ACROPOLIS: A generiC fRamework fOr Photodisintegration Of LIght elementS

Paul Frederik Depta[a], Marco Hufnagel[a,b], and Kai Schmidt-Hoberg[a]

[a] *DESY, Notkestraße 85, D-22607 Hamburg, Germany*

[b] *Service de Physique Théorique, Université Libre de Bruxelles, Boulevard du Triomphe, CP225, B-1050 Brussels, Belgium*

E-mail: frederik.depta@desy.de, marco.hufnagel@desy.de, kai.schmidt-hoberg@desy.de

ABSTRACT: The remarkable agreement between observations of the primordial light element abundances and the corresponding theoretical predictions within the standard cosmological history provides a powerful method to constrain physics beyond the standard model of particle physics (BSM). For a given BSM model these primordial element abundances are generally determined by *(i) Big Bang Nucleosynthesis* and *(ii) possible subsequent disintegration processes*. The latter potentially change the abundances due to late-time high-energy injections which may be present in these scenarios. While there are a number of public codes for the first part, no such code is currently available for the second. Here we close this gap and present ACROPOLIS, A generiC fRamework fOr Photodisintegration Of LIght elementS. The widely discussed cases of decays as well as annihilations can be run without prior coding knowledge within example programs. Furthermore, due to its modular structure, ACROPOLIS can easily be extended also to other scenarios.

For the most recent version of this manual, please download it directly from the GitHub repository at https://github.com/skumblex/acropolis.

Contents

1 Introduction

The remarkable overall agreement between the inferred abundances of light elements and the corresponding predictions within the standard model of particle physics (SM) implies that any deviation from standard cosmology for sub-MeV temperatures is strongly constrained [1–4]. On the other hand, there is overwhelming evidence for the existence of dark matter (DM) suggesting that physics beyond the SM (BSM) is necessary to explain all phenomena we observe in nature. To evaluate the viability of a given BSM model it is therefore crucial to calculate its effect on the primordial element abundances and to compare them with the observationally inferred values.

There are two different temperature ranges in the early universe which determine the various element abundances: Once the universe has cooled down to about 1 MeV – at which point the typical energy of photons in the thermal bath has dropped significantly below the relevant binding energies – protons and neutrons start to fuse into light nuclei such as deuterium, helium, and lithium. This process of *Big Bang Nucleosynthesis* (BBN) can be tracked numerically by solving the appropriate Boltzmann equations describing the various fusion processes and a number of public codes such as `AlterBBN` [5], `PArthENoPE` [6], and `PRIMAT` [7] are available for this task. For most scenarios, the fusion processes have completed at about 10 keV and the nuclear abundances remain frozen at their asymptotic values until other processes become effective that might further change the abundances. For the standard cosmological history this would only happen at very late times, e.g. due to a reprocessing of the abundances in stellar fusion reactions.

However, many BSM models predict late-time decays or residual annihilations of dark sector particles which can lead to subsequent nuclear disintegration processes due to electromagnetic or hadronic showers, further changing the 'would-be' abundance values from BBN. For rather *heavy* dark sectors both photodisintegration and hadrodisintegration are relevant and the treatment of photodisintegration is simplified by the fact that the resulting photon spectrum has a universal form for sufficiently large injection energies depending only on the injection time and total energy injected [8]. Recently however the idea of *light* dark sectors with masses in the MeV to GeV range has attracted a lot of attention [9–22] and the light element abundances have been studied for such setups [23–28]. What makes the correct treatment of such scenarios more involved is that the coupled evolution equations of all particles that can emerge in electromagnetic decays have to be solved explicitly, as the approximation of the 'universal photon spectrum' breaks down if the energy of the initial decay products is too low [29]. However, unlike the case for BBN, there is currently no code available to treat possible late-time modifications of the abundances due to photodisintegration. Here we close this gap and present `ACROPOLIS`, `A` generi`C` f`R`amework f`O`r `P`hotodisintegration `O`f `LI`ght element`S`. The widely discussed cases of decays as well as annihilations are already implemented in example programs which can be run without prior coding knowledge. Furthermore, due to its modular structure, `ACROPOLIS` can easily be extended also to other scenarios and could also be linked to larger computational frameworks such as `CosmoBIT` [30].

2 Theoretical background

We follow the procedure detailed in [24], which generalised the one from [29] by not only including photons but also electrons and positrons. To render this manual self-contained we repeat all relevant steps. Following the convention in the literature on photodisintegration, we consider the phase-space distribution function f_x of particles $x \in \{e^\pm, \gamma\}$ differential in the energy E. Taking into account the degrees of freedom of x, g_x, this is related to the more conventional distribution f_x differential in the momentum p via

$$f_x(E) = g_x f_x(p) \frac{E\,p}{2\pi^2} \,, \tag{2.1}$$

where $E = \sqrt{m_x^2 + p^2}$ is the energy with mass m_x and p is the momentum.

2.1 Electromagnetic cascade

The late-time injection of high-energetic electromagnetic particles into the SM plasma induces an electromagnetic cascade that leads to non-thermal parts of the photon, electron, and positron spectra. Denoting particles from the thermal background with a subscript 'th' the most relevant interactions are

1. Double photon pair creation $\gamma\gamma_{\text{th}} \to e^+e^-$,

2. Photon-photon scattering $\gamma\gamma_{\text{th}} \to \gamma\gamma$,

3. Bethe-Heitler pair creation $\gamma N \to e^+e^-N$ with $N \in \{^1\text{H}, {}^4\text{He}\}$,

4. Compton scattering $\gamma e^-_{\text{th}} \to \gamma e^-$, and

5. Inverse Compton scattering $e^\pm \gamma_{\text{th}} \to e^\pm \gamma$.

Other processes are suppressed by small number densities and can be neglected, cf. appendix A. Before turning to the appropriate Boltzmann equation for the description of the electromagnetic cascade, let us briefly comment on why photodisintegration is only possible for late-time and high-energy injections into the SM plasma. For photons with energies above the threshold for double photon pair creation $E^{\text{th}}_{e^+e^-} \simeq m_e^2/(22T)$ [31], where m_e is the electron mass and T is the photon temperature, this process is much more efficient than the other reactions, thus rapidly depleting these high-energy photons. This implies that any photodisintegration process can only occur if $E^{\text{th}}_{e^+e^-}$ is above the threshold for the various disintegration reactions, i.e. if T is small enough translating to

1. $T \lesssim 5.34\,\text{keV}$ for D-disintegration with $E^{\text{th}}_{\text{D}} \approx 2.22\,\text{MeV}$,

2. $T \lesssim 1.90\,\text{keV}$ for ^{3}H-disintegration with $E^{\text{th}}_{^3\text{H}} \approx 6.26\,\text{MeV}$,

3. $T \lesssim 2.16\,\text{keV}$ for ^{3}He-disintegration with $E^{\text{th}}_{^3\text{He}} \approx 5.49\,\text{MeV}$,

4. $T \lesssim 0.60\,\text{keV}$ for ^{4}He-disintegration with $E^{\text{th}}_{^4\text{He}} \approx 19.81\,\text{MeV}$,

5. $T \lesssim 3.21\,\text{keV}$ for ^{6}Li-disintegration with $E^{\text{th}}_{^6\text{Li}} \approx 3.70\,\text{MeV}$,

6. $T \lesssim 4.81 \, \text{keV}$ for ^{7}Li-disintegration with $E_{^7\text{Li}}^{\text{th}} \approx 2.47 \, \text{MeV}$, and

7. $T \lesssim 7.48 \, \text{keV}$ for ^{7}Be-disintegration with $E_{^7\text{Be}}^{\text{th}} \approx 1.59 \, \text{MeV}$.

We observe that photodisintegration will become effective for rather low temperatures where BBN has already finished, so that the two processes simply factorise. This implies in particular that the initial abundances for photodisintegration correspond to the final abundances of BBN. Even though we do not employ it within ACROPOLIS, let us also remark that for injection energies above $E_{e^+e^-}^{\text{th}}$, the non-thermal part of the photon spectrum is close to the universal spectrum [8]

$$
\mathrm{f}_{\gamma,\text{univ}}(E) \sim \begin{cases} K_0 (E/E_X)^{-3/2} & \text{for } E < E_X \ , \\ K_0 (E/E_X)^{-2} & \text{for } E_X < E < E_{e^+e^-}^{\text{th}} \ , \\ 0 & \text{for } E > E_{e^+e^-}^{\text{th}} \ , \end{cases} \tag{2.2}
$$

where $K_0 = E_0 E_X^{-2} [2 + \ln(E_{e^+e^-}^{\text{th}}/E_X)]^{-1}$ and $E_X = m_e^2/(80T)$. In ACROPOLIS we solve the full Boltzmann equation numerically in which case the cutoff at $E_{e^+e^-}^{\text{th}}$ is replaced by an exponential suppression reflecting the aforementioned suppression of photodisintegration reactions due to efficient double photon pair creation.

As the total interaction rates Γ_x are large compared to the Hubble rate H, the expansion of the universe can be neglected in the calculation of the non-thermal photon spectrum [8, 32]. Suppressing the t- and $T(t)$-dependencies, the Boltzmann equation reads

$$
\frac{\partial \mathrm{f}_x(E)}{\partial t} \simeq S_x(E) - \Gamma_x(E) \mathrm{f}_x(E) + \sum_{x'} \int_E^\infty \mathrm{d}E' \, K_{x' \to x}(E, E') \mathrm{f}_{x'}(E') \ , \tag{2.3}
$$

where $S_x(E)$ is the source term for the production of x with energy E, $\Gamma_x(E)$ is the total interaction rate for x with energy E, and $K_{x' \to x}(E, E')$ is the differential interaction rate going from particle x' with energy E' to particle x with energy E. These rates contain the electromagnetic cascade reactions listed above and are given in appendix A. The electromagnetic cascade reactions quickly establish a quasi-static equilibrium with $\partial \mathrm{f}_x/\partial t \simeq 0$ [8, 32], hence we search for solutions of the integral equation

$$
\mathrm{f}_x(E) = \frac{1}{\Gamma_x(E)} \left(S_x(E) + \sum_{x'} \int_E^\infty K_{x' \to x}(E, E') \mathrm{f}_{x'}(E') \, \mathrm{d}E' \right) \ . \tag{2.4}
$$

We consider source terms arising from monochromatic or continuous high-energy injections, with the former one being typically realised for decays or residual annihilations of non-relativistic particles into two-particle SM final states. Hence, we parameterize the source terms via

$$
S_x(E) = S_x^{(0)} \delta(E - E_0) + S_x^{(c)}(E) \ , \tag{2.5}
$$

where E_0 is the injection energy and $S_x^{(0)}$ ($S_x^{(c)}(E)$) encodes the discrete (continuous) part of the injection spectrum.

									E^{th} [MeV]
D	+	γ	$\to$	p	+	n				2.22
^{3}H	+	γ	$\to$	D	+	n				6.26
^{3}H	+	γ	$\to$	p	+	n	+	n		8.48
^{3}He	+	γ	$\to$	D	+	p				5.49
^{3}He	+	γ	$\to$	n	+	p	+	p		7.12
^{4}He	+	γ	$\to$	^{3}H	+	p				19.81
^{4}He	+	γ	$\to$	^{3}He	+	n				20.58
^{4}He	+	γ	$\to$	D	+	D				23.84
^{4}He	+	γ	$\to$	D	+	n	+	p		26.07
^{6}Li	+	γ	$\to$	^{4}He	+	n	+	p		3.70
^{6}Li	+	γ	$\to$	X	+	^3A				15.79
^{7}Li	+	γ	$\to$	^{3}H	+	^{4}He				2.47
^{7}Li	+	γ	$\to$	n	+	^{6}Li				7.25
^{7}Li	+	γ	$\to$	$2n$	+	p	+	^{4}He		10.95
^{7}Be	+	γ	$\to$	^{3}He	+	^{4}He				1.59
^{7}Be	+	γ	$\to$	p	+	^{6}Li				5.61
^{7}Be	+	γ	$\to$	$2p$	+	n	+	^{4}He		9.30

Table 1. Reactions and threshold energies from [8] for the processes we consider in eq. (2.8).

