3",
    "KeplerMLS", "QMLS",
```

```
    # Millilensing
    "CRSMLS1", "CRSMLS2", "CRSMLS3",

    # Capture by stars
    "WDNS",

    # Wide-binary stability
    "WBS1", "WBS2",

    # Disruption of globular clusters
    "DGC1", "DGC2",

    # Eridanus II survival
    "ERI1", "ERI2",

    # Disk heating and tidal streams
    "DHTS1", "DHTS2",

    # Dynamical friction on halo objects
    "DFHO1", "DFHO2", "DFHO3", "DFHO4",

    # Disruption/distortion in galaxy clusters
    "DDGC1", "DDGC2", "DDGC3",

    # Milky Way-type galaxies
    "MW1", "MW2",

    # Dwarf galaxies
    "DG1", "DG2",

    # Clusters of galaxies
    "CG1", "CG2",

    # First baryonic clouds
    "FBC1", "FBC2",

    # WMAP constraints
    "WMAPCMB1", "WMAPCMB2", "WMAPCMB3",

    # FIRAS constraints
    "FIRASCMB1", "FIRASCMB2", "FIRASCMB3",

```

```python
    # CMB dipole limit
    "CMBDL",
]

#constraint default values

defaultValue=1;
defaultName="empty"

# 03 - Main parameters and functions
# Sun mass
ms=1.989e33 #(g)

# MW galaxy halo mass
#ref: Gardner S.,McDermott S.D.,Yanny B.,2021,PrPNP,121,103904.
mhalo=1e12*ms #(g)

#MW galaxy halo mass
#ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq. 16]
mx=5.1*1e14 #(g)

# Horizon mass mh [g] at time t [s]
# ref: e.g. Sobrinho J.L.G., Augusto P.,
# Gon alves A.L., 2016, MNRAS, 463, 2348 (eq. 02)
def mh(t):
    return 1e15*(t/1e-23)  #(g)

# Instant of time th (s) for which the horizon mass is m (g)
def th(m):
    return m/1e38 #(s)

# Converts time in seconds to temperature in GeV
def Ttime(ti):
    return 0.00121/sqrt(ti)

# logarithmic step of time for the X-axis
stept=0.005

# time interval considered on a logarithmic scale
min1=-30 # log (th ~ 1 s) minimum value
max1=17 # log (th ~ 1 s) maximum value
```

```python

# X-axis - log (time ~ 1 s)
logTime=[round(i,6) for i in np.arange(min1,max1+stept,stept)]

# X-axis - log (mass ~ 1 g)
logMass=[log10(mh(10**logTime[i])) for i in range(len(logTime))]

# 04 - Degrees of freedom g(T)
# T in GeV
# Parameters for the numerical approximation of g(T) when 120 MeV < T < 10^16 GeV
# ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 (Table 1)me=511e-6  # MeV -> GeV
a=[1, 1.11724, 3.12672e-1, -4.68049e-2, -2.65004e-2, -1.19760e-3, 1.82812e-4, 1.36436e-4, 8.55051e-5, 1.22840e-5, 3.82259e-7, -6.87035e-9]
b=[1.43382e-2, 1.37559e-2, 2.92108e-3, -5.38533e-4, -1.62496e-4, -2.87906e-5, -3.84278e-6, 2.78776e-6, 7.40342e-7, 1.17210e-7, 3.72499e-9, -6.74107e-11]
c=[1, 6.07869e-1, -1.54485e-1, -2.24034e-1, -2.82147e-2, 2.90620e-2, 6.86778e-3, -1.00005e-3, -1.69104e-4, 1.06301e-5, 1.69528e-6, -9.33311e-8]
d=[7.07388e1, 9.18011e1, 3.31892e1, -1.39779, -1.52558, -1.97857e-2, -1.60146e-1, 8.22615e-5, 2.02651e-2, -1.82134e-5, 7.83943e-5, 7.13518e-5]

# Degrees of freedom of the energy density
# High energy domain - 120 MeV < T < 10^16 GeV
# ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 (eq. C.1)
def grH(T):
    return sum(a[i]*(np.log(T))**i for i in range(12))/sum(b[i]*(np.log(T))**i for i in range(12))

# Degrees of freedom of the entropy density
# High energy domain - 120 MeV < T < 10^16 GeV
# ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 (eq. C.2)
def gsH(T):
    return grH(T)/(1+sum(c[i]*(np.log(T))**i for i in range(12))/sum(d[i]*(np.log(T))**i for i in range
```

```
(12)))

# Fitting functions for the Low energy domain - T < 120 MeV
# ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 (eq. C.5 to C.9)
def fp(x):
    return np.exp(-1.04855*x)*(1+1.03757*x+0.508630*x**2+0.0893988*x**3)
def bp(x):
    return np.exp(-1.03149*x)*(1+1.03317*x+0.398264*x**2+0.0648056*x**3)
def fs(x):
    return np.exp(-1.04190*x)*(1+1.03400*x+0.456426*x**2+0.0595248*x**3)
def bs(x):
    return np.exp(-1.03365*x)*(1+1.03397*x+0.342548*x**2+0.0506182*x**3)
def Sfit(x):
    return 1+7/4*np.exp(-1.0419*x)*(1+1.034*x+0.456426*x**2+0.0595249*x**3)

#constants
me=511*(10**-6)
mu=0.1056
mpi0=0.135
mpipm=0.140
m1=0.5
m2=0.77
m3=1.2
m4=2

# Degrees of freedom of the energy density
# Low energy domain - T < 120 MeV
# ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 (eq. C.3)
def grL(T):
    return 2.030+1.353*(Sfit(me/T))**(4/3)+3.495*fp(me/T)+3.446*fp(mu/T)+1.05*bp(mpi0/T)+2.08*bp(mpipm/T)+4.165*bp(m1/T)+30.55*bp(m2/T)+80.4*bp(m3/T)+8209*bp(m4/T)

# Degrees of freedom of the entropy density
```

```python
# Low energy domain - T < 120 MeV
# ref: Saikawa K., Shirai S., 2018, JCAP, 2018, 035 (eq. C.4)
def gsL(T):
    return 2.008+1.923*Sfit(me/T)+3.442*fs(me/T)+3.468*fs(mu/T)+1.034*bs(mpi0/T)+2.068*bs(mpipm/T)+4.16*bs(m1/T)+30.55*bs(m2/T)+90*bs(m3/T)+6209*bs(m4/T)

