pyBBN

User Guide

Nashwan Sabti[1], Andrii Magalich[2], Anastasiia Filimonova[3]

Maintained by Nashwan Sabti

Corresponding to code version v1.0

July 15, 2020

[1]nashwan.sabti@kcl.ac.uk
[2]magalich@lorentz.leidenuniv.nl
[3]filimonova@thphys.uni-heidelberg.de

Contents

Chapter 1

pyBBN

1.1 Installation and usage

1. Clone the repository from the GitHub page

   ```
   git clone --recursive https://github.com/ckald/pyBBN
   ```

2. Install dependencies using pip (usually preinstalled on Linux and Mac OS X along with the standard CPython interpreter). Currently the code supports only Python3.5+. In case the default `python3` command points to an older version, install Python3.5+ and use the exact version (e.g., `python3.5` or `python3.6`) everywhere instead of `python3`.

   ```
   python3 -m pip install -r requirements.txt
   ```

   In case there are any authorization issues, try the `sudo` command.

3. Compile modified KAWANO nucleosynthesis code

   ```
   cd KAWANO/
   make all
   ```

4. Compile `pyBBN` code extensions

   ```
   make all
   ```

If the command `python3` does not correspond to Python3.5+, use `PYTHON=python3.x make all`.

5. To run a script, use

```
PYTHONPATH=. python3 some_script.py
```

For example, to run the cosmic neutrino background temperature test using Python3.5+

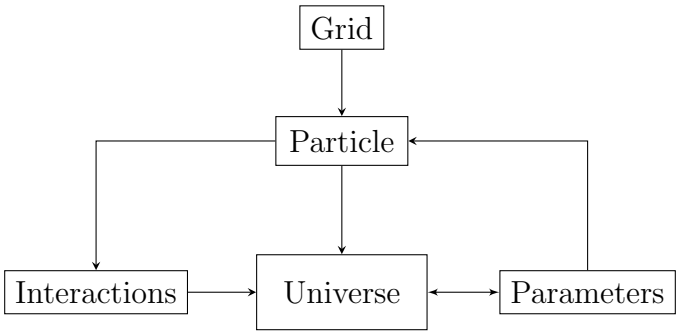
```
PYTHONPATH=. python3 tests/cosmic_neutrino_temperature.py
```

1.2 Building blocks of the code

Simulations are done in two steps: (I) The background physics and the rates of the reactions in Eqs. **??** - **??** are computed in `pyBBN`. This involves solving the system of equations in Subsection **??** for the evolution of temperature, scale factor and distribution functions of decoupled species. (II) The cosmological quantities together with the aforementioned rates are tabulated and passed to an external code, the modified `KAWANO` code [1], that takes care of the nuclear physics part of the simulation and outputs the light element abundances.

1.2.1 pyBBN

The main building blocks of pyBBN are classes. Each box below represent a class.



- Grid: The user has a choice between a linear and logarithmic spaced grid. For given HNL mass and mixing angle, the code computes the maximum momentum of the grid and

the grid resolution (such that changes in distribution functions can be resolved) for all particles in the simulation. This feature is only available for linear spaced grids.

- Particle: This class contains all particle characteristics and switches to different regimes when the temperature drops below certain thresholds. For each regime, a suitable routine is appointed.
- Interactions: Both SM interactions and interactions involving HNLs are incorporated as class member functions, which return a helper class that takes care of creating all necessary collision integrals for the actual interactions.
- Parameters: The initial values of cosmological variables are stored here and altered during each step of the simulation. The new values are then passed on to the Particle class.
- Universe: The master object that governs the calculation. Particles are added to this class and loaded with those interactions in which they are involved. The timestep of the simulation can be chosen either as a dimensionful linear scale factor or as a logarithm of scale factor.

1.2.2 Modified KAWANO

The modified KAWANO code obtains tabulated values of cosmological parameters and rates of **??** - **??** to take care of the nuclear physics part of the simulation. There is a Python module that computes these rates during each step and in the end communicates with the modified KAWANO code. The present code is updated from the version used in [1]. It is now more automated and fixes a bug present in the previous version, which is elaborated on later.