For the numerical solution of eq. (2.4) we subtract the term containing the δ-distribution by defining

$$\mathrm{F}_x(E) := \mathrm{f}_x(E) - \frac{S_x^{(0)} \delta(E - E_0)}{\Gamma_x(E)} \tag{2.6}$$

such that

$$\Gamma_x(E)\mathrm{F}_x(E) = S_x^{(c)}(E) + \sum_{x'} \left[\frac{K_{x' \to x}(E, E_0) S_{x'}^{(0)}}{\Gamma_{x'}(E_0)} + \int_E^\infty K_{x' \to x}(E, E') \mathrm{F}_{x'}(E') \, \mathrm{d}E' \right] . \tag{2.7}$$

2.2 Non-thermal nucleosynthesis

With the previously calculated photon spectra f_γ we can now evaluate the effect on the primordial light element abundances $N \in \{n, p, D, {}^3\mathrm{H}, {}^3\mathrm{He}, {}^4\mathrm{He}, {}^6\mathrm{Li}, {}^7\mathrm{Li}, {}^7\mathrm{Be}\}$. The corresponding Boltzmann equation is given by

$$\dot{Y}_N(t) = \sum_j Y_j(t) \int_0^\infty \mathrm{d}E \, \mathrm{f}_\gamma(t, E) \sigma_{j\gamma \to N}(E) - Y_N(t) \sum_{j'} \int_0^\infty \mathrm{d}E \, \mathrm{f}_\gamma(t, E) \sigma_{N\gamma \to j'}(E) , \tag{2.8}$$

where $Y_N = n_N/n_b$ with n_N (n_b) the number density of N (baryons) and σ_r the cross section for the reaction r. Here we implement all reactions that are shown in tab. 1 by adopting the analytical expressions for the rates $1 - 17$ from [8]; however, we modify the

prefactor of reaction 7 from 17.1 mb to 20.7 mb as suggested by [32] in order to match the most recent EXFOR data. Note that these disintegration reactions can be neglected in eq. (2.3) due to the low number density of nuclei, thus enabling us to calculate the photon spectrum without knowledge of the light element abundances.

3 Numerical solution techniques

3.1 Electromagnetic cascade

For the numerical solution of eq. (2.7) we can exploit the fact that only particles with energies less than the injection energy can be produced, leading to vanishing spectra for $E > E_0$. Moreover, we are only interested in energies above a minimal energy $E_{\min}$ given by the lowest threshold energy of the photodisintegration reactions. In the code we therefore set $E_{\min} = 1.5\,\text{MeV}$ ($E_{7\text{Be}}^{\text{th}} \approx 1.59\,\text{MeV}$). With the relevant energy range $[E_{\min}, E_0]$ we then define a grid of energies spaced evenly on a log-scale, $\epsilon_i := E_{\min} \times (E_0/E_{\min})^{i/(M-1)}$, where $i \in \{0, \ldots, M-1\}$, $(\epsilon_0, \epsilon_{M-1}) = (E_{\min}, E_0)$. By default we choose 150 points per decade, which is usually sufficient to ensure convergence for all parameter points. Evaluating eq. (2.7) at each individual grid point we then find [33]

$$
\Gamma_x(\epsilon_i)\text{F}_x(\epsilon_i) = S_x^{(\text{c})}(\epsilon_i) + \sum_{x'} \left[\frac{K_{x' \to x}(\epsilon_i, E_0)S_{x'}^{(0)}}{\Gamma_{x'}(E_0)} + \int_{\ln(\epsilon_i)}^{\ln(E_0)} \text{d}y\, e^y K_{x' \to x}(\epsilon_i, e^y)\text{F}_{x'}(e^y) \right]
$$

$$
\simeq S_x^{(\text{c})}(\epsilon_i) + \sum_{x'} \left[\frac{K_{x' \to X}(\epsilon_i, E_0)S_{x'}^{(0)}}{\Gamma_{X'}(E_0)} + \frac{\Delta y}{2} \left(2 \sum_{j=i+1}^{M-2} \epsilon_j K_{x' \to x}(\epsilon_i, \epsilon_j)\text{F}_{x'}(\epsilon_j) \right.\right.
$$

$$
\left.\left. + \epsilon_i K_{x' \to x}(\epsilon_i, \epsilon_i)\text{F}_{x'}(\epsilon_i) + E_0 K_{x' \to x}(\epsilon_i, E_0)\text{F}_{x'}(E_0) \right) \right],
\tag{3.1}
$$

with $\Delta y = \ln(E_0/E_{\min})/(M-1)$. In the last step, we have used the trapezoidal integration rule and the sum $\sum_{j=i+1}^{M-2}$ is understood to vanish for $i + 1 > M - 2$. This expression is valid for $i < M - 1$, while for $i = M - 1$ we simply have

$$
\text{F}_x(E_0) = \frac{S_x^{(\text{c})}(E_0)}{\Gamma_x(E_0)} + \sum_{x'} \frac{K_{x' \to x}(E_0, E_0)S_{x'}^{(0)}}{\Gamma_x(E_0)\Gamma_{x'}(E_0)} .
\tag{3.2}
$$

Assuming that $\text{F}_x(E_j)$ has already been calculated for $j > i$, eq. (3.1) can be interpreted as a linear system of three equations for the unknown variables $\text{F}_x(E_i)$. Consequently, by defining $\bar{\text{F}}(\epsilon_i) := \left[\text{F}_\gamma(\epsilon_i), \text{F}_{e^-}(\epsilon_i), \text{F}_{e^+}(\epsilon_i)\right]^{\text{T}}$, we have

$$
\bar{\text{F}}(\epsilon_i) = a(\epsilon_i) + B(\epsilon_i)\bar{\text{F}}(\epsilon_i)
\tag{3.3}
$$

with

$$
\begin{aligned}
\left[a(\epsilon_i)\right]_x \;:=\; \frac{1}{\Gamma_x(\epsilon_i)} \sum_{x'} \Bigg[& \frac{K_{x'\to x}(\epsilon_i, E_0) S_{x'}^{(0)}}{\Gamma_{x'}(E_0)} + \frac{\Delta y}{2}\Bigg(2 \sum_{j=i+1}^{M-2} \epsilon_j K_{x'\to x}(\epsilon_i, \epsilon_j) F_{x'}(\epsilon_j) \\
& + E_0 K_{x'\to x}(\epsilon_i, E_0) F_{x'}(E_0)\Bigg)\Bigg] + \frac{S_x^{(c)}(\epsilon_i)}{\Gamma_x(\epsilon_i)} \;,
\end{aligned}
\tag{3.4}
$$

$$
\left[B(\epsilon_i)\right]_{xx'} := \frac{\Delta y}{2} \frac{\epsilon_i K_{x'\to x}(\epsilon_i, \epsilon_i)}{\Gamma_x(\epsilon_i)} \;.
\tag{3.5}
$$

Given the knowledge of $\left[a(\epsilon_i)\right]_x$ and $\left[B(\epsilon_i)\right]_{xx'}$, the linear eq. (3.3) can then be solved using standard techniques to calculate the values of $\bar{F}(\epsilon_i)$. As $\left[a(\epsilon_i)\right]_x$ and $\left[B(\epsilon_i)\right]_{xx'}$ explicitly depend on $\bar{F}(\epsilon_j)$ for $j > i$, we start solving eq. (3.2) at $i = M-1$ to successively determine solutions of eq. (3.3) at $i < M-1$.

3.2 Non-thermal nucleosynthesis

For the numerical solution of eq. (2.8) we first define

$$
\Gamma_r(t) := \int_0^\infty f_\gamma(t, E) \sigma_r(E) \, dE \;,
\tag{3.6}
$$

which transforms eq. (2.8) into the form

$$
\dot{Y}_N(t) = \sum_j Y_j(t) \Gamma_{j\gamma\to N}(t) - Y_N(t) \sum_{j'} \Gamma_{N\gamma\to j'}(t) \;.
\tag{3.7}
$$

Consequently, by also defining $\bar{Y}(t) := \left[Y_n(t), Y_p(t), Y_D(t), ...\right]^T$ and after substituting $t \to T$ by means of the time-temperature relation, we find

$$
\frac{d\bar{Y}(T)}{dT} = \mathcal{R}(T) \bar{Y}(T)
\tag{3.8}
$$

with

$$
\left[\mathcal{R}(T)\right]_{NN'} := \frac{dt}{dT} \times \left[\Gamma_{N'\gamma\to N}(T) - \delta_{NN'} \sum_{j'} \Gamma_{N\gamma\to j'}(T) \right] \;.
\tag{3.9}
$$

Here, the matrix $\left[\mathcal{R}(T)\right]_{NN'}$ can be obtained via numerical integration, and eq. (3.8) is an ordinary system of differential equations, which is solved by

$$
\bar{Y}(T) = \exp\left(\int_{T_{\max}}^T \mathcal{R}(T') \, dT' \right) \bar{Y}_0
\tag{3.10}
$$

with the initial condition $\bar{Y}(T_{\max}) = \bar{Y}_0$ at some maximal (initial) temperature $T_{\max}$. The matrix exponential $\exp(\cdot)$ can be evaluated numerically by *(i)* diagonalising the matrix $\int_{T_0}^T \mathcal{R}(T') dT'$ with the corresponding unitary transformation $U_{\mathcal{R}}(T)$, *(ii)* taking the exponential of the diagonal matrix by exponentiating each eigenvalue individually, and *(iii)* transforming the resulting matrix back using $U_{\mathcal{R}}^{-1}(T)$.

4 Example models

In this section, we briefly describe the physics of the two example models which are implemented in `ACROPOLIS`. In the next section, we will then discuss how to utilise the corresponding python scripts `decay` and `annihilation`.

4.1 Decay of a decoupled MeV-scale BSM particle

As a first example model implemented in `ACROPOLIS`, we consider the non-relativistic decay of a decoupled MeV-scale BSM particle ϕ with mass m_ϕ and lifetime τ_ϕ, e.g. an MeV-scale mediator. This implementation closely follows [24, 34], but only considers a number density of ϕ fixed at a reference temperature T_0 that simply redshifts, i.e. remains comovingly constant, until it decays with a lifetime τ_ϕ. As pointed out in [34] this is in fact a consistent assumption for the parameter ranges where photodisintegration may be relevant, even when taking into account inverse decays of ϕ.

The source terms entering in eq. (2.5) for the non-relativistic decay of a decoupled BSM particle ϕ with mass m_ϕ and lifetime τ_ϕ are given by [24, 25, 35, 36]

$$E_0 = \frac{m_\phi}{2} \, , \tag{4.1}$$

$$S_\gamma^{(0)} = \mathrm{BR}_{\gamma\gamma} \times \frac{2n_\phi}{\tau_\phi} \, , \tag{4.2}$$

$$S_{e^-}^{(0)} = S_{e^+}^{(0)} = \mathrm{BR}_{e^+e^-} \times \frac{n_\phi}{\tau_\phi} \, , \tag{4.3}$$

$$S_\gamma^{(c)}(E) = \frac{S_{e^\pm}^{(0)}}{E_0} \times \frac{\alpha}{\pi} \frac{1 + (1-x)^2}{x} \ln\left(\frac{4E_0^2(1-x)}{m_e^2}\right) \times \Theta\left(1 - \frac{m_e^2}{4E_0^2} - x\right) \, , \tag{4.4}$$

$$S_{e^-}^{(c)}(E) = S_{e^+}^{(c)}(E) = 0 \, , \tag{4.5}$$

where n_ϕ is the number density of ϕ, $\mathrm{BR}_{\gamma\gamma}$ ($\mathrm{BR}_{e^+e^-}$) is the branching ratio for decays into two photons (electron-positron pairs), α is the fine-structure constant, m_e is the electron mass, and $x = E/E_0$. Here, the continuous part $S_\gamma^{(c)}(E)$ of the photon source term is solely due to final-state radiation of photons from injected electron-positron pairs. If there is no contribution from inverse decays to the abundance of ϕ below a reference temperature T_0, i.e. for $T_0 \ll m_\phi$, with corresponding time t_0, its number density can be parametrised as ($t \geq t_0$)

$$n_\phi(t) \simeq n_\phi(T_0) \times \left(\frac{R(t_0)}{R(t)}\right)^3 \exp\left(-\frac{t}{\tau_\phi}\right) \, , \tag{4.6}$$

where R is the scale factor.