# Global degrees of freedom High/Low energy domain
def gr(T):
    return grL(T) if T<0.12 else grH(T)
def gs(T):
    return gsL(T) if T<0.12 else gsH(T)

# Contributions of the different species of particles to the PBH evaporation process
# ref: Gibilisco M., 1997, IJMPA, 12, 4167 (eq. 2.3)
def fx(m):
    return (1.569+0.569*np.exp(-m/(4.53e14))+6*np.exp(-m/(1.60e14))+3*np.exp(-m/(9.6e13))+3*np.exp(-m/(2.56e13))+np.exp(-m/(2.68e13))+3*np.exp(-m/(9.07e12))+3*np.exp(-m/(0.48e12))+0.963*np.exp(-m/(1.10e14)))

# 5 - Functions beta, betaprime and f
# fraction f(M) of the halo in PBHs as a function of betaprime(M)
# ref: Carr B. et al., 2021, RPPh, 84, 116902 (eq. 57)
def f(mi,betaprimei):
    return 3.81e8*betaprimei*1/sqrt(mi/ms)

# betaprime(M) as a function of f(M)
# ref: Carr B. et al., 2021, RPPh, 84, 116902 (eq. 57)
def betaprime(mi,fi):
    return fi/(3.81e8)*sqrt(mi/ms)

# betaprime(M) as a function of f(M)
# ref: Carr B. et al., 2021, RPPh, 84, 116902 (eq. 57)
gamma=1

# Hubble parameter present day value: H0 = 100 h km/s/Mpc
# ref: Particle Data Group - Cosmological Parameters
```

```
# Revised September 2021 by O.Lahav and A.R.Liddle
h=0.674

# beta(t) as a function of betaprime(t)
# ref: Carr B. et al., 2021, RPPh, 84, 116902 (eq. 8)
def beta(betaprimei,ti):
    return (1/sqrt(gamma))*(gs(Ttime(ti))/106.75)**(1/4)*(h/0.67)**2*betaprimei

# Compute: f, betaprime and beta
# INPUTs:
# min0 = log(tmin / 1 s) for this constraint
# max0 = log(tmax / 1 s) for this constraint
# f0 = function which defines the constraint

# OUTPUT:
# list with lists of the form:
# [log(t / 1 s), log(m / 1 g), log(f), log(betaprime), log(beta), constraint name]
# For points outside the constraints application domain it is returned:
# [log(t / 1 s), log(m / 1 s), defaultValue, defaultValue, defaultValue, defaultName]
def FB(min0,max0,f0):
    res=[]
    for i in range(len(logTime)):
        if logTime[i]<min0 or logTime[i]>max0:
            res.append([logTime[i],logMass[i],defaultValue,defaultValue,defaultValue])
        else:
            taux=10**logTime[i]
            res.append([logTime[i],logMass[i],log10(f0(mh(taux))),log10(betaprime(mh(taux),f0(mh(taux)))),log10(beta(betaprime(mh(taux),f0(mh(taux))),taux)),cName])
    return res

# 6 - List of Constraints
Global=[] # list to save the results of all the constraints
ActiveConstraints=[]
#list to store the names of all considered constraints

```

```
# BBNDH: Big Bang Nucleosynthesis D/H
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [Figs. 4
    and 5]
# MASS RANGE D/H: 10^10 g < m < 10^13.3 g
# within this mass range we have betaprime = 10^-24 (
    approx.)
if constraints["BBNDH"]==1:
    cName="Big Bang Nucleosynthesis D/H"
    def BBNDH(m):
        return 3.81e8*1e-24*(m/ms)**-(1/2)
    tmin=log10(th(1e10))
    tmax=log10(th(10**13.3))
    Global.append(FB(tmin,tmax,BBNDH))
    ActiveConstraints.append(cName)

# BBNY: Big Bang Nucleosynthesis Y
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [Figs. 4
    and 5]
# MASS RANGE Y: 10^9 g < m < 10^10 g
# within this mass range we have betaprime = 2*10^-19 (
    approx.)
if constraints["BBNY"]==1:
    cName="Big Bang Nucleosynthesis Y"
    def BBNY(m):
        return 3.81e8*2e-19*(m/ms)**-(1/2)
    tmin=log10(th(1e9))
    tmax=log10(th(1e10))
    Global.append(FB(tmin,tmax,BBNY))
    ActiveConstraints.append(cName)

# GE: Generation of entropy
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.24]
# MASS RANGE: m < 10^9 g - doesn't have lower limit
if constraints["GE"]==1:
    cName="Generation of entropy"
    def GE(m):
        return 3.81e8*1e-5*(m/1e9)**-1*(m/ms)**-0.5
    tmin=min1
    tmax=log10(th(1e9))
    Global.append(FB(tmin,tmax,GE))
    ActiveConstraints.append(cName)

# CMBSD: CMB spectral distortions
```

```
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.25]
# MASS RANGE: 10^11 g < m < 10^13 g
if constraints["CMBSD"]==1:
    cName="CMB spectral distortions"
    def CMBSD(m):
        return 3.81e8*1e-16*(m/1e11)**-1*(m/ms)**-0.5
    tmin=log10(th(1e11))
    tmax=log10(th(1e13))
    Global.append(FB(tmin,tmax,CMBSD))
    ActiveConstraints.append(cName)

# CMBA: CMB small scale anisotropies
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [Eq.28]
# MASS RANGE: 2.5*10^13 g < m < 2.4*10^14 g
if constraints["CMBA"]==1:
    cName="CMB small scale anisotropies"
    fh=0.1
    def CMBA(m):
        return 3.81e8*3e-30*(fh/0.1)**-1*(m/1e13)**3.1*(
            m/ms)**-0.5
    tmin=log10(th(2e13))
    tmax=log10(th(2.4e14))
    Global.append(FB(tmin,tmax,CMBA))
    ActiveConstraints.append(cName)

# EGB1: Extragalactic gamma - ray background 1 (
  evaporated)
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [Eq.32]
# MASS RANGE: m < mx g
if constraints["EGB1"]==1:
    cName="Extragalactic gamma - ray background 1 (
      evaporated)"
    epsilon=0.4
    def EGB1(m):
        return 3.81e8*5e-28*(m/mx)**(-5/2-2*epsilon)*(m/
            ms)**-0.5
    tmin=min1
    tmax=log10(th(mx))
    Global.append(FB(tmin,tmax,EGB1))
    ActiveConstraints.append(cName)