1.3 Template of test file

The test file below can be run by

```
PYTHONPATH=. python3 tests/template.py
```

Depending on the number of grid samples, a test like the one below currently takes 4 - 5 days to finish on a 64-core node.

```
import numpy as np
import os
from collections import defaultdict

import environment
from particles import Particle
from library.SM import particles as SMP, interactions as SMI
from library.NuMSM import particles as NuP, interactions as NuI
from evolution import Universe
from common import CONST, UNITS, Params, utils, LinearSpacedGrid, kinematics
from interactions.four_particle.cpp.integral import CollisionIntegralKind

# environment module contains default options that can be overwritten
# CONST class contains physical constants
# utils module contains various utilities used throughout the code
# kinematics module computes initial temperature and relevant grid parameters
# See user guide for explanation of CollisionIntegralKind

mass = 200 * UNITS.MeV
theta = 1e-4

# Makes a folder with mass and mixing angle in its name
folder = utils.ensure_dir(
    os.path.split(__file__)[0],
    "output",
    "mass={:e}_theta={:e}_flav={}".format(mass / UNITS.MeV, theta, "el")
)

T_initial = kinematics.decoupling_temperature(mass, theta)
T_freeze = 0.1 * UNITS.MeV
T_final = 0.0008 * UNITS.MeV
# y = ln(a) and dy = da/a
params = Params(T=T_initial, dy=0.003125)

# Create universe master object
universe = Universe(params=params, folder=folder)

# Grid parameters
# std means a default resolution of 4 or 8 samples / MeV
samples_HNL, max_mom_HNL = kinematics.grid_params_HNL(mass, theta)
samples_neutrino, max_mom_neutrino = kinematics.grid_params_neutrino(mass, theta)
samples_muon, max_mom_muon = kinematics.grid_params_lepton(mass, theta)
samples_electron, max_mom_electron = kinematics.grid_params_lepton(mass, theta, std=True)
samples_pion, max_mom_pion = kinematics.grid_params_meson(mass, theta)

linear_grid_HNL = LinearSpacedGrid(MOMENTUM_SAMPLES=samples_HNL, MAX_MOMENTUM=max_mom_HNL)
linear_grid_neutrino = LinearSpacedGrid(MOMENTUM_SAMPLES=samples_neutrino, MAX_MOMENTUM=
    max_mom_neutrino)
```

```
linear_grid_muon = LinearSpacedGrid(MOMENTUM_SAMPLES=samples_muon, MAX_MOMENTUM=max_mom_muon)
linear_grid_electron = LinearSpacedGrid(MOMENTUM_SAMPLES=samples_electron, MAX_MOMENTUM=
    max_mom_electron)
linear_grid_pion = LinearSpacedGrid(MOMENTUM_SAMPLES=samples_pion, MAX_MOMENTUM=max_mom_pion)

# Particle instances with default (specified in environment module) and non-default grid
photon = Particle(**SMP.photon)

neutrino_e = Particle(**SMP.leptons.neutrino_e, grid=linear_grid_neutrino)
neutrino_mu = Particle(**SMP.leptons.neutrino_mu, grid=linear_grid_neutrino)
neutrino_tau = Particle(**SMP.leptons.neutrino_tau, grid=linear_grid_neutrino)

electron = Particle(**SMP.leptons.electron, grid=linear_grid_electron)
muon = Particle(**SMP.leptons.muon, grid=linear_grid_muon)

sterile = Particle(**NuP.dirac_sterile_neutrino(mass, theta), grid=linear_grid_HNL)

up = Particle(**SMP.quarks.up)
down = Particle(**SMP.quarks.down)
strange = Particle(**SMP.quarks.strange)
charm = Particle(**SMP.quarks.charm)
gluon = Particle(**SMP.gluon)

charged_pion = Particle(**SMP.hadrons.charged_pion, grid=linear_grid_pion)
neutral_pion = Particle(**SMP.hadrons.neutral_pion, grid=linear_grid_pion)

# Modify decoupling temperature of particles
sterile.decoupling_temperature = T_initial
neutrino_e.decoupling_temperature = 5. * UNITS.MeV
neutrino_mu.decoupling_temperature = 5. * UNITS.MeV
neutrino_tau.decoupling_temperature = 5. * UNITS.MeV

# Add particles that should be in the system by default
universe.add_particles([
    photon,

    electron,
    muon,

    neutrino_e,
    neutrino_mu,
    neutrino_tau,

    sterile,
])

# Depending on the initial temperature, there are either quarks or mesons present
# lambda_QCD is the QCD phase transition temperature
if T_initial > CONST.lambda_QCD:
```

```
    universe.add_particles([
    up,
    down,
    strange,
    charm,
    gluon
    ])
else:
    universe.add_particles([
    neutral_pion,
    charged_pion
    ])

# Dictionary of mixing angles. Here only nu_e-mixing is enabled
thetas = defaultdict(float, {
    'electron': theta
})

# All relevant interactions
SM_interactions = SMI.neutrino_interactions(
               leptons=[electron, muon],
               neutrinos=[neutrino_e, neutrino_mu, neutrino_tau],