Whenever relevant, photodisintegration provides a very powerful probe of decaying MeV-scale BSM particles, constraining even very small abundances. We can thus assume that the energy density in the universe is dominated by the SM (and at late times also DM) energy densities, i.e. $m_\phi n_\phi \ll \rho_{\mathrm{SM}} + \rho_{\mathrm{DM}}$, and that the time-temperature relation of the SM is not changed by any BSM processes for $t > t_0$. This enables us to provide tables for the SM (photon) temperature, its derivative w.r.t. time, the neutrino temperature, the Hubble rate,

and the scale factor. Further approximating that standard BBN is not changed by any BSM processes including the presence and decay of ϕ implies that no separate BBN calculation must be performed and the results of a calculation using `AlterBBN` v1.4 [5, 37] are provided. The photodisintegration calculation is performed for temperatures $T_{\max} \geq T \geq T_{\min}$ with

$$T_{\max} = T(t = \tau_\phi) \times 10^{1/2} \, , \tag{4.7}$$

$$T_{\min} = T(t = \tau_\phi) \times 10^{-3/2} \tag{4.8}$$

around the decay time giving results of very good accuracy.

With the python script `decay` we thus provide an example program to calculate the primordial light element abundances after photodisintegration due to the decay of a BSM particle ϕ with mass m_ϕ, lifetime τ_ϕ, a reference temperature T_0 with corresponding number density $n_\phi(T_0)$, and branching ratio $\mathrm{BR}_{e^+e^-}$ ($\mathrm{BR}_{\gamma\gamma}$) for decays in electron-positron pairs (two photons).

4.2 Residual annihilations of MeV-scale dark matter

For the second example model implemented in `ACROPOLIS`, we consider residual annihilations of DM (or more generally any annihilating BSM particle) following [26]. We assume that DM consists of a self-conjugate[1] fermion χ with mass m_χ and an abundance (number density n_χ) fixed such that $\Omega_{\mathrm{DM}}h^2 = 0.12$ [39]. The source terms entering in eq. (2.5) are then given by [26]

$$E_0 = m_\chi \, , \tag{4.9}$$

$$S_\gamma^{(0)} = \mathrm{BR}_{\gamma\gamma} \times \langle\sigma v\rangle n_\chi^2 \, , \tag{4.10}$$

$$S_{e^-}^{(0)} = S_{e^+}^{(0)} = \mathrm{BR}_{e^+e^-} \times \frac{1}{2}\langle\sigma v\rangle n_\chi^2 \, , \tag{4.11}$$

$$S_\gamma^{(c)}(E) = \frac{S_{e^\pm}^{(0)}}{E_0} \times \frac{\alpha}{\pi}\frac{1 + (1-x)^2}{x} \ln\left(\frac{4E_0^2(1-x)}{m_e^2}\right) \times \Theta\left(1 - \frac{m_e^2}{4E_0^2} - x\right) \, , \tag{4.12}$$

$$S_{e^-}^{(c)}(E) = S_{e^+}^{(c)}(E) = 0 \, , \tag{4.13}$$

where $\mathrm{BR}_{\gamma\gamma}$ ($\mathrm{BR}_{e^+e^-}$) is the branching ratio for annihilations into two photons (electron-positron pairs), α is the fine-structure constant, m_e is the electron mass, $x = E/E_0$, and we expand the thermally averaged annihilation cross section[2] in powers of the relative velocity v_{rel}

$$\langle\sigma v\rangle \simeq a + b\langle v_{\mathrm{rel}}^2\rangle \, . \tag{4.14}$$

Here, s-wave DM annihilations are dominated by a whereas p-wave DM annihilations are dominated by b. The thermally averaged relative velocity squared is given by

$$\langle v_{\mathrm{rel}}^2\rangle \simeq \frac{6T_\chi(T)}{m_\chi} \, , \tag{4.15}$$

---

[1]While only the case for self-conjugate DM is directly implemented in `ACROPOLIS`, the case of non self-conjugate DM differs only by a factor of 2 in $\langle\sigma v\rangle$. This factor is exact, unlike for freeze-out calculations [38].

[2]This is the sum of the one for annihilations in two photons and in electron-positron pairs, weighted according to their branching ratios.

where T_χ is the DM temperature evolving before and after kinetic decoupling from the photon heat bath at T^{kd} according to

$$T_\chi(T) = \begin{cases} T & \text{if } T \geq T^{\mathrm{kd}} \,, \\ T^{\mathrm{kd}}(R(T^{\mathrm{kd}})/R(T))^2 & \text{if } T < T^{\mathrm{kd}} \,. \end{cases} \tag{4.16}$$

Similar to section 4.1 we assume that the energy density in the universe is dominated by the SM and DM χ, and that the time-temperature relation of the SM is not influenced by the presence of χ or any other BSM particle for the temperatures relevant for photodisintegration. Also assuming standard BBN without any alterations due to BSM processes enables us to use the same data as before for the light element abundances as well as all temperatures, the Hubble rate, and the scale factor. The calculation is performed for temperatures $T_{\max} \geq T \geq T_{\min}$ with

$$T_{\max} = 2 \times \frac{m_e^2}{22 E_{\min}} \,, \tag{4.17}$$

$$T_{\min} = T_{\max} \times 10^{-4} \,, \tag{4.18}$$

where $E_{\min} = 1.5\,\mathrm{MeV}$ is just below the lowest disintegration threshold ($E_{7\mathrm{Be}}^{\mathrm{th}} \approx 1.59\,\mathrm{MeV}$). This range gives results of very good accuracy, as for larger temperatures the non-thermal photon spectrum is exponentially suppressed for energies above any disintegration threshold, while for smaller temperatures the annihilation rate is suppressed at least $\propto T^{-6}$.

To summarise, we provide the example program `annihilation` to calculate the primordial light element abundances after photodisintegration due to residual annihilations of DM particles χ with mass m_χ, s (p)-wave part a (b) of the thermally averaged annihilation cross section, kinetic decoupling temperature T^{kd}, and branching ratio into electron-positron pairs $\mathrm{BR}_{e^+e^-}$ (two photons $\mathrm{BR}_{\gamma\gamma}$).

5 Running ACROPOLIS

5.1 Installation

ACROPOLIS is open-source software, licensed under GPL3. The source code is publicly available and can be cloned from GitHub by executing the command

```
$ git clone https://github.com/skumblex/acropolis.git
```

which creates a new folder named `acropolis` with the repository inside of your current working directory. ACROPOLIS has been tested with `Python` version ≥ 3.7 on macOS 10.15 and Ubuntu 20.04, and the following packages must be available (older versions might also work but have not been thoroughly tested):

- `NumPy` $\geq 1.19.1$ (matrix manipulation)

- `SciPy` $\geq 1.5.2$ (numerical integration)

- Numba $\geq 0.51.2$ (just-in-time compilation)

The most recent versions of these dependencies can be collectively installed at user-level, i.e. without the need for root access, by executing the command

```
$ python3 -m pip install numpy, scipy, numba --user
```

If these dependencies conflict with those for other programs in your work environment, it is strongly advised to utilise the capabilities of Python's virtual environments.

Inside of the git repository there also is an optional database file `data/rates.db.gz`, which contains pre-calculated values for some of the computationally expensive reaction rates. By default, `ACROPOLIS` uses this file to speed up the calculation by $\sim \mathcal{O}(10\%)$[3]. However, if this behavior is not desired for some reason, i.e. when running on a batch system, `ACROPOLIS` also allows to disable the usage of this file (cf. sec. 6.3).

5.2 Using the predefined models

`ACROPOLIS` is designed to allow for an easy implementation of arbitrary models, but also contains reference implementations for the example models that were introduced in section 4. In fact, these two models and the corresponding scripts that are bundled with `ACROPOLIS` should already suffice to treat most conceivable scenarios, implying that – for most applications – no additional coding is required. In the present section we will therefore first discuss how to correctly utilise these reference models, and only afterwards discuss how to implement additional ones in section 6.

5.2.1 Running the wrapper scripts

Within the `ACROPOLIS` main directory there are two executables, `decay` and `annihilation`, which wrap the scenarios discussed in section 4.1 and section 4.2, respectively. Both of these files need to be called with six command-line arguments each, a list of which can be obtained by running the command of choice without any arguments at all. Using the notation from section 4, the required parameters are

- `decay`: $\quad m_\phi$ [MeV] $\quad \tau_\phi$ [s] $\quad T_0$ [MeV] $\quad n_\phi/n_\gamma|_{T_0} \quad \mathrm{BR}_{ee} \quad \mathrm{BR}_{\gamma\gamma}$

- `annihilation`: $\quad m_\chi$ [MeV] $\quad a$ [cm^3/s] $\quad b$ [cm^3/s] $\quad T_{\mathrm{kd}}$ [MeV] $\quad \mathrm{BR}_{ee} \quad \mathrm{BR}_{\gamma\gamma}$

For example, the command

```
$ ./decay 10 1e5 10 1e-10 0 1
```

calculates the abundances after photodisintegration in the presence of an unstable particle ϕ with mass $m_\phi = 10\,\mathrm{MeV}$, lifetime $\tau_\phi = 10^5\,\mathrm{s}$, and number density $n_\phi/n_\gamma|_{T_0} = 10^{-10}$ at $T_0 = 10\,\mathrm{MeV}$ decaying exclusively into two photons (so $\mathrm{BR}_{ee} = 0$ and $\mathrm{BR}_{\gamma\gamma} = 1$).

---

[3]Several benchmark points that show the runtime of the code with and without the additional database files, can be found in appendix B.

Running the executables, information regarding the current state of the calculation is provided and the final output is given by a 9x3 matrix comprising the final abundances after photodisintegration for the given parameter point. For our example above the corresponding output roughly looks as follows:

```
INFO    : Extracting and reading database files.
INFO    : Finished after 1.5s.
INFO    : Calculating non-thermal spectra and reaction rates.
INFO    : Finished after 22.2s.
INFO    : Running non-thermal nucleosynthesis.
INFO    : Finished after 10.4s.
```

	mean	high	low
n	1.32309e−08	1.29265e−08	1.35767e−08
p	7.53096e−01	7.53161e−01	7.53036e−01
H2	1.92530e−05	1.88101e−05	1.97562e−05
H3	5.53065e−08	2.28921e−08	3.76285e−07
He3	7.63949e−06	3.27977e−06	7.82619e−06
He4	6.17092e−02	6.16965e−02	6.17237e−02
Li6	8.39044e−15	2.70471e−14	1.30028e−15
Li7	1.98142e−11	7.64325e−12	6.47310e−11
Be7	3.24947e−10	1.29785e−10	3.05210e−10

The different columns in the matrix contain the abundances $Y_N = n_N/n_b$ for the nine nuclei $N \in \{n, p, {}^2\mathrm{H}, {}^3\mathrm{H}, {}^3\mathrm{He}, {}^4\mathrm{He}, {}^6\mathrm{Li}, {}^7\mathrm{Li}, {}^7\mathrm{Be}\}$ as indicated. In each column slightly different initial conditions are set, corresponding to the SM results that are obtained by running the public code `AlterBBN` [5, 37] with the mean (first column), high (second column), and low (third column) values of the implemented nuclear reaction rates.[4] The variation between the values of different columns can therefore be used to approximate the theoretical errors for different abundances.