# EGB2: Extragalactic gamma - ray background 2 (
  evaporated)
```

```
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [Eq.33]
# MASS RANGE: m > mx g
if constraints["EGB2"]==1:
    epsilon=0.4
    cName="Extragalactic gamma - ray background 2 (evaporated)"
    def EGB2(m):
        return 3.81e8*5e-26*(m/mx)**(7/2+epsilon)*(m/ms)**-0.5
    tmin=log10(th(mx))
    tmax=max1
    Global.append(FB(tmin,tmax,EGB2))
    ActiveConstraints.append(cName)

# EGB3: Extragalactic gamma - ray background 3 (exploding)
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [Fig.18 - betaprime ~ 10^-29]
# MASS RANGE: approx mx
if constraints["EGB3"]==1:
    cName="Extragalactic gamma - ray background 3 (exploding)"
    EGB3width=1.75/100
    def EGB3(m):
        return 3.81e8*1e-29*(m/ms)**(-1/2)
    tmin = log10(th(mx-mx*EGB3width))
    tmax = log10(th(mx+mx*EGB3width))
    Global.append(FB(tmin, tmax, EGB3))
    ActiveConstraints.append(cName)

# EGB4: Extragalactic gamma-ray background 4 (evaporating)
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.62]
# MASS RANGE: m > 5*10^14 g
if constraints["EGB4"]==1:
    cName="Extragalactic gamma-ray background 4 (evaporating)"
    epsilon=0.25
    def EGB4(m):
        return 2e-8*(m/(mx))**(3+epsilon)
    tmin=log10(th(mx))
    tmax=max1
    Global.append(FB(tmin,tmax,EGB4))
```

```
    ActiveConstraints.append(cName)

# EGAP: Extragalactic antiprotons
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.48]
# MASS RANGE: 4*10^13 g < m < 2*10^14 g
if constraints["EGAP"]==1:
    cName="Extragalactic antiprotons"
    fx0=1.9
    r0=9e-4
    def EGAP(m):
        return 3.81e8*3e-24*(r0/1e-5)*(fx(m)/fx0)*(m/mx)**0.5*(m/ms)**(-0.5)
    tmin=log10(th(4e13))
    tmax=log10(th(2e14))
    Global.append(FB(tmin,tmax,EGAP))
    ActiveConstraints.append(cName)

# LSP: Emission of the Lightest Supersymmetric Particle LSP
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.49]
# MASS RANGE: m < 10^11 * (mLSP / 100)^(-1) g
if constraints["LSP"]==1:
    cName="Emission of the Lightest Supersymmetric Particle LSP"
    mLSP=32.3 # GeV
    def LSP(m):
        return 3.81e8*1e-18*(m/1e11)**-0.5*(mLSP/100)**-1*(m/ms)**-0.5
    tmin=min1
    tmax=log10(th(1e11*(mLSP/100)**-1))
    Global.append(FB(tmin,tmax,LSP))
    ActiveConstraints.append(cName)

# DUP: Decay of unstable particles : e.g. gravitino, neutralino
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.50]
# MASS RANGE: m < 10^9 g
if constraints["DUP"]==1:
    cName="Decay of unstable particles : e.g. gravitino, neutralino"
    YPBH=1e-14
    xphi=0.006
    def DUP(m):
```

```
        return 3.81e8*5e-19*(m/1e9)**-0.5*(YPBH/1e-14)*(
            xphi/0.006)**-1*(m/ms)**-0.5
    tmin=min1
    tmax=log10(th(1e9))
    Global.append(FB(tmin,tmax,DUP))
    ActiveConstraints.append(cName)

# GRBFL: Femtolensing of GRBs
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.63]
# MASS RANGE: 5*10^16 g < m < 10^19 g
if constraints["GRBFL"]==1:
    cName="Femtolensing of GRBs"
    def GRBFL(m):
        return 0.1
    tmin = log10(th(5e16))
    tmax = log10(th(1e19))
    Global.append(FB(tmin, tmax, GRBFL))
    ActiveConstraints.append(cName)

# EROSMLS1: EROS microlensing of stars 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.65]
# MASS RANGE: 6 10^-8 ms < m < 15 ms
if constraints["EROSMLS1"]==1:
    cName="EROS microlensing of stars 1"
    def EROSMLS1(m):
        return 1
    tmin=log10(th(ms*6*1e-8))
    tmax=log10(th(15*ms))
    Global.append(FB(tmin,tmax,EROSMLS1))
    ActiveConstraints.append(cName)

# EROSMLS2: EROS microlensing of stars 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.65]
# MASS RANGE: 10^-6   ms < m < 1   ms
if constraints["EROSMLS2"]==1:
    cName="EROS microlensing of stars 2"
    def EROSMLS2(m):
        return 0.1
    tmin=log10(th(ms*1e-6))
    tmax=log10(th(ms))
    Global.append(FB(tmin, tmax, EROSMLS2))
    ActiveConstraints.append(cName)

```

```
# EROSMLS3: EROS microlensing of stars 3
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.65]
# MASS RANGE: 10^-3    ms < m < 0.1    ms
if constraints["EROSMLS3"]==1:
    cName="EROS microlensing of stars 3"
    def EROSMLS3(m):
        return 0.04
    tmin=log10(th(ms*1e-3))
    tmax=log10(th(ms*0.1))
    Global.append(FB(tmin, tmax, EROSMLS3))
    ActiveConstraints.append(cName)

# OGLEMLS1: OGLE microlensing of stars 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.66]
# MASS RANGE: 10^-7    ms < m < 1    ms
if constraints["OGLEMLS1"]==1:
    cName="OGLE microlensing of stars 1"
    def OGLEMLS1(m):
        return 1
    tmin=log10(th(ms*1e-7))
    tmax=log10(th(ms))
    Global.append(FB(tmin, tmax, OGLEMLS1))
    ActiveConstraints.append(cName)

# OGLEMLS2: OGLE microlensing of stars 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.66]
# MASS RANGE: 10^-6    ms < m < 10^-2    ms
if constraints["OGLEMLS2"]==1:
    cName="OGLE microlensing of stars 2"
    def OGLEMLS2(m):
        return 0.1
    tmin=log10(th(ms* 1e-6))
    tmax=log10(th(ms * 1e-2))
    Global.append(FB(tmin, tmax, OGLEMLS2))
    ActiveConstraints.append(cName)

# OGLEMLS3: OGLE microlensing of stars 3
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.66]
# MASS RANGE: m ~ 10^-4 * ms
if constraints["OGLEMLS3"]==1:
    cName="OGLE microlensing of stars 2"
    def OGLEMLS3(m):
        return 0.01
```