Before comparing the output of `ACROPOLIS` with observationally inferred primordial abundances, it is important to note that the values returned by `ACROPOLIS` do not fully take into account the three decay channels ${}^3\mathrm{H} \to {}^3\mathrm{He} + e^- + \bar{\nu}_e$, $n \to p + e^- + \bar{\nu}_e$, and ${}^7\mathrm{Be} + e^- \to {}^7\mathrm{Li} + \nu_e$, which only happen/conclude at a later cosmological time. This way, the user can in principle implement additional effects into their analysis that happen between the end of photodisintegration and the decay times. This implies that the abundances today $Y_N^* \neq Y_N$ are different than the ones returned by `ACROPOLIS` and in general it is necessary to perform the following three operations

Process	Abundance today		
$n \to p + e^- + \bar{\nu}_e$	$Y_n^* = 0$		$Y_p^* = Y_p + Y_n$
${}^3\mathrm{H} \to {}^3\mathrm{He} + e^- + \bar{\nu}_e$	$Y_{{}^3\mathrm{H}}^* = 0$		$Y_{{}^3\mathrm{He}}^* = Y_{{}^3\mathrm{He}} + Y_{{}^3\mathrm{H}}$
${}^7\mathrm{Be} + e^- \to {}^7\mathrm{Li} + \nu_e$	$Y_{{}^7\mathrm{Be}}^* = 0$		$Y_{{}^7\mathrm{Li}}^* = Y_{{}^7\mathrm{Be}} + Y_{{}^7\mathrm{Li}}$

---

[4]This can easily be changed, meaning that it is possible to use arbitrary sets of initial abundances for the calculation. We will discuss this later in section 6.2.

In fact, the decay of tritium might become relevant for photodisintegration reactions that happen around or beyond $\tau_{^3\text{H}} = 3.89 \times 10^8\,\text{s}$. However, this is at most a percent effect, since the abundance of tritium is much smaller than the one of helium-3.

5.2.2 A first look at the source code

While the output of the wrapper scripts is sufficient to check whether a certain parameter point is excluded or not, it is sometimes important to not only print but also to further process the final abundances. This step, however, requires some knowledge regarding the internal workings of ACROPOLIS, which can be obtained by taking a closer look at the actual source code. Using the file `decay` as an example, much of the important information is comprised in the following two lines of code:

```python
#! /usr/bin/env python3

[...]

from acropolis.models import DecayModel

[...]

Yf = DecayModel(mphi, tau, temp0, n0a, bree, braa).run_disintegration()

[...]
```

In this code snippet, the relevant model is first loaded from the module `acropolis.models`, which includes implementations for both example models. More precisely, the scenarios from section 4.1 (`decay`) and section 4.2 (`annihilation`) are implemented in the classes `acropolis.models.DecayModel` and `acropolis.models.AnnihilationModel`, respectively. After loading the model, a new instance of the respective class is initialised with the appropriate input parameters – which here are obtained by parsing the command-line arguments –, and the calculation is initiated by calling the method `run_disintegration()`. The latter function returns a `numpy.ndarray` of dimension $9 \times n$ containing the final abundances, with n being the number of different sets of initial conditions, i.e. $n = 3$ by default. Finally, the resulting array `Yf` can either be pretty-printed (like in the wrapper scripts) or it can be used as an input for further calculations. In fact, these two lines of code are everything it takes to calculate the abundances after photodisintegration for a given model, regardless of whether the model ships with ACROPOLIS or was implemented by the user.

5.2.3 Running parameter scans

ACROPOLIS provides two classes for parameter scans, `acropolis.scans.BufferedScanner` and `acropolis.scans.ScanParameter`. In order to understand how these two classes work we show the following code snippet, which performs a parameter scan for the model from sec 4.1 with $m_\phi = 50\,\text{MeV}$, $T_0 = 10\,\text{MeV}$, $\text{BR}_{ee} = 0$, and $\text{BR}_{\gamma\gamma} = 1$ in the $\tau_\phi - n_\phi/n_\gamma|_{T_0}$ parameter plane.

```
#! /usr/bin/env python3

[...]

from acropolis.models import DecayModel
from acropolis.scans  import ScanParameter, BufferedScanner

[...]

res = BufferedScanner( DecayModel,
                       mphi  = 50.,
                       tau   = ScanParameter(3, 10, 200),
                       temp0 = 10.,
                       n0a   = ScanParameter(-14, -3, 200, fast=True),
                       bree  = 0.,
                       braa  = 1.
                     ).perform_scan()

[...]
```

Here, the previously mentioned classes are first loaded from the module `acropolis.scans` together with the model that is used for the scan, in this case `DecayModel`. Then a new instance of `BufferedScanner` is initiated, which takes as a first argument the model that is used for the calculation and afterwards a set of keyword arguments with names that are identical to the ones in the constructor of the model. These keyword arguments can either be a `float` or an instance of `ScanParameter`. While the `float` parameters are kept constant, instances of `ScanParameter` are scanned over. The range for the scan is defined by the arguments of `ScanParameter(ivalue, fvalue, num, spacing)` and is constructed internally using `NumPy` functions by either calling `np.logspace(ivalue, fvalue, num)` for `spacing="log"` (default) or `np.linspace(ivalue, fvalue, num)` for `spacing="lin"`. Hence, the above code performs a scan over $\tau_\phi \in [10^3\,\mathrm{s}, 10^{10}\,\mathrm{s}]$ and $n_\phi/n_\gamma|_{T_0} \in [10^{-14}, 10^{-3}]$ with 200 data points in each direction, distributed equidistantly on a log-scale. There is one additional argument that can be passed to `ScanParameter`, `fast`, which can either be `True` or `False`. To understand the importance of this parameter, let us note that according to eq. (2.7), the spectra $\kappa F_x(E)$ with some constant κ are a solution of the cascade equation with source terms $\kappa S_x(E)$ if the spectra $F_x(E)$ are valid solutions of the cascade equation with source terms $S_x(E)$. In these two cases, the corresponding matrices in eq. (3.9) are then $\kappa \mathcal{R}(T)$ and $\mathcal{R}(T)$, respectively. Hence, for each parameter that simply scales the source terms, it is not necesary to recalculate the non-thermal spectra and nuclear rates for each parameter point. Instead, we can simply rescale previously obtained solutions. The argument `fast` is used to determine whether this procedure is used or not. In the example above with `fast=True` for `n0a` this means that for a given value of τ_ϕ, $\mathcal{R}(T)$ is calculated only once for $n_\phi/n_\gamma|_{T_0} = 10^{-14}$ and then simply rescaled for other values of $n_\phi/n_\gamma|_{T_0}$. This way, the initial calculation for each `tau` takes $\sim \mathcal{O}(1\,\mathrm{min})$, while the calculation of different points for `n0a` then merely takes $\sim \mathcal{O}(1\,\mu\mathrm{s})$. The following tables show which parameters

can be used with `fast=True` without spoiling the calculation (i.e. those parameters that only enter the source terms as a prefactor and do not appear elsewhere in the calculation):

<div align="center">DecayModel</div>

Parameter	fast=True
mphi	No
tau	No
temp0	No
n0a	Yes
bree	No
braa	No

<div align="center">AnnihilationModel</div>

Parameter	fast=True
mchi	No
a	Yes (if b=0)
b	Yes (if a=0)
tempkd	No
bree	No
braa	No

Finally, once the instance of `BufferedScanner` is created, the scan is initiated via a call to the method `perform_scan()`. This function performs the scan on several cores, the number of which can be specified via the optional argument `cores`. If the latter is not specified (set to -1), only one (all available) cores are used. The given function then returns the array `res`, which contains one line for each parameter combination that was used in the scan. Each line is hence composed of the current parameter combination (two columns) and the corresponding final abundances (nine columns for each set of initial conditions).

To illustrate the performance of this framework, we performed several scans in both models, the results of which are shown in figures 1 and 2. In addition to the overall 95% C.L. limit (black) we also indicate the parts of parameter space that are excluded due to an underproduction of helium-4 (blue), over- or underproduction of deuterium (orange/grey), and overproduction of helium-3 relative to deuterium (green).[5]

In figure 1 (left) we show the constraints for fixed $m_\phi = 50\,\mathrm{MeV}$ in the $\tau_\phi - (n_\phi/n_\gamma)|_{T=T_0}$ plane with $T_0 = 10\,\mathrm{MeV}$ and decays into two photons. The limits start around $\tau_\phi \sim 10^4\,\mathrm{s}$, quickly become very stringent with increasing lifetime, and eventually flatten out excluding a number density far below the photon number density at $T_0 = 10\,\mathrm{MeV}$. Note that due to this high sensitivity we are able to assume a standard cosmological history apart from photodisintegration as detailed in section 4.1. In the right panel of figure 1 we show the constraints for fixed $\tau_\phi = 10^7\,\mathrm{s}$ (right) in the $m_\phi - (n_\phi/n_\gamma)|_{T=T_0}$ plane with $T_0 = 10\,\mathrm{MeV}$ and decays into two photons. The limits start at twice the disintegration threshold for deuterium, $m_\phi = 2E_\mathrm{D}^\mathrm{th} \approx 4.4\,\mathrm{MeV}$. Apart from some regions with more complex structure due to different disintegration reactions the limits become increasingly strong with larger m_ϕ as the energy density injected into the SM becomes larger. For masses above the pion threshold in particular, $m_\phi \gtrsim 280\,\mathrm{MeV}$, hadrodisintegration may become relevant if ϕ has non-vanishing couplings to quarks, implying that $\mathrm{BR}_{\gamma\gamma} + \mathrm{BR}_{e^+e^-} < 1$ in general. Also muons are kinematically available in the mass region (which are currently not implemented in `ACROPOLIS`).

---

[5]Here we adopt the latest recommendations for the observed abundances of $\mathcal{Y}_\mathrm{p} = (2.45 \pm 0.03) \times 10^{-1}$ and $\mathrm{D}/^1\mathrm{H} = (2.547 \pm 0.035) \times 10^{-5}$ from [40], where we took into account the uncertainty due to the baryon-to-photon ratio from Planck for $\mathrm{D}/^1\mathrm{H}$, cf. [27, 39], as well as $^3\mathrm{He}/\mathrm{D} = (8.3 \pm 1.5) \times 10^{-1}$ as an upper limit from [41]. The nuclear rate uncertainties are taken into account as detailed in [24].
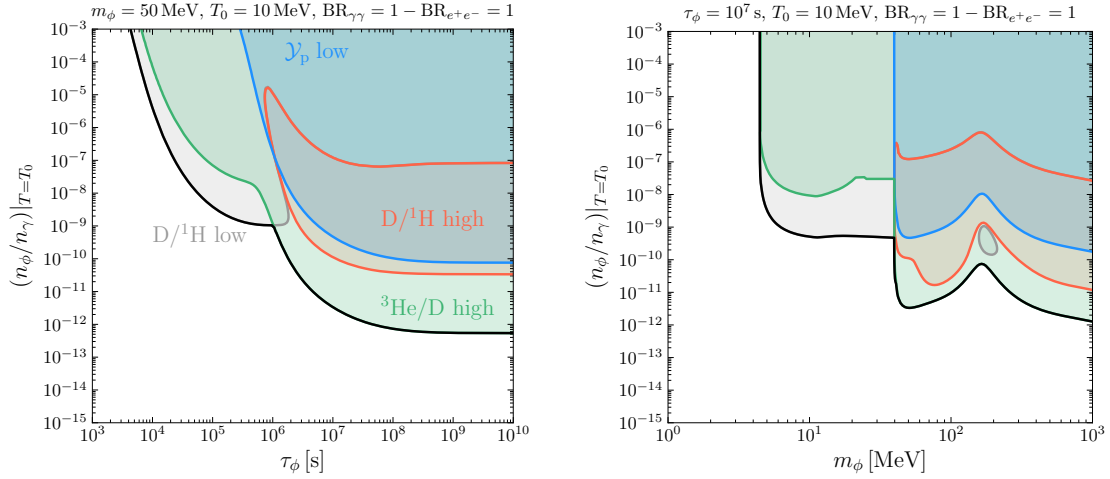
Figure 1. 95% C.L. constraints for decay of a decoupled MeV-scale BSM particle (implemented in `DecayModel`) into two photons ($\mathrm{BR}_{\gamma\gamma} = 1 - \mathrm{BR}_{e^+e^-} = 1$) in $\tau_\phi - (n_\phi/n_\gamma)|_{T=T_0}$ plane (left) and $m_\phi - (n_\phi/n_\gamma)|_{T=T_0}$ plane (right) with $T_0 = 10\,\mathrm{MeV}$ and $m_\phi = 50\,\mathrm{MeV}$ (left) as well as $\tau_\phi = 10^7\,\mathrm{s}$ (right). The limits from individual observables are shown separately: primordial deuterium abundance (orange high, grey low), helium-4 mass fraction $\mathcal{Y}_\mathrm{p}$ (blue), and helium-3 abundance normalised by deuterium (green). The overall 95% C.L. BBN limit is given by the black full line as an envelope of individual 95% C.L. constraints neglecting correlations. Using $(n_\phi/n_\gamma)|_{T=T_0}$, i.e. `n0a`, as a `fast` parameter on a single computing node with two `AMD EPYC 7402 24-Core Processors` the scans took ~ 25 min (left) and ~ 35 min (right) for a 200×200 grid.

The scans for figure 1 took ~ 25 min (left) and ~ 35 min (right) for a 200×200 grid on two `AMD EPYC 7402 24-Core Processors`, clearly highlighting the performance improvement due to the `fast` parameter $(n_\phi/n_\gamma)|_{T=T_0}$, i.e. `n0a`, making the number of points in this direction computationally inexpensive (cf. also appendix B). The runtime is thus determined mostly by the number of points in the direction of τ_ϕ or m_ϕ (not `fast`).

In figure 2 we show the constraints for residual annihilations of DM into electron-positron pairs ($\mathrm{BR}_{\gamma\gamma} = 1 - \mathrm{BR}_{e^+e^-} = 0$) for purely s-wave annihilations (left, $b = 0$) and purely p-wave annihilations (right, $a = 0$, $T_\mathrm{kd} = 1\,\mathrm{MeV}$) as implemented in `AnnihilationModel`. These limits start at the disintegration threshold of deuterium, $m_\chi = E_\mathrm{D}^\mathrm{th} \approx 2.2\,\mathrm{MeV}$, and closely resemble those presented in [26], albeit with updated observationally inferred primordial abundances. We therefore refer to [26] for a detailed discussion. The scans took ~ 85 min each for a 200×200 grid on the aforementioned computing node.

6 Implementing your own models

6.1 The model framework `acropolis.models`

While the provided example models should suffice to tackle most problems of interest, it may sometimes still happen that a scenario cannot directly be mapped to the standard
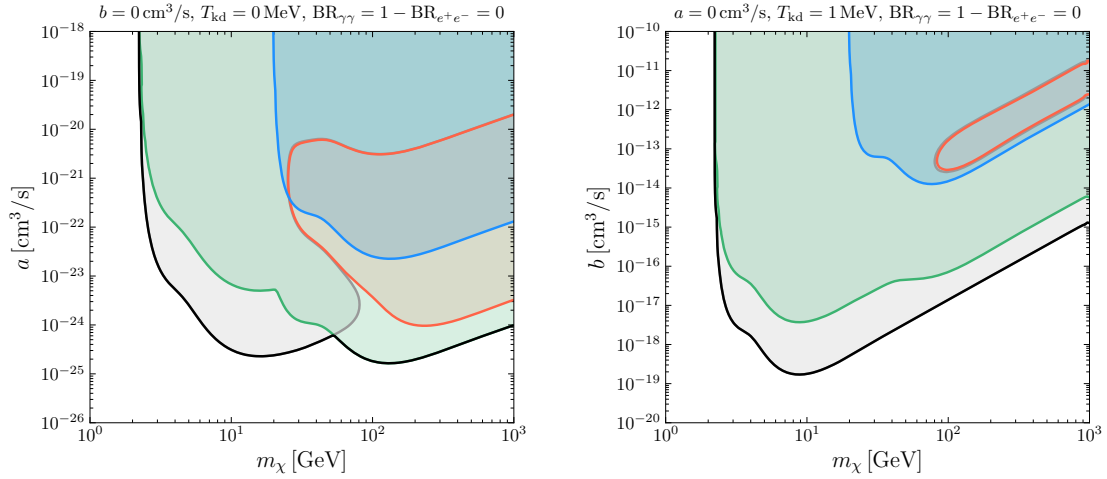
Figure 2. 95% C.L. constraints for residual annihilations of DM (implemented in `AnnihilationModel`) into electron-positron pairs ($BR_{\gamma\gamma} = 1 - BR_{e^+e^-} = 0$) for purely s-wave annihilations (left, $b = 0$) and purely p-wave annihilations (right, $a = 0$, $T_{kd} = 1\,\mathrm{MeV}$). For the explanation of the colour-coding see figure 1. Using $(n_\phi/n_\gamma)|_{T=T_0}$, i.e. `n0a`, as a `fast` parameter on a single computing node with two `AMD EPYC 7402 24-Core Processors` the scans took ~ 85 min each for a 200×200 grid.

implementation in `ACROPOLIS`. For such cases, `ACROPOLIS` provides further tools that allow for an easy implementation of additional models. The most important class in this context is `acropolis.models.AbstractModel`, which is an abstract base class containing most of the low-level implementation needed to run its method `run_disintegration()`. In fact, using this class as a base, any new model can be implemented in only two steps:

(i) create a new class, say `NewModel`, that uses `AbstractModel` as a base class, and

(ii) implement all abstract methods that are provided by `AbstractModel`, i.e.[6]

- `AbstractModel._temperature_range`
- `AbstractModel._source_photon_0`
- `AbstractModel._source_electron_0`
- `AbstractModel._source_photon_c`

Consequently, each new model class must therefore feature the scaffold (compare this e.g. to the implementations of `DecayModel` and `AnnihilationModel`)

---

[6]The functions `AbstractModel._source_positron_0/c` also exist, though they are not abstract and, by default, simply return the output of `AbstractModel._source_electron_0/c`, which is justified for most scenarios. However, if your specific scenario predicts different source terms for electrons and positrons, it is always possible to simply overwrite the former function. To reduce overhead, the method `AbstractModel._source_electron_c` is also not abstract and returns 0 by default.

```
from acropolis.models import AbstractModel

class NewModel(AbstractModel):

    def _temperature_range(self):
        [...]

    def _source_photon_0(self, T):
        [...]

    def _source_electron_0(self, T):
        [...]

    def _source_photon_c(self, E, T):
        [...]
```

In the following sections, we will further discuss how to implement these four abstract methods. Apart from these methods, the constructor of `AbstractModel`, which is to be called in the constructor of `NewModel`, also needs the injection energy E_0 for the monochromatic part of the source term (cf. eq. (2.5)) and an instance of the class `InputInterface` containing the necessary input data. This will be detailed in section 6.1.3.

6.1.1 The functions for the source terms

The functions `_source_photon_0` and `_source_electron_0` are associated with the respective source terms $S_\gamma^{(0)}(T)$ and $S_{e^-}^{(0)}(T)$ that enter in eq. (2.5). Both of these functions take as their only argument the temperature T [in MeV] and are expected to return the corresponding source term [in $1/\text{MeV}^2$]. As an example, let us take a look at `DecayModel`, which implements the source terms in eqs. (4.2) and (4.3) via

```
def _source_photon_0(self, T):
    return self._sBRaa * 2. * self._number_density(T) * hbar/self._sTau

def _source_electron_0(self, T):
    return self._sBRee * self._number_density(T) * hbar/self._sTau
```

Here, `self._sBRee` and `self._sTau` are two of the input parameters of the model – which are set in the constructor (cf. section 6.1.3) – and `hbar` is a constant that has been imported from `acropolis.params`. We will take a closer look at the available parameters in section 6.3. Finally, the function `self._number_density(T)` implements $n_\phi(T)$ from eq. (4.6) and is exclusive to `DecayModel`.[7]

Similarly, the function `_source_photon_c` is associated with the continuous source term $S_\gamma^{(c)}(E, T)$ entering eq. (2.5). This function takes two arguments, the energy E and the temperature T [both in MeV], and is expected to return the corresponding source term [in

---

[7]While it is only necessary to implement the four abstract methods of `AbstractModel`, it is of course also possible to provide other (private) methods that are needed within the new model.

$1/\text{MeV}^3$]. As an example, we show the corresponding implementation in `DecayModel` of the source term from eq. (4.4), which encodes final-state radiation of photons:

```python
def _source_photon_c(self, E, T):
    EX = self._sE0

    x = E/EX
    y = me2/(4.*EX**2.)

    if 1. - y < x:
        return 0.

    _sp = self._source_electron(T)

    return (_sp/EX) * (alpha/pi) * (1.+(1.-x)**2.)/x * log((1.-x)/y)
```

Here, `self._sE0` is the injection energy – which is set in the constructor – and `me2`, `alpha`, and `pi` are constants that have been imported from `acropolis.params`.

6.1.2 The function for the temperature range

Additionally to the source terms from the previous section, the only other function that needs to be implemented is `_temperature_range`, which is expected to return a two-dimensional list with the minimal and the maximal temperature spanning the range needed in eq. (3.8). Here, it is important to ensure that this range covers all temperatures for which photodisintegration is actually relevant. Coming back to `DecayModel` as an example, photodisintegration happens around the lifetime of the particle at $t \sim \tau_\phi$, and a suitable temperature range is $[10^{-3/2}T(\tau_\phi), 10^{1/2}T(\tau_\phi)]$, since the bulk of photodisintegration reactions happens only for $t > \tau_\phi$. The actual implementation in `DecayModel` is given by

```python
def _temperature_range(self):
    # The number of degrees-of-freedom to span
    mag = 2.
    # Calculate the approximate decay temperature
    Td = self._sII.temperature( self._sTau )
    # Calculate Tmin and Tmax from Td
    Td_ofm = log10(Td)
    # Here we choose -1.5 (+0.5) orders of magnitude
    # below (above) the approx. decay temperature,
    # since the main part happens after t = \tau
    Tmin = 10.**(Td_ofm - 3.*mag/4.)
    Tmax = 10.**(Td_ofm + 1.*mag/4.)

    return (Tmin, Tmax)
```

Apart from the previously discussed model parameter `self._sTau`, this implementation only involves the variable `self._sII`, which is set in the constructor (cf. section 6.1.3) and constitutes an instance of the class `acropolis.input.InputInterface`. This class is the second most important class next to `acropolis.models.AbstractModels` as it provides

an interface for all the files that are used as an input for the calculation. For example, this class wraps functions like $T(t)$ (i.e. `InputInterface.temperature` from above), which are relevant for eqs. like (2.8) and therefore need to be provided as an input. In section 6.2 we will discuss this class in more detail and also go over the different inputs that are required for a successful calculation.

6.1.3 The model constructor

When calling the constructor of a new model, it crucial to also invoke the constructor of the abstract base class `AbstractModel`. The latter one takes two arguments, the first one being the injection energy E_0 for the monochromatic part of the source term (cf. eq. (2.5)) and the second one being an instance of the previously mentioned class `InputInterface`. When considering `DecayModel` as an example, which features an injection energy $E_0 = m_\phi/2$, this leads to a constructor of the following form

```python
def __init__(self, mphi, tau, temp0, n0a, bree, braa):
    # Initialize the Input_Interface
    self._sII   = InputInterface("data/sm.tar.gz")

    [...]

    # The injection energy
    self._sE0   = mphi/2.

    [...]

    # Call the super constructor
    super(DecayModel, self).__init__(self._sE0, self._sII)
```

Here, an instance of `InputInterface` is constructed from the data that is stored in the file `data/sm.tar.gz`. To fully understand the code, we therefore have to discuss the importance of this class, which we do in the next section.