```
    tmin=log10(th(ms*5*1e-5))
    tmax=log10(th(ms*5*1e-4))
    Global.append(FB(tmin, tmax, OGLEMLS3))
    ActiveConstraints.append(cName)

# KeplerMLS: Kepler microlensing of stars
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.67]
# MASS RANGE: 2*10^-9 * ms < m < 10^-7 * ms
if constraints["KeplerMLS"]==1:
    cName="Kepler microlensing of stars"
    def KeplerMLS(m):
        return 0.3
    tmin=log10(th(ms*2e-9))
    tmax=log10(th(ms*1e-7))
    Global.append(FB(tmin,tmax,KeplerMLS))
    ActiveConstraints.append(cName)

# QMLS: Quasar microlensing
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.68]
# MASS RANGE: 10^-3 * ms < m < 60 * ms
if constraints["QMLS"]==1:
    cName="Quasar microlensing"
    def QMLS(m):
        return 1
    tmin=log10(th(ms*1e-3))
    tmax=log10(th(ms*60))
    Global.append(FB(tmin,tmax,QMLS))
    ActiveConstraints.append(cName)

# CRSMLS1: Millilensing of compact radio sources 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.69]
# MASS RANGE: m < 10^5 * ms
if constraints["CRSMLS1"]==1:
    cName="Millilensing of compact radio sources 1"
    def CRSMLS1(m):
        return (m/(2e4*ms))**(-2)
    tmin=min1
    tmax=log10(th(ms*1e5))
    Global.append(FB(tmin,tmax,CRSMLS1))
    ActiveConstraints.append(cName)

# CRSMLS2: Millilensing of compact radio sources 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.69]
```

```
# MASS RANGE: 10^5 * ms < m < 10^8 * ms
if constraints["CRSMLS2"]==1:
    cName="Millilensing of compact radio sources 2"
    def CRSMLS2(m):
        return 0.06
    tmin=log10(th(ms*1e5))
    tmax=log10(th(ms*1e8))
    Global.append(FB(tmin,tmax,CRSMLS2))
    ActiveConstraints.append(cName)

# CRSMLS3: Millilensing of compact radio sources 3
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.69]
# MASS RANGE: m > 10^8 * ms
if constraints["CRSMLS3"]==1:
    cName="Millilensing of compact radio sources 3"
    def CRSMLS3(m):
        return (m/(4e8*ms))**2
    tmin=log10(th(ms*1e8))
    tmax=max1
    Global.append(FB(tmin,tmax,CRSMLS3))
    ActiveConstraints.append(cName)

# WDNS: Capture of PBHs by white dwarfs and neutron stars
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.74]
# MASS RANGE: 2.5*10^18 g < m < 10^25 g
if constraints["WDNS"]==1:
    cName="Capture of PBHs by white dwarfs and neutron stars"
    def WDNS(m):
        return (m/(4.7e24))*(1-np.exp(-m/(2.9e23)))**(-1)
    tmin=log10(th(2.5e18))
    tmax=log10(th(1e25))
    Global.append(FB(tmin,tmax,WDNS))
    ActiveConstraints.append(cName)

# WBS1: Wide - binary stability 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.75]
# MASS RANGE: 500 ms < m < 10^3 ms
if constraints["WBS1"]==1:
    cName="Wide - binary stability 1"
    def WBS1(m):
```

```
        return (500*ms)/m
    tmin=log10(th(500*ms))
    tmax=log10(th(1e3*ms))
    Global.append(FB(tmin,tmax,WBS1))
    ActiveConstraints.append(cName)

# WBS2: Wide - binary stability 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.75]
# MASS RANGE: 10^3 ms < m < 10^8 ms
if constraints["WBS2"]==1:
    cName="Wide - binary stability 2"
    def WBS2(m):
        return 0.5
    tmin=log10(th(1e3*ms))
    tmax=log10(th(1e8*ms))
    Global.append(FB(tmin,tmax,WBS2))
    ActiveConstraints.append(cName)

# DGC1: Disruption of globular clusters 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.76]
# MASS RANGE: 3^10^4 ms < m < 10^6 ms
if constraints["DGC1"]==1:
    cName="Disruption of globular clusters 1"
    def DGC1(m):
        return (3e4*ms)/m
    tmin=log10(th(3e4*ms))
    tmax=log10(th(1e6*ms))
    Global.append(FB(tmin,tmax,DGC1))
    ActiveConstraints.append(cName)

# DGC2: Disruption of globular clusters 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.76]
# MASS RANGE: 10^6 ms < m < 10^11 ms
if constraints["DGC2"]==1:
    cName="Disruption of globular clusters 2"
    def DGC2(m):
        return 0.03
    tmin=log10(th(1e6*ms))
    tmax=log10(th(1e11*ms))
    Global.append(FB(tmin,tmax,DGC2))
    ActiveConstraints.append(cName)

# ERI1: Eridanus II - survival of star clusters 1
```

```
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.77]
# MASS RANGE: m < 10^4 ms
if constraints["ERI1"]==1:
    cName="Eridanus II - survival of star clusters 1"
    def ERI1(m):
        return (m/(3.7*ms))**-1/(1.1-0.1*np.log(m/ms))
    tmin=min1
    tmax=log10(th(ms*1e4))
    Global.append(FB(tmin,tmax,ERI1))
    ActiveConstraints.append(cName)

# ERI2: Eridanus II - survival of star clusters 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.77]
# MASS RANGE: m > 10^4 ms
if constraints["ERI2"]==1:
    cName="Eridanus II - survival of star clusters 2"
    def ERI2(m):
        return m/(1e6*ms)
    tmin=log10(th(ms*1e4))
    tmax=max1
    Global.append(FB(tmin,tmax,ERI2))
    ActiveConstraints.append(cName)

# DHTS1: Disk heating and tidal streams 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.78]
# MASS RANGE: m < 3*10^9 ms
if constraints["DHTS1"]==1:
    cName="Disk heating and tidal streams 1"
    def DHTS1(m):
        return (3e6*ms)/m
    tmin=min1
    tmax=log10(th(ms*3e9))
    Global.append(FB(tmin,tmax,DHTS1))
    ActiveConstraints.append(cName)

# DHTS2: Disk heating and tidal streams 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.78]
# MASS RANGE: m > 3*10^9 ms
if constraints["DHTS2"]==1:
    cName="Disk heating and tidal streams 2"
    def DHTS2(m):
        return m/mhalo
    tmin=log10(th(ms*3*1e9))
```