6.2 The input framework `acropolis.input`

The calculation of the abundances after photodisintegration cannot proceed without certain (model-dependent) inputs, including the baryon-to-photon ratio, the initial abundances after BBN, and the dynamics of the background plasma as encoded in functions such as $T(t)$, $T_\nu(T)$, and $H(T)$. All of these inputs are collectively wrapped by and accessible via the previously mentioned class `acropolis.input.InputInterface`. The constructor of this class only takes a single argument, which is expected to be the name of a `.tar.gz` file containing the following three files

```
input.tar.gz
|
|___ param_file.dat
|___ cosmo_file.dat
|___ abundance_file.dat
```

One such file that ships with `ACROPOLIS` is `data/sm.tar.gz`, which assumes a standard cosmological history without an appreciable impact of BSM physics apart from photodisintegration. As mentioned above this is a very good approximation for parameter regions which are close to the resulting limit, as photodisintegration strongly constrains even very small abundances, implying that changes to the SM values are small. Correspondingly, this file is used for both example models, as in these cases the abundance of the participating dark-sector particles gives a negligible contribution. However, this might not always be the case, potentially necessitating the construction of model-specific input files. We go over these different `*.dat` files and their content below and discuss how to access them from an instance of `InputInterface`. This is crucial since this class might encode vital information that is needed to implement the different source terms and the temperature range.

6.2.1 The file *param_file.dat*

This file contains all input parameters that cannot be provided in the model constructor, e.g. if they are closely tied to the evolution of the background plasma. Here, the minimal requirement is to provide a value for the baryon-to-photon ratio,[8] but more parameters can be incorporated via additional lines of the form `key=value`. For example, the file `data/sm.tar.gz:param_file.dat` only contains one line,

eta=6.137e−10

However, independent of the number of lines in this file, after constructing an instance of `InputInterface`, all included parameters can be accessed by simply calling the method `InputInterface.parameter(key)` with the corresponding `key`. For example in the case of `sm.tar.gz`, calling `parameter("eta")` would return `6.137e-10`.

6.2.2 The file *cosmo_file.dat*

This file encodes the cosmological evolution of the background plasma. It must contain at least five columns (separated by spaces) including discrete grids for the following quantities:

- time t [in s]

- temperature T [in MeV]

- time-temperature relation $\mathrm{d}T/\mathrm{d}t$ [in MeV^2]

- neutrino temperature T_ν [in MeV]

- Hubble rate H [in MeV]

The grids need to be equidistantly spaced on a log-scale of time t. Given an instance of `InputInterface`, the interpolated data of these mandatory entries is accessible via the predefined methods

---

[8]This parameter also enters the initial abundances and therefore is closely tied to other inputs.

$$T(t) \quad \texttt{InputInterface.temperature(t)}$$

$$t(T) \quad \texttt{InputInterface.time(T)}$$

$$\tfrac{\mathrm{d}T}{\mathrm{d}t}(T) \quad \texttt{InputInterface.dTdt(T)}$$

$$T_\nu(T) \quad \texttt{InputInterface.neutrino\_temperature(T)}$$

$$H(T) \quad \texttt{InputInterface.hubble\_rate(T)}$$

$$R(T) \quad \texttt{InputInterface.scale\_factor(T)}$$

Besides these mandatory columns, it is also possible to add an arbitrary number of additional ones, e.g. containing quantities that are needed to implement the source terms or the temperature range. The interpolated data of the columns can then be accessed by calling the more generic method `InputInterface.cosmo_column(yc, val, xc=1)`. This function can be used to generically calculate $y(x)$ at $x =$`val` via interpolation of any two columns in *cosmo_file.dat*, where y is determined from the entries in column `yc` and x is determined from the entries in column `xc`. This function therefore does not only allow to evaluate user-defined columns but also different correlations between the mandatory quantities, like e.g. $t(H)$. Possible examples include

| yc	xc	$y(x)$
1	0	$T(t)$
4	3	$H(T_\nu)$
0	4	$t(H)$
5	1	$C_5(T)$

In the last example, column `yc=5` is assumed to exist (i.e. that it was added by the user), in which case C_5 denotes the cosmological quantity that is tabulated in this column.

6.2.3 The file *abundance_file.dat*

This file contains the initial abundances (before photodisintegration, but after nucleosynthesis) that are used for the calculation. This file must contain at least one column with the abundances for n, p, ^{2}H, ^{3}H, ^{3}He, ^{4}He, ^{6}Li, ^{7}Li, and ^{7}Be. Additional columns are also allowed and – as previously mentioned – the code calculates the resulting abundances after photodisintegration for each set of initial values (i.e. for each column). After constructing an instance of `InputInterface`, the different initial abundances can be collectively accessed by calling the method `InputInterface.bbn_abundances()`, which returns a $9 \times n$ array with n being the number of columns. Alternatively, the first column can also be separately obtained by calling `InputInterface.bbn_abundance_0()` instead.

6.3 The parameters in `acropolis.params`

All of the constants that are used within `ACROPOLIS` can be found in `acropolis.params`. This files not only contains mathematical (`zeta3`,...) and physical (`alpha, me, hbar`,...) constants, but also some algorithm-specific parameters such as the number of points per
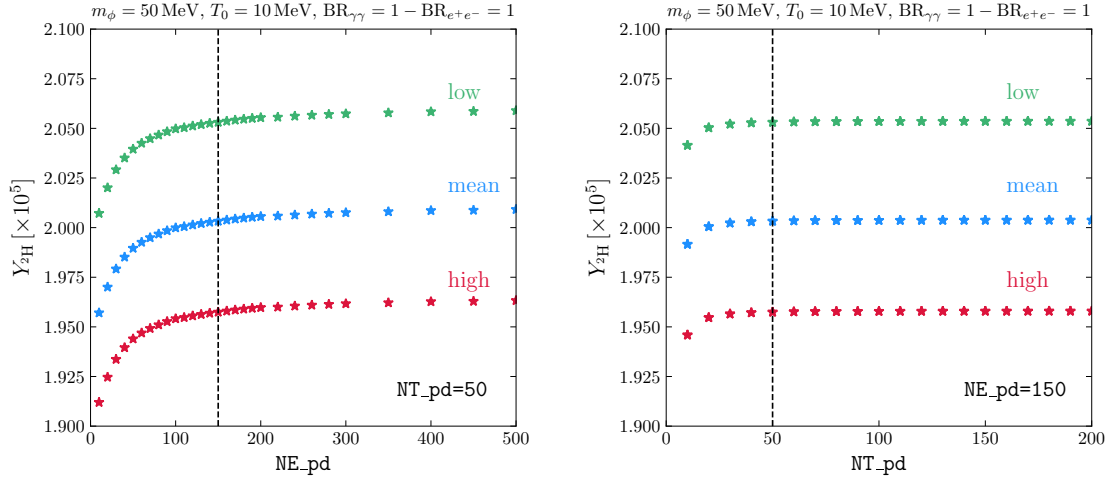
Figure 3. Convergence of the abundance of deuterium as a function of the grid points `NE_pd` and `NT_pd`. The dashed line indicates the default value in `ACROPOLIS`.

decade used to construct the energy (`NE_pd`) and temperature (`NT_pd`) grids. All of these parameters are well documented in the file itself. There are three parameters at the beginning of this file that can be set to `True` or `False`:

- `usedb` (default: `True`) – if this parameter is set to `True`, the optional database files are used to speed up the calculation at the cost of additional I/O operations. If this parameter is set to `False` instead, all reaction rates are calculated from scratch. This might be especially useful when running on a batch system. Some benchmark that have been calculated with and without the use of the database files can be found in appendix B.

- `verbose` (default: `True`) – if this parameter is set to `True`, the code prints all message types to the screen, including `INFO`, `WARNING`, and `ERROR`. If this parameter is set to `False` instead, `INFO` messages are not printed.

- `debug` (default: `False`) – if this parameter is set to `True`, additional debug info is printed, including additional information on the position at which certain warnings or errors appear.

For all other parameters, it is not advised to perform any manual changes, since these were selected in a way to ensure a great compromise between runtime and accuracy around the exclusion region.[9] We demonstrate this in figure 3, where we show the final abundance of deuterium for different values of `NE_pd` and `NT_pd`. We find that the abundances indeed converge for `NE_pd`,`NT_pd`$\to \infty$, while the default values for `NE_pd` and `NT_pd` (indicated by the dashed lines) lead to a result that deviates only at the 0.1% level – much smaller than the difference that is caused by the reaction rate uncertainties (different colours).

---

[9]Far away from the exclusion line, the results might not be perfectly accurate. If, for some reason, you also require precise results in this region, try increasing the values of `NE_pd` and `NT_pd`.

7 Conclusions

In this work, we present ACROPOLIS, A generiC fRamework fOr Photodisintegration Of LIght elementS, the first public code for calculating the effects of photodisintegration of light elements in the early universe. ACROPOLIS performs this by first computing the non-thermal part of the photon spectrum arising due to late-time high-energetic injections into the SM plasma and then calculating its effect on the primordial light element abundances. We provide and discuss two example programs covering a plethora of interesting applications via *(i)* the decay of a decoupled MeV-scale BSM particle and *(ii)* residual annihilations of DM. Furthermore, we detail how additional models can easily be implemented in the modular structure of ACROPOLIS.

Acknowledgments

A Rates for the cascade processes

In this appendix, we collect for completeness all relevant total and differential interaction rates $\Gamma_x(E)$ and $K_{x' \to x}(E, E')$ for the cascade processes of high-energetic photons, electrons, and positrons on the background photons, electrons, and nuclei (see eqs. (2.4) and (2.7)). Large parts are directly taken from [24].

Target densities

The thermal photon spectrum differential in energy $f_\gamma(\bar{\epsilon})$ is given by

$$f_\gamma(\bar{\epsilon}) = \frac{\bar{\epsilon}^2}{\pi^2} \times \frac{1}{\exp(\bar{\epsilon}/T) - 1} \, , \tag{A.1}$$

while the total baryon number density can be calculated from the baryon-to-photon ratio η and the number density of photons $n_\gamma(T)$,

$$n_b(T) = \eta \times n_\gamma(T) = \eta \times \frac{2\zeta(3)}{\pi^2} T^3 \, . \tag{A.2}$$

Via charge neutrality we obtain for the number density of background electrons

$$n_e(T) = \sum_N Z_N n_N \simeq \left[Y_p(T) + 2Y_{^4\text{He}}(T) \right] \times n_b(T), \quad Y_N(T) = \frac{n_N(T)}{n_b(T)} \, . \tag{A.3}$$

At the times relevant to photodisintegration ($t \gtrsim 10^4$ s), BBN has already terminated and the nuclear abundances $Y_N(T)$ are approximately constant. Hence, in the following, we neglect the temperature dependence of $Y_N(T)$, and fix them to their values directly after BBN. Note that a change due to photodisintegration is only relevant when the corresponding scenario is excluded anyhow.

Final-state radiation: $DS \to e^+e^-\gamma$

Following [25], the source term for final-state radiation can directly be calculated from the source term of electrons or positrons via the expression [35, 36]

$$S_\gamma^{(\mathrm{FSR})}(E) = \frac{S_{e^\pm}^{(0)}}{E_0} \times \frac{\alpha}{\pi} \frac{1 + (1-x)^2}{x} \ln\left(\frac{4E_0^2(1-x)}{m_e^2}\right) \times \Theta\left(1 - \frac{m_e^2}{4E_0^2} - x\right) \tag{A.4}$$

with $x = E/E_0$.

Double photon pair creation: $\gamma\gamma_{\mathrm{th}} \to e^+e^-$

The rate for double photon pair creation is given by [31][10]

$$\Gamma_\gamma^{(\mathrm{DP})}(E) = \frac{1}{8E^2} \times \int_{m_e^2/E}^\infty d\bar\epsilon \, \frac{\mathrm{f}_\gamma(\bar\epsilon)}{\bar\epsilon^2} \times \int_{4m_e^2}^{4E\bar\epsilon} ds \, s \cdot \sigma_{\mathrm{DP}}\left(\beta = \sqrt{1 - 4m_e^2/s}\right) \tag{A.5}$$

with the total cross-section

$$\sigma_{\mathrm{DP}}(\beta) = \frac{\pi\alpha^2}{2m_e^2} \times (1 - \beta^2)\left[(3 - \beta^4)\ln\left(\frac{1+\beta}{1-\beta}\right) - 2\beta\left(2 - \beta^2\right)\right] . \tag{A.6}$$

This process is only relevant above the threshold of production of electron-positron pairs $E \gtrsim m_e^2/(22T)$, allowing us to set $\Gamma_\gamma^{(\mathrm{DP})}(E) = 0$ for $E < m_e^2/(22T)$.

The differential rate for double photon pair creation entering the calculation of the electron and positron spectrum[11] was originally calculated in [42] and is given by[12]

$$K_{\gamma \to e^\pm}^{(\mathrm{DP})}(E, E') = \frac{\pi\alpha^2 m_e^2}{4} \times \frac{1}{E'^3} \int_{m_e^2/E'}^\infty d\bar\epsilon \, \frac{\mathrm{f}_\gamma(\bar\epsilon)}{\bar\epsilon^2} \, G(E, E', \bar\epsilon) , \tag{A.7}$$

with

$$G(E, E', \bar\epsilon) = \frac{4(E' + \bar\epsilon)^2}{E(E' + \bar\epsilon - E)} \ln\left(\frac{4\bar\epsilon E(E' + \bar\epsilon - E)}{m_e^2(E' + \bar\epsilon)}\right)$$

$$+ \left(\frac{m_e^2}{\bar\epsilon(E' + \bar\epsilon)} - 1\right) \frac{(E' + \bar\epsilon)^4}{E^2(E' + \bar\epsilon - E)^2}$$

$$+ \frac{2\left[2\bar\epsilon(E' + \bar\epsilon) - m_e^2\right](E' + \bar\epsilon)^2}{m_e^2 E(E' + \bar\epsilon - E)} - 8\frac{\bar\epsilon(E' + \bar\epsilon)}{m_e^2} \tag{A.8}$$

for $m_e < E_{\mathrm{lim}}^- < E < E_{\mathrm{lim}}^+$,

$$2E_{\mathrm{lim}}^\pm = E' + \bar\epsilon \pm (E' - \bar\epsilon)\sqrt{1 - \frac{m_e^2}{E'\bar\epsilon}} , \tag{A.9}$$

and $G(E, E', \bar\epsilon) = 0$ otherwise. As explained above, we further set $K_{\gamma \to e^\pm}^{(\mathrm{DP})}(E, E') = 0$ for $E' < m_e^2/(22T)$.

---

[10]Correcting a typo in eq. (27) of [31].

[11]Here, the notation $\gamma \to e^\pm$ in the index of $K_{x' \to x}$ indicates that the corresponding expression is valid for $x' \to x \in \{\gamma \to e^+, \gamma \to e^-\}$ and consequently enters eq. (2.7) twice.

[12]Correcting a typo in eq. (28) of [31].

Photon-photon scattering: $\gamma\gamma_{\rm th} \to \gamma\gamma$

The total and differential interaction rates for photon-photon scattering have been originally calculated in [43], and are given by[13]

$$\Gamma_\gamma^{(\rm PP)}(E) = \frac{1946}{50625\pi} \times \frac{8\pi^4}{63} \times \alpha^4 m_e \times \left(\frac{E}{m_e}\right)^3 \left(\frac{T}{m_e}\right)^6 , \tag{A.10}$$

and

$$K_{\gamma\to\gamma}^{(\rm PP)}(E, E') = \frac{1112}{10125\pi} \times \frac{\alpha^4}{m_e^8} \times \frac{8\pi^4 T^6}{63} \times E'^2 \left[1 - \frac{E}{E'} + \left(\frac{E}{E'}\right)^2\right]^2 . \tag{A.11}$$

In principle, these expressions are only valid for $E \lesssim m_e^2/T$ [31]. However, for energies larger than this, photon-photon scattering is in any case negligible compared to double photon pair creation, making it unnecessary to impose this additional constraint.

Bethe-Heitler pair creation: $\gamma N \to N e^+ e^-$

The total rate for Bethe-Heitler pair creation at energies $E \geq 4m_e$ and up to order m_e^2/E^2 can be written as [31, 44][14]

$$\Gamma_\gamma^{(\rm BH)}(E) \simeq \frac{\alpha^3}{m_e^2} \times \left(\sum_N Z_N^2 n_N(T)\right) \times \left(\left[\frac{28}{9}\ln(2k) - \frac{218}{27}\right]\right.$$
$$\left. + \left(\frac{2}{k}\right)^2 \left[\frac{2}{3}\ln(2k)^3 - \ln(2k)^2 + \left(6 - \frac{\pi^2}{3}\right)\ln(2k) + 2\zeta(3) + \frac{\pi^2}{6} - \frac{7}{2}\right]\right)\Bigg|_{k=E/m_e} \tag{A.12}$$

Here, we only take into account scattering off ^{1}H and ^{4}He, which implies

$$\sum_N Z_N^2 n_N(T) \simeq \sum_{N \in \{p, {}^4{\rm He}\}} Z_N^2 n_N(T) = \left[Y_p(T) + 4Y_{{}^4{\rm He}}(T)\right] \times n_b(T) , \tag{A.13}$$

since the abundances of all other nuclei are strongly suppressed. Furthermore, for energies in the range $2m_e < E \leq 4\,{\rm MeV}$, the interaction rate is essentially constant [32], $\Gamma_\gamma^{(\rm BH)}(E) \simeq \Gamma_\gamma^{(\rm BH)}(E = 4\,{\rm MeV})$.

The differential rate for Bethe-Heitler pair creation is given by [31, 45]

$$K_{\gamma\to e^\pm}^{(\rm BH)}(E, E') = \left(\sum_N Z_N^2 n_N(T)\right) \times \frac{{\rm d}\sigma_{\rm BH}(E, E')}{{\rm d}E} \times \Theta(E' - E - m_e) , \tag{A.14}$$

---

[13]Correcting a typo in eq. (31) of [31] and in eq. (5) of [29].
[14]We checked that higher order terms do not change the final results.

with the differential cross-section

$$\frac{\mathrm{d}\sigma_{\mathrm{BH}}(E, E')}{\mathrm{d}E} = \frac{\alpha^3}{m_e^2} \times \left(\frac{p_+ p_-}{E'^3}\right) \times \left[-\frac{4}{3} - 2E_+ E_- \frac{p_+^2 + p_-^2}{p_+^2 p_-^2} \right.$$

$$+ m_e^2 \left(l_- \frac{E_+}{p_-^3} + l_+ \frac{E_-}{p_+^3} - \frac{l_+ l_-}{p_+ p_-} \right)$$

$$+ L \left(-\frac{8E_+ E_-}{3p_+ p_-} + \frac{E'^2}{p_+^3 p_-^3} \left(E_+^2 E_-^2 + p_+^2 p_-^2 - m_e^2 E_+ E_- \right) \right)$$

$$\left. - L \frac{m_e^2 E'}{2p_+ p_-} \left(l_+ \frac{E_+ E_- - p_+^2}{p_+^3} + l_- \frac{E_- E_+ - p_-^2}{p_-^3} \right) \right] , \qquad (A.15)$$

where we have defined

$$E_- := E , \qquad E_+ := E' - E , \qquad p_\pm := \sqrt{E_\pm^2 - m_e^2} \qquad (A.16)$$

$$L := \ln \left(\frac{E_+ E_- + p_+ p_- + m_e^2}{E_+ E_- - p_+ p_- + m_e^2} \right) , \qquad l_\pm := \ln \left(\frac{E_\pm + p_\pm}{E_\pm - p_\pm} \right) . \qquad (A.17)$$

The Θ-function appearing in eq. (A.14) ensures that we fulfill energy conservation in the integration of E' over the range $[E, \infty]$ in eq. (2.7).

Compton scattering: $\gamma e_{\mathrm{th}}^- \to \gamma e^-$

The total rate for Compton scattering can be found in [29, 31] and is given by

$$\Gamma_\gamma^{(\mathrm{CS})}(E) = \frac{2\pi\alpha^2}{m_e^2} \times n_e(T) \times \frac{1}{x} \left[\left(1 - \frac{4}{x} - \frac{8}{x^2} \right) \ln(1 + x) + \frac{1}{2} + \frac{8}{x} - \frac{1}{2(1+x)^2} \right] \Bigg|_{x=2E/m_e} . \qquad (A.18)$$

Furthermore, the differential rate for the energy of the scattered photon reads [29, 31][15]

$$K_{\gamma \to \gamma}^{(\mathrm{CS})}(E, E') = \Theta(E - E'/(1 + 2E'/m_e)) \times \frac{\pi\alpha^2}{m_e} \times n_e(T) \times$$

$$\frac{1}{E'^2} \left[\frac{E'}{E} + \frac{E}{E'} + \left(\frac{m_e}{E} - \frac{m_e}{E'} \right)^2 - 2m_e \left(\frac{1}{E} - \frac{1}{E'} \right) \right] \qquad (A.19)$$

with the Θ-function corresponding to a vanishing rate above the Compton edge.

Following [31], the differential rate relevant for the spectrum of electrons can be deduced from eq. (A.19),

$$K_{\gamma \to e^-}^{(\mathrm{CS})}(E, E') = K_{\gamma \to \gamma}^{(\mathrm{CS})}(E' + m_e - E, E') . \qquad (A.20)$$

---

[15]Correcting a typo in eq. (10) of [29].

Inverse Compton scattering: $e^\pm \gamma_{\mathrm{th}} \to e^\pm \gamma$

The differential rate for production of photons from inverse Compton scattering was originally calculated in [46] and can be written as

$$K^{(\mathrm{IC})}_{e^\pm \to \gamma}(E, E') = 2\pi\alpha^2 \times \frac{1}{E'^2} \int_0^\infty \mathrm{d}\bar\epsilon \, \frac{\mathrm{f}_\gamma(\bar\epsilon)}{\bar\epsilon} \, F(E, E', \bar\epsilon) \times \Theta(E' - E - m_e) \,. \qquad (\mathrm{A}.21)$$

For $\bar\epsilon \le E \le 4\bar\epsilon E'^2 / (m_e^2 + 4\bar\epsilon E')$, the function $F(E, E', \bar\epsilon)$ is given by[16]

$$F(E, E', \bar\epsilon) = 2q\ln(q) + (1 + 2q)(1 - q) + \frac{\Gamma_\epsilon^2 q^2}{2 + 2\Gamma_\epsilon q}(1 - q) \,, \qquad (\mathrm{A}.22)$$

with

$$\Gamma_\epsilon = \frac{4\bar\epsilon E'}{m_e^2} \,, \qquad q = \frac{E}{\Gamma_\epsilon(E' - E)} \,, \qquad (\mathrm{A}.23)$$

and $F(E, E', \bar\epsilon) = 0$ otherwise.[17] Again, the Θ-function in eq. (A.21) ensures energy conservation upon integration of E' over the range $[E, \infty]$.

The total rate for inverse Compton scattering entering the calculation of the electron and positron spectrum is given by [31, 46][18]

$$\Gamma^{(\mathrm{IC})}_{e^\pm}(E) = 2\pi\alpha^2 \times \frac{1}{E^2} \int_0^\infty \mathrm{d}E_\gamma \int_0^\infty \mathrm{d}\bar\epsilon \, \frac{\mathrm{f}_\gamma(\bar\epsilon)}{\bar\epsilon} F(E_\gamma, E, \bar\epsilon) \,. \qquad (\mathrm{A}.24)$$

Finally, the differential rate for the production of electrons and positrons can be written as [31, 46]

$$K^{(\mathrm{IC})}_{e^\pm \to e^\pm}(E, E') = 2\pi\alpha^2 \times \frac{1}{E'^2} \int_0^\infty \mathrm{d}\bar\epsilon \, \frac{\mathrm{f}_\gamma(\bar\epsilon)}{\bar\epsilon} \, F(E' + \bar\epsilon - E, E', \bar\epsilon) \,. \qquad (\mathrm{A}.25)$$

Additional processes not considered in our calculation

Other processes such as

- Coulomb scattering $e^\pm e^-_{\mathrm{th}} \to e^\pm e^-$ and $N e^-_{\mathrm{th}} \to N e^-$,

- Thompson scattering $N\gamma_{\mathrm{th}} \to N\gamma$,

- Magnetic moment scattering $N e^-_{\mathrm{th}} \to N e^-$ or

- Electron-positron annihilation $e^+ e^-_{\mathrm{th}} \to \gamma\gamma$

are suppressed by the small density of background electrons or nuclei $n_e, n_N \ll n_\gamma$ and can therefore be neglected.