```
    tmax=max1
    Global.append(FB(tmin,tmax,DHTS2))
    ActiveConstraints.append(cName)

# DFHO1: Dynamical friction on halo objects 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79]
# MASS RANGE: m < 5*10^5 ms
if constraints["DFHO1"]==1:
    cName="Dynamical friction on halo objects 1"
    def DFHO1(m):
        return (m/(2e4*ms))**(-10/7)
    tmin=min1
    tmax=log10(th(ms*5e5))
    Global.append(FB(tmin,tmax,DFHO1))
    ActiveConstraints.append(cName)

# DFHO2: Dynamical friction on halo objects 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79]
# MASS RANGE: 5*10^5 ms < m < 2*10^6 ms
if constraints["DFHO2"]==1:
    cName="Dynamical friction on halo objects 2"
    def DFHO2(m):
        return (m/(4e4*ms))**-2
    tmin=log10(th(ms*5e5))
    tmax=log10(th(ms*2e6))
    Global.append(FB(tmin,tmax,DFHO2))
    ActiveConstraints.append(cName)

# DFHO3: Dynamical friction on halo objects 3
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79]
# MASS RANGE: 2*10^6 ms < m < 10^8 ms
if constraints["DFHO3"]==1:
    cName="Dynamical friction on halo objects 3"
    def DFHO3(m):
        return (m/(0.1*ms))**-0.5
    tmin=log10(th(ms*2e6))
    tmax=log10(th(ms*1e8))
    Global.append(FB(tmin,tmax,DFHO3))
    ActiveConstraints.append(cName)

# DFHO4: Dynamical friction on halo objects 4
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.79]
# MASS RANGE: m > 10^8 ms
```

```
if constraints["DFHO4"]==1:
    cName="Dynamical friction on halo objects 4"
    def DFHO4(m):
        return m/mhalo
    tmin=log10(th(ms*1e8))
    tmax=max1
    Global.append(FB(tmin,tmax,DFHO4))
    ActiveConstraints.append(cName)

# DDGC1: Disruption/distortion of galaxies in clusters 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.80]
# MASS RANGE: 10^10 ms < m < 10^11 ms
if constraints["DDGC1"]==1:
    cName="Disruption/distortion of galaxies in clusters
        1"
    def DDGC1(m):
        return 1e10*ms/m
    tmin=log10(th(ms*1e10))
    tmax=log10(th(ms*1e11))
    Global.append(FB(tmin,tmax,DDGC1))
    ActiveConstraints.append(cName)

# DDGC2: Disruption/distortion of galaxies in clusters 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.80]
# MASS RANGE: 10^11 ms < m < 10^13 ms
if constraints["DDGC2"]==1:
    cName="Disruption/distortion of galaxies in clusters
        2"
    def DDGC2(m):
        return 0.1
    tmin=log10(th(ms*1e11))
    tmax=log10(th(ms*1e13))
    Global.append(FB(tmin,tmax,DDGC2))
    ActiveConstraints.append(cName)

# DDGC3: Disruption/distortion of galaxies in clusters 3
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.80]
# MASS RANGE: m > 10^13 ms
if constraints["DDGC3"]==1:
    cName="Disruption/distortion of galaxies in clusters
        3"
    def DDGC3(m):
        return m/(2*1e14*ms)
```

```
    tmin=log10(th(ms*1e13))
    tmax=max1
    Global.append(FB(tmin,tmax,DDGC3))
    ActiveConstraints.append(cName)

# MW1: MW type galaxies 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.88]
# MASS RANGE: 10^6 ms < m < 10^9 ms
if constraints["MW1"]==1:
    cName="MW type galaxies 1"
    def MW1(m):
        return 1e6*ms/m
    tmin=log10(th(ms*1e6))
    tmax=log10(th(ms*1e9))
    Global.append(FB(tmin,tmax,MW1))
    ActiveConstraints.append(cName)

# MW2: MW type galaxies 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.88]
# MASS RANGE: 10^9 ms < m < 10^12 ms
if constraints["MW2"]==1:
    cName="MW type galaxies 2"
    def MW2(m):
        return m/(1e12*ms)
    tmin=log10(th(ms*1e9))
    tmax=log10(th(ms*1e12))
    Global.append(FB(tmin,tmax,MW2))
    ActiveConstraints.append(cName)

# DG1: Dwarf type galaxies 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.89]
# MASS RANGE: 10^5 ms < m < 3*10^7 ms
if constraints["DG1"]==1:
    cName="Dwarf type galaxies 1"
    def DG1(m):
        return 1e5*ms/m
    tmin=log10(th(ms*1e5))
    tmax=log10(th(ms*3e7))
    Global.append(FB(tmin,tmax,DG1))
    ActiveConstraints.append(cName)

# DG2: Dwarf type galaxies 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.89]
```

```
# MASS RANGE: 3*10^7 ms < m < 10^10 ms
if constraints["DG2"]==1:
    cName="Dwarf type galaxies 2"
    def DG2(m):
        return m/(1e10*ms)
    tmin=log10(th(ms*3e7))
    tmax=log10(th(ms*1e10))
    Global.append(FB(tmin,tmax,DG2))
    ActiveConstraints.append(cName)

# CG1: Clusters of galaxies 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.90]
# MASS RANGE: 10^7 ms < m < 3*10^10 ms
if constraints["CG1"]==1:
    cName="Clusters of galaxies 1"
    def CG1(m):
        return (1e7*ms)/m
    tmin=log10(th(ms*1e7))
    tmax=log10(th(ms*3e10))
    Global.append(FB(tmin,tmax,CG1))
    ActiveConstraints.append(cName)

# CG2: Clusters of galaxies 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.90]
# MASS RANGE: 3*10^10 ms < m < 10^14 ms
if constraints["CG2"]==1:
    cName="Clusters of galaxies 2"
    def CG2(m):
        return m/(1e14*ms)
    tmin=log10(th(ms*3e10))
    tmax=log10(th(ms*1e14))
    Global.append(FB(tmin,tmax,CG2))
    ActiveConstraints.append(cName)

# FBC1: First baryonic clouds 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.91]
# MASS RANGE: 10^3 ms < m < 3*10^4 ms
if constraints["FBC1"]==1:
    cName="First baryonic clouds 1"
    def FBC1(m):
        return (1e3*ms)/m
    tmin=log10(th(ms*1e3))
    tmax=log10(th(ms*3e4))
```