---

[16] Correcting a typo in eq. (49) of [31].

[17] According to [46], the function $F(E, E', \bar\epsilon)$ takes a different form for $E < \bar\epsilon$. However, this part of parameter space is practically irrelevant for our considerations.

[18] Correcting a typo in eq. (48) of [31].

B Some benchmarks

In this appendix we present benchmarks in order to better understand the actual runtime of the code. We ran the script `decay` with different values of m_ϕ and τ_ϕ while fixing $n_\phi/n_\gamma|_{T_0} = 10^{-10}$ at $T_0 = 10\,\mathrm{MeV}$, $\mathrm{BR}_{ee} = 0$ and $\mathrm{BR}_{\gamma\gamma} = 1$ (changing the latter parameters does not change the runtime). Using one core of an `Intel Core i5-6500 CPU @ 3.20GHz`, we obtain the following results

`./decay` m_ϕ [MeV] τ_ϕ [s] `10 1e-10 0 1`

m_ϕ [MeV]	τ_ϕ [s]	runtime (with db)	runtime (without db)
10	10^5	0min 44s	1min 16s
50	10^5	3min 15s	5min 19s
100	10^5	5min 13s	7min 52s
10	10^7	0min 26s	0min 39s
50	10^7	2min 46s	3min 26s
100	10^7	4min 00s	5min 15s

Given these results, it is worth noting that the runtime critically depends on the values of m_ϕ and τ_ϕ, since these parameters determine the energy and temperature range that is used for the calculation. For the `decay` model the relevant intervals are given by $[E_{\min}, m_\phi/2]$ and $[10^{1/2}T(\tau_\phi), 10^{-3/2}T(\tau_\phi)]$, respectively. Since the number of points per decade are fixed per default, larger values of m_ϕ lead to a larger energy grid and thus to a longer runtime.

Benchmarking the annihilation model, we ran the script `annihilation` for different values of m_χ while fixing $a = 10^{-25}\ \mathrm{cm}^3/\mathrm{s}$, $b = 0$, $T_{\mathrm{kd}} = 0$, $\mathrm{BR}_{ee} = 0$ and $\mathrm{BR}_{\gamma\gamma} = 1$ (changing the latter parameters again does not change the runtime). In this case, by using the same CPU, we find

`./annihilation` m_χ [MeV] `1e-25 0 0 0 1`

m_ϕ [MeV]	runtime (with db)	runtime (without db)
10	02min 03s	03min 52s
50	07min 20s	11min 34s
100	10min 46s	16min 37s

Again we find that the runtime is increased for larger values of m_χ, which determines the energy range $[E_{\min}, m_\chi]$.

We thus conclude that, depending on the model and choice of parameters, the runtime (with database files) can easily vary between $\mathcal{O}(10\,\mathrm{s})$ and $\mathcal{O}(1\,\mathrm{min})$. In order to still enable efficient (and fast) parameter scans, `ACROPOLIS` comes with a dedicated scanning framework, cf. section 5.2.3.

References

[1] V. F. Shvartsman, *Density of relict particles with zero rest mass in the universe*, *Pisma Zh. Eksp. Teor. Fiz.* **9** (1969) 315–317. [JETP Lett.9,184(1969)].

[2] G. Steigman, D. N. Schramm, and J. E. Gunn, *Cosmological Limits to the Number of Massive Leptons*, *Phys. Lett.* **66B** (1977) 202–204.

[3] R. J. Scherrer and M. S. Turner, *Primordial Nucleosynthesis with Decaying Particles. 1. Entropy Producing Decays. 2. Inert Decays*, *Astrophys. J.* **331** (1988) 19–32. [Astrophys. J.331,33(1988)].

[4] R. H. Cyburt, B. D. Fields, K. A. Olive, and T.-H. Yeh, *Big Bang Nucleosynthesis: 2015*, *Rev. Mod. Phys.* **88** (2016) 015004.

[5] A. Arbey, J. Auffinger, K. P. Hickerson, and E. S. Jenssen, *AlterBBN v2: A public code for calculating Big-Bang nucleosynthesis constraints in alternative cosmologies*, *Comput. Phys. Commun.* **248** (2020) 106982.

[6] O. Pisanti, A. Cirillo, S. Esposito, F. Iocco, G. Mangano, G. Miele, and P. D. Serpico, *PArthENoPE: Public Algorithm Evaluating the Nucleosynthesis of Primordial Elements*, *Comput. Phys. Commun.* **178** (2008) 956–971.

[7] C. Pitrou, A. Coc, J.-P. Uzan, and E. Vangioni, *Precision big bang nucleosynthesis with improved helium-4 predictions*, *Submitted to Phys. Rept.* (2018).

[8] R. H. Cyburt, J. R. Ellis, B. D. Fields, and K. A. Olive, *Updated nucleosynthesis constraints on unstable relic particles*, *Phys. Rev.* **D67** (2003) 103521.

[9] B. Batell, M. Pospelov, and A. Ritz, *Exploring Portals to a Hidden Sector Through Fixed Targets*, *Phys. Rev.* **D80** (2009) 095024.

[10] S. Andreas, C. Niebuhr, and A. Ringwald, *New Limits on Hidden Photons from Past Electron Beam Dumps*, *Phys. Rev.* **D86** (2012) 095019.

[11] K. Schmidt-Hoberg, F. Staub, and M. W. Winkler, *Constraints on light mediators: confronting dark matter searches with B physics*, *Phys. Lett.* **B727** (2013) 506–510.

[12] R. Essig, J. Mardon, M. Papucci, T. Volansky, and Y.-M. Zhong, *Constraining Light Dark Matter with Low-Energy e^+e^- Colliders*, *JHEP* **11** (2013) 167.

[13] E. Izaguirre, G. Krnjaic, P. Schuster, and N. Toro, *New Electron Beam-Dump Experiments to Search for MeV to few-GeV Dark Matter*, *Phys. Rev.* **D88** (2013) 114015.

[14] B. Batell, R. Essig, and Z. Surujon, *Strong Constraints on Sub-GeV Dark Sectors from SLAC Beam Dump E137*, *Phys. Rev. Lett.* **113** (2014), no. 17 171802.

[15] M. J. Dolan, F. Kahlhoefer, C. McCabe, and K. Schmidt-Hoberg, *A taste of dark matter: Flavour constraints on pseudoscalar mediators*, *JHEP* **03** (2015) 171. [Erratum: JHEP07,103(2015)].

[16] G. Krnjaic, *Probing Light Thermal Dark-Matter With a Higgs Portal Mediator*, *Phys. Rev.* **D94** (2016), no. 7 073009.

[17] M. J. Dolan, T. Ferber, C. Hearty, F. Kahlhoefer, and K. Schmidt-Hoberg, *Revised constraints and Belle II sensitivity for visible and invisible axion-like particles*, *JHEP* **12** (2017) 094.

[18] E. Izaguirre, Y. Kahn, G. Krnjaic, and M. Moschella, *Testing Light Dark Matter Coannihilation With Fixed-Target Experiments*, *Phys. Rev.* **D96** (2017) 055007.

[19] S. Knapen, T. Lin, and K. M. Zurek, *Light Dark Matter: Models and Constraints*, *Phys. Rev.* **D96** (2017), no. 11 115021.

[20] J. Beacham et al., *Physics Beyond Colliders at CERN: Beyond the Standard Model Working Group Report*, .

[21] K. Bondarenko, A. Boyarsky, T. Bringmann, M. Hufnagel, K. Schmidt-Hoberg, and A. Sokolenko, *Direct detection and complementary constraints for sub-GeV dark matter*, JHEP **03** (2020) 118.

[22] A. Filimonova, R. Schäfer, and S. Westhoff, *Probing dark sectors with long-lived particles at BELLE II*, .

[23] M. Hufnagel, K. Schmidt-Hoberg, and S. Wild, *BBN constraints on MeV-scale dark sectors. Part I. Sterile decays*, JCAP **1802** (2018) 044.

[24] M. Hufnagel, K. Schmidt-Hoberg, and S. Wild, *BBN constraints on MeV-scale dark sectors. Part II. Electromagnetic decays*, JCAP **1811** (2018) 032.

[25] L. Forestell, D. E. Morrissey, and G. White, *Limits from BBN on Light Electromagnetic Decays*, .

[26] P. F. Depta, M. Hufnagel, K. Schmidt-Hoberg, and S. Wild, *BBN constraints on the annihilation of MeV-scale dark matter*, JCAP **1904** (2019) 029.

[27] P. F. Depta, M. Hufnagel, and K. Schmidt-Hoberg, *Robust cosmological constraints on axion-like particles*, JCAP **05** (2020) 009.

[28] M. Kawasaki, K. Kohri, T. Moroi, K. Murai, and H. Murayama, *Big-bang nucleosynthesis with sub-GeV massive decaying particles*, .

[29] V. Poulin and P. D. Serpico, *Nonuniversal BBN bounds on electromagnetically decaying particles*, Phys. Rev. **D91** (2015), no. 10 103007.

[30] J. J. Renk et al., *CosmoBit: A GAMBIT module for computing cosmological observables and likelihoods*, .

[31] M. Kawasaki and T. Moroi, *Electromagnetic cascade in the early universe and its application to the big bang nucleosynthesis*, Astrophys. J. **452** (1995) 506.

[32] K. Jedamzik, *Big bang nucleosynthesis constraints on hadronically and electromagnetically decaying relic neutral particles*, Phys. Rev. **D74** (2006) 103509.

[33] M. Hufnagel, *Primordial Nucleosynthesis in the Presence of MeV-scale Dark Sectors*. PhD thesis, Hamburg U., Hamburg, 2020.

[34] P. F. Depta, M. Hufnagel, and K. Schmidt-Hoberg, *Updated BBN constraints on electromagnetic decays of MeV-scale particles*, .

[35] J. Mardon, Y. Nomura, D. Stolarski, and J. Thaler, *Dark Matter Signals from Cascade Annihilations*, JCAP **0905** (2009) 016.

[36] A. Birkedal, K. T. Matchev, M. Perelstein, and A. Spray, *Robust gamma ray signature of WIMP dark matter*, .

[37] A. Arbey, *AlterBBN: A program for calculating the BBN abundances of the elements in alternative cosmologies*, Comput. Phys. Commun. **183** (2012) 1822–1831.

[38] T. Bringmann, P. F. Depta, M. Hufnagel, and K. Schmidt-Hoberg, *Precise dark matter relic abundance in decoupled sectors*, .

[39] N. Aghanim et al., *Planck 2018 results. VI. Cosmological parameters*, .

[40] P. Zyla et al., *Review of Particle Physics*, PTEP **2020** (2020), no. 8 083C01.

[41] J. Geiss and G. Gloeckler, *Isotopic Composition of H, HE and NE in the Protosolar Cloud*, Space Science Reviews **106** (Apr, 2003).

[42] F. A. Aharonian, A. M. Atoian, and A. M. Nagapetian, *Photoproduction of electron-positron pairs in compact X-ray sources*, Astrofizika **19** (Apr., 1983) 323–334.

[43] R. Svensson and A. A. Zdziarski, *Photon-photon scattering of gamma rays at cosmological distances*, Astrophys. J. **349** (1990) 415–428.

[44] L. C. Maximon, *Simple analytic expressions for the total born approximation cross section for pair production in a coulomb field*, JOURNAL OF RESEARCH of the Notional Bureau of Standards (1968).

[45] L. D. Landau, V. B. Berestetskii, E. M. Lifshitz, and L. P. Pitaevskii, *Relativistic quantum theory.* Course of theoretical physics. Pergamon, Oxford, 1971. Trans. from the Russian.

[46] F. C. Jones, *Calculated Spectrum of Inverse-Compton-Scattered Photons*, Phys. Rev. **167** (1968) 1159–1169.