```
    Global.append(FB(tmin,tmax,FBC1))
    ActiveConstraints.append(cName)

# FBC2: First baryonic clouds 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.91]
# MASS RANGE: 3*10^4 ms < m < 10^6 ms
if constraints["FBC2"]==1:
    cName="First baryonic clouds 2"
    def FBC2(m):
        return m/(1e6*ms)
    tmin=log10(th(ms*3e4))
    tmax=log10(th(ms*1e6))
    Global.append(FB(tmin,tmax,FBC2))
    ActiveConstraints.append(cName)

# WMAPCMB1: WMAP - CMB anisotropies/distortions 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.92]
# MASS RANGE: 30 ms < m < 10^4 ms
if constraints["WMAPCMB1"]==1:
    cName="WMAP - CMB anisotropies/distortions 1"
    def WMAPCMB1(m):
        return (m/(30*ms))**(-2)
    tmin=log10(th(ms*30))
    tmax=log10(th(ms*1e4))
    Global.append(FB(tmin,tmax,WMAPCMB1))
    ActiveConstraints.append(cName)

# WMAPCMB2: WMAP - CMB anisotropies/distortions 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.92]
# MASS RANGE: 10^4 ms < m < 10^11 ms
if constraints["WMAPCMB2"]==1:
    cName="WMAP - CMB anisotropies/distortions 2"
    def WMAPCMB2(m):
        return 1e-5
    tmin=log10(th(ms*1e4))
    tmax=log10(th(ms*1e11))
    Global.append(FB(tmin,tmax,WMAPCMB2))
    ActiveConstraints.append(cName)

# WMAPCMB3: WMAP - CMB anisotropies/distortions 3
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.92]
# MASS RANGE: m > 10^11 ms
if constraints["WMAPCMB3"]==1:
```

```
    cName="WMAP - CMB anisotropies/distortions 3"
    def WMAPCMB3(m):
        return m/(1e16*ms)
    tmin=log10(th(ms*1e11))
    tmax=max1
    Global.append(FB(tmin,tmax,WMAPCMB3))
    ActiveConstraints.append(cName)

# FIRASCMB1: FIRAS - CMB anisotropies/distortions 1
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.93]
# MASS RANGE: 1 ms < m < 10^3 ms
if constraints["FIRASCMB1"]==1:
    cName="FIRAS - CMB anisotropies/distortions 1"
    def FIRASCMB1(m):
        return (m/ms)**(-2)
    tmin=log10(th(ms*1))
    tmax=log10(th(ms*1e3))
    Global.append(FB(tmin,tmax,FIRASCMB1))
    ActiveConstraints.append(cName)

# FIRASCMB2: FIRAS - CMB anisotropies/distortions 2
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.93]
# MASS RANGE: 10^3 ms < m < 10^14 ms
if constraints["FIRASCMB2"]==1:
    cName="FIRAS - CMB anisotropies/distortions 2"
    def FIRASCMB2(m):
        return 0.015
    tmin=log10(th(ms*1e3))
    tmax=log10(th(ms*1e14))
    Global.append(FB(tmin,tmax,FIRASCMB2))
    ActiveConstraints.append(cName)

# FIRASCMB3: FIRAS - CMB anisotropies/distortions 3
# ref: Carr B. et al., 2021, RPPh, 84, 116902 [eq.93]
# MASS RANGE: m > 10^14 ms
if constraints["FIRASCMB3"]==1:
    cName="FIRAS - CMB anisotropies/distortions 3"
    def FIRASCMB3(m):
        return m/(1e16*ms)
    tmin=log10(th(ms*1e14))
    tmax=max1
    Global.append(FB(tmin,tmax,FIRASCMB3))
    ActiveConstraints.append(cName)
```

```
# CMBDL: CMB dipole limit
# ref: Carr B., Kuhnel F., Visinelli L., 2021, MNRAS, 501, 2029 [Fig.1]
# MASS RANGE: 10^18.5 ms < m < 10^21.5 ms
if constraints["CMBDL"]==1:
    cName="CMB dipole limit"
    def CMBDL(m):
        return 10**(-1.5*log10(m)+78.7)
    tmin=log10(th(ms*10**18.5))
    tmax=log10(th(ms*10**21.5))
    Global.append(FB(tmin,tmax,CMBDL))
    ActiveConstraints.append(cName)

# 7 - Frames for the different figures
#frame for f(t)
figfTime,ax1=plt.subplots()
#frame for f(M)
figfMass,ax2=plt.subplots()
#frame for betaprime(M)
figbetaprimeMass,ax3=plt.subplots()
#frame for betaprime(T)
figbetaprimeTime,ax4=plt.subplots()
#frame for beta(M)
figbetaMass,ax5=plt.subplots()
#frame for beta(M)
figbetaTime,ax6=plt.subplots()

# Axis limits, titles and subtitles
ax1.set_xlim(-30, 17)
ax1.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{t} / 1\,\mathrm{s}\right)$")
ax1.set_ylim(-12, 2)
ax1.set_ylabel(r"$\mathrm{Log}_{10}\left(\mathrm{f}\,(\mathrm{t})\right)$")

ax2.set_xlim(8, 55)
ax2.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{M} / 1\,\mathrm{g}\right)$")
ax2.set_ylim(-12, 2)
ax2.set_ylabel(r"$\mathrm{Log}_{10}\left(\mathrm{f}\,(\mathrm{M})\right)$")

```

```
ax3.set_xlim(8, 55)
ax3.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{M} / 1\,\mathrm{g}\right)$")
ax3.set_ylim(-30, 0)
ax3.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta' \, (\mathrm{M})\right)$")

ax4.set_xlim(-30, 17)
ax4.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{t} / 1\,\mathrm{s}\right)$")
ax4.set_ylim(-30, 0)
ax4.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta' \, (\mathrm{t})\right)$")

ax5.set_xlim(8, 55)
ax5.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{M} / 1\,\mathrm{g}\right)$")
ax5.set_ylim(-30, 0)
ax5.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta(\mathrm{M})\right)$")

ax6.set_xlim(-30, 17)
ax6.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{t} / 1\,\mathrm{s}\right)$")
ax6.set_ylim(-30, 0)
ax6.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta(\mathrm{t})\right)$")

# 8 - Generating the global figures
for i in range(len(Global)):
    if constraints.get(constraintsnames[i], 0) != 1:
        continue

    w = [(Global[i][j][0], Global[i][j][2]) for j in range(len(Global[i]))]
    ax1.plot(*zip(*w))

    w = [(Global[i][j][1], Global[i][j][2]) for j in range(len(Global[i]))]
    ax2.plot(*zip(*w))

    w = [(Global[i][j][1], Global[i][j][3]) for j in range(len(Global[i]))]
```

```
    ax3.plot(*zip(*w))

    w = [(Global[i][j][0], Global[i][j][3]) for j in range(len(Global[i]))]
    ax4.plot(*zip(*w))

    w = [(Global[i][j][1], Global[i][j][4]) for j in range(len(Global[i]))]
    ax5.plot(*zip(*w))

    w = [(Global[i][j][0], Global[i][j][4]) for j in range(len(Global[i]))]
    ax6.plot(*zip(*w))

# 9 - Generate figures with the minimum values only
minf_time = []
minf_mass = []
minbl_time = []
minbl_mass = []
minb_time = []
minb_mass = []
minCName = []

i = 0  # pointing to the first instant (or mass)
while i < len(Global[0]):

    # Identify the minimum values across the 56 constraint curves at this point i
    list_f = []
    list_bl = []
    list_b = []

    for j in range(len(Global)):
        list_f.append(Global[j][i][2])
        list_bl.append(Global[j][i][3])
        list_b.append(Global[j][i][4])

    minf_time.append((Global[0][i][0], min(list_f)))
    minf_mass.append((Global[0][i][1], min(list_f)))
    minbl_time.append((Global[0][i][0], min(list_bl)))
    minbl_mass.append((Global[0][i][1], min(list_bl)))
    minb_time.append((Global[0][i][0], min(list_b)))
    minb_mass.append((Global[0][i][1], min(list_b)))
```

```

    wx1 = [Global[j][i][4] for j in range(len(Global))]
    resMIN = "-"

    for j in range(len(Global)):
        if Global[j][i][4] == min(wx1):
            if len(Global[j][i]) > 5:
                resMIN = Global[j][i][5]
            else:
                resMIN = "Empty"

    minCName.append([
        Global[0][i][0],
        resMIN
    ])

    i=i+1

# Generate graphics of the minimum values
figftimeMin,ax7=plt.subplots()
figfmassMin,ax8=plt.subplots()
figbetaprimetimeMin,ax9=plt.subplots()
figbetaprimemassMin,ax10=plt.subplots()
figbetatimeMin,ax11=plt.subplots()
figbetamassMin,ax12=plt.subplots()

# Axis limits, titles and subtitles
ax7.plot(*zip(*minf_time))
ax8.plot(*zip(*minf_mass))
ax9.plot(*zip(*minbl_time))
ax10.plot(*zip(*minbl_mass))
ax11.plot(*zip(*minb_time))
ax12.plot(*zip(*minb_mass))

ax7.set_xlim(-30, 17)
ax7.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{t} / 1\,\mathrm{s}\right)$")
ax7.set_ylim(-12, 2)
ax7.set_ylabel(r"$\mathrm{Log}_{10}\left(\mathrm{f}\,(\mathrm{t})\right)$")

ax8.set_xlim(8, 55)
ax8.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{M} /
```

```
    1\,\mathrm{g}\right)$")
ax8.set_ylim(-12, 2)
ax8.set_ylabel(r"$\mathrm{Log}_{10}\left(\mathrm{f}\,(\
    mathrm{M})\right)$")

ax9.set_xlim(-30, 17)
ax9.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{t} /
    1\,\mathrm{s}\right)$")
ax9.set_ylim(-30, 0)
ax9.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta' \, (\
    mathrm{t})\right)$")

ax10.set_xlim(8, 55)
ax10.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{M} /
    1\,\mathrm{g}\right)$")
ax10.set_ylim(-30, 0)
ax10.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta' \, (\
    mathrm{M})\right)$")

ax11.set_xlim(-30, 17)
ax11.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{t} /
    1\,\mathrm{s}\right)$")
ax11.set_ylim(-30, 0)
ax11.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta(\mathrm{
    t})\right)$")

ax12.set_xlim(8, 55)
ax12.set_xlabel(r"$\mathrm{Log}_{10}\left(\mathrm{M} /
    1\,\mathrm{g}\right)$")
ax12.set_ylim(-30, 0)
ax12.set_ylabel(r"$\mathrm{Log}_{10}\left(\beta(\mathrm{
    M})\right)$")

# 10 - Display all figures
plt.show()

# Create reduced list of minCName with only the segments
    where the applied constraint changes

filtered_minCName = [minCName[0]]
for j in range(1, len(minCName)):
    if minCName[j][1] != minCName[j-1][1]:
        filtered_minCName.append(minCName[j])
```

```

# Display used constraints
print("Used Constraints:")
try:
    from IPython.display import display
    import pandas as pd
    display(pd.DataFrame(ActiveConstraints))
except:
    print(ActiveConstraints)

print("Relevant Constraints:")
try:
    import pandas as pd
    display(pd.DataFrame(filtered_minCName, columns=["logTime", "Constraint"]))
except:
    for prime in filtered_minCName:
        print(prime)

# 11 - SAVE FILES and GRAPHICS
# Saving data
# Saving data na pasta de trabalho atual
np.savetxt("python-ActiveConstraints.txt", ActiveConstraints, fmt='%s')
np.savetxt("python-RelevantConstraints.txt", filtered_minCName, fmt='%s\t%s')
np.savetxt("python-constraintsFT.txt", minf_time, fmt='%.3f %.15f')
np.savetxt("python-constraintsFM.txt", minf_mass, fmt='%.3f %.15f')
np.savetxt("python-constraintsBLT.txt", minbl_time, fmt='%.3f %.15f')
np.savetxt("python-constraintsBLM.txt", minbl_mass, fmt='%.3f %.15f')
np.savetxt("python-constraintsBT.txt", minb_time, fmt='%.3f %.15f')
np.savetxt("python-constraintsBM.txt", minb_mass, fmt='%.3f %.15f')

# Saving graphics (.pdf and .eps) na mesma pasta
figfTime.savefig("python-constraintsFigFTglobal.pdf")
figfMass.savefig("python-constraintsFigFMglobal.pdf")
figbetaprimeMass.savefig("python-
```

```
constraintsfigbetaprimeMglobal.pdf")
figbetaprimeTime.savefig("python-constraintsfigbetaprimeTglobal.pdf")
figbetaMass.savefig("python-constraintsFigBMglobal.pdf")
figbetaTime.savefig("python-constraintsFigBTglobal.pdf")

figftimeMin.savefig("python-constraintsFigFT.pdf")
figfmassMin.savefig("python-constraintsFigFM.pdf")
figbetaprimetimeMin.savefig("python-constraintsfigbetaprimeT.pdf")
figbetaprimemassMin.savefig("python-constraintsfigbetaprimeM.pdf")
figbetatimeMin.savefig("python-constraintsFigBT.pdf")
figbetamassMin.savefig("python-constraintsFigBM.pdf")

figftimeMin.savefig("python-constraintsFigFTglobalMin.eps", format='eps')
figfmassMin.savefig("python-constraintsFigFMglobalMin.eps", format='eps')
figbetaprimetimeMin.savefig("python-constraintsfigbetaprimeTglobalMin.eps", format='eps')
figbetaprimemassMin.savefig("python-constraintsfigbetaprimeMglobalMin.eps", format='eps')
figbetatimeMin.savefig("python-constraintsFigBTglobalMin.eps", format='eps')
figbetamassMin.savefig("python-constraintsFigBMglobalMin.eps", format='eps')
```

Referências

Abreu P., Adam W., Adye T., Adzic P., Ajinenko I., Albrecht Z., Alderweireld T., et al., 2000, *Limits on the masses of supersymmetric particles at* $\sqrt{s}$ *=189 GeV*, PhLB, 489, 38. doi:10.1016/S0370-2693(00)00888-1

Burrows A., Wang T., Vartanyan D., 2024, *Channels of Stellar-mass Black Hole Formation*, arXiv, arXiv:2412.07831. doi:10.48550/arXiv.2412.07831

Carr B. J., 1975, *The primordial black hole mass spectrum*, ApJ, 201, 1. doi:10.1086/153853

Carr B., Kohri K., Sendouda Y., Yokoyama J., 2021, *Constraints on Primordial Black Holes*, RPPh, 84, 116902. doi:10.1088/1361-6633/ac1e31

Carr B., Kühnel F., Visinelli L., 2021, *Constraints on Stupendously Large Black Holes*, MNRAS, 501, 2029. doi:10.1093/mnras/staa3651

Chisholm J. R., Dodelson S., Kolb E. W., 2003, *Stellar-Mass Black Holes in the Solar Neighborhood*, ApJ, 596, 437. doi:10.1086/377628

Freitas S. M. A., Sobrinho J. L. G., 2021, *A list of 48 Binary Black Hole mergers*, Relatório do trabalho desenvolvido ao abrigo do programa Estágios de Verão 2021 da Secretaria Regional da Educação - Direção Regional de Juventude e Desporto, 38pp; Universidade da Madeira 2021.

Gaia Collaboration, Panuzzo P., Mazeh T., Arenou F., Holl B., Caffau E., Jorissen A., et al., 2024, *Discovery of a dormant 33 solar-mass black hole in pre-release Gaia astrometry*, A&A, 686, L2. doi:10.1051/0004-6361/202449763

Gardner S., McDermott S. D., Yanny B., 2021, *The Milky Way, Coming into Focus: Precision Astrometry Probes its Evolution, and its Dark Matter*, PrPNP, 121, 103904. doi:10.1016/j.ppnp.2021.103904

Gibilisco M., 1997, *The influence of quarks and gluons jets coming from primordial black holes on the reionization of the Universe*, IJMPA, 12, 4167. doi:10.1142/S0217751X97002280

Gravity Collaboration, Abuter R., Aimar N., Amaro Seoane P., Amorim A., Bauböck M., Berger J. P., et al., 2023, *Polarimetry and astrometry of NIR flares as event horizon scale, dynamical probes for the mass of Sgr A**, A&A, 677, L10. doi:10.1051/0004-6361/202347416

Hawking S., 1971, *Gravitationally collapsed objects of very low mass*, MNRAS, 152, 75. doi:10.1093/mnras/152.1.75

Kormendy J., Ho L. C., 2013, *Coevolution (Or Not) of Supermassive Black Holes and Host Galaxies*, ARA&A, 51, 511. doi:10.1146/annurev-astro-082708-101811

Matplotlib Developers, 2023, *Matplotlib (Version 3.8)*

Niemeyer J. C., Jedamzik K., 1999, *Dynamics of primordial black hole formation*, PhRvD, 59, 124013. doi:10.1103/PhysRevD.59.124013

Nóbrega S., Sobrinho J. L. G., 2020, *A list of 101 Supermassive Black Holes*, Relatório do trabalho desenvolvido ao abrigo do programa Estágios de Verão 2019 da Secretaria Regional da Educação - Direção Regional de Juventude e Desporto, 18pp; Universidade da Madeira 2020.

Novikov I. D., Polnarev A. G., Starobinskii A. A., Zeldovich I. B., 1979, *Primordial black holes*, A&A, 80, 104

NumPy Developers, 2023, *NumPy (Version 1.26)*

Polarski D., Dolgov A., 2001, *Classicality of Primordial Fluctuations and Primordial Black Holes*, IJMPD, 10, 927. doi:10.1142/S021827180100161X

Python Software Foundation, 2023, *Python (Version 3.11)*

Saikawa K., Shirai S., 2018, *Primordial gravitational waves, precisely: The role of thermodynamics in the Standard Model*, JCAP, 2018, 035. doi:10.1088/1475-7516/2018/05/035

Sobrinho J. L. G., Augusto P., 2014, *Direct detection of black holes via electromagnetic radiation*, MNRAS, 441, 2878. doi:10.1093/mnras/stu786

Sobrinho J. L. G., Augusto P., 2024, *Primordial intermediate and supermassive black hole formation during the electron-positron annihilation epoch*, MNRAS, 531, L40. doi:10.1093/mnrasl/slae028

Sobrinho J. L. G., Augusto P., Gonçalves A. L., 2016, *New thresholds for primordial black hole formation during the QCD phase transition*, MNRAS, 463, 2348. doi:10.1093/mnras/stw2138

University of Tartu, 2023, *Thonny IDE, Version 4.1.7*, Tartu, Estonia.

van den Bosch R. C. E., de Zeeuw P. T., 2010, *Estimating black hole masses in triaxial galaxies*, MNRAS, 401, 1770. doi:10.1111/j.1365-2966.2009.15832.x

Volonteri M., 2010, *Formation of supermassive black holes*, A&ARv, 18, 279. doi:10.1007/s00159-010-0029-x

Wolfram Research Inc., 2005, *Mathematica, Version 5.0*, Champaign, IL.