             )

# Remove SM interactions for muons
SM_interactions = utils.interaction_filter(['Muon'], SM_interactions)

sterile_quark = NuI.sterile_quark_interactions(
                thetas=thetas, sterile=sterile,
                neutrinos=[neutrino_e, neutrino_mu, neutrino_tau],
                leptons=[electron, muon],
                quarks=[up, down, strange, charm]
            )

sterile_quark = utils.interaction_filter(
                ['Up quark', 'Down quark', 'Charm quark',
                'Strange quark', 'Top quark', 'Bottom quark', 'Gluon'],
                sterile_quark
                )

HNL_leptons_interactions = NuI.sterile_leptons_interactions(
    thetas=thetas, sterile=sterile,
    neutrinos=[neutrino_e, neutrino_mu, neutrino_tau],
    leptons=[electron, muon]
)

HNL_hadrons_interactions = NuI.sterile_hadrons_interactions(
        thetas = thetas, sterile=sterile,
        neutrinos = [neutrino_e],
```

```
        leptons = [electron],
        mesons = [neutral_pion, charged_pion]
)

# These interactions are for the unstable HNL decay products
# The one below is for subsequent interactions of muons coming from HNL decay
Secondary_interactions_leptons = NuI.interactions_decay_products(
    interactions_primary=[HNL_leptons_interactions],
    neutrinos=[neutrino_e, neutrino_mu, neutrino_tau],
    leptons=[electron, muon],
    mesons=[],
    photon=[],
    kind=CollisionIntegralKind.F_decay
)

# muon_dec is set 'True' if muon decay is already included (to avoid double counting)
Secondary_interactions_mesons = NuI.interactions_decay_products(
    interactions_primary=[HNL_hadrons_interactions],
    muon_dec=True,
    neutrinos=[neutrino_e, neutrino_mu, neutrino_tau],
    leptons=[electron, muon],
    mesons=[neutral_pion, charged_pion],
    photon=[photon],
    kind=CollisionIntegralKind.F_decay
)

# Add interactions that should be in the system by default
universe.interactions += (
    SM_interactions + HNL_leptons_interactions + Secondary_interactions_leptons
)

# Depending on initial temperature, either quarks or mesons are present
if T_initial > CONST.lambda_QCD:
    universe.interactions += (
    sterile_quark
    )
else:
    universe.interactions += (
    HNL_hadrons_interactions + Secondary_interactions_mesons
    )

# Initialize the KAWANO module with pointers to electron and nu_e
universe.init_kawano(electron=electron, neutrino=neutrino_e)

# Function that allows for monitoring simulation progress
# Example below writes file with a, T, aT, energy and number density for neutrinos and HNL
def step_monitor(universe):
    if universe.step % 10 == 0:
        for particle in [neutrino_e, neutrino_mu, neutrino_tau, sterile]:
```

```
194             with open(os.path.join(folder, particle.name.replace(' ', '_') + ".rho.txt"), 'a')
                     as f:
195                 f.write('{:e}\t{:e}\t{:e}\t{:e}\n'.format(
196                     universe.params.a,
197                     universe.params.T / UNITS.MeV,
198                     universe.params.aT / UNITS.MeV,
199                     particle.energy_density / UNITS.MeV**4,
200                     particle.density / UNITS.MeV**3
201                 ))
202
203 universe.step_monitor = step_monitor
204
205 # Overview of some parameters of the simulation
206 print('Mass:', mass / UNITS.MeV, 'theta^2:', theta**2)
207 print('T_initial:', T_initial / UNITS.MeV)
208 print('T_freeze:', T_freeze/ UNITS.MeV)
209 print('T_final:', T_final / UNITS.MeV)
210 print('HNL\n','Samples: ', samples_HNL, 'MAX_MOM:', max_mom_HNL / UNITS.MeV, 'T_dec:', sterile
    .decoupling_temperature / UNITS.MeV)
211 print('neutrino\n','Samples: ', samples_neutrino, 'MAX_MOM:', max_mom_neutrino / UNITS.MeV, '
    T_dec:', neutrino_e.decoupling_temperature / UNITS.MeV)
212 print('electron\n','Samples: ', samples_electron, 'MAX_MOM:', max_mom_electron / UNITS.MeV, '
    T_dec:', electron.decoupling_temperature / UNITS.MeV)
213 print('muon\n','Samples: ', samples_muon, 'MAX_MOM:', max_mom_muon / UNITS.MeV, 'T_dec:', muon
    .decoupling_temperature / UNITS.MeV)
214 print('pion\n','Samples: ', samples_pion, 'MAX_MOM:', max_mom_pion / UNITS.MeV, 'T_dec:',
    neutral_pion.decoupling_temperature / UNITS.MeV)
215
216 # When initial temperature is higher than QCD transition temperature, system
217 # must evolve till lambda_QCD, undergo QCD phase transition (replacing quarks by mesons)
218 # and then continue evolving. At T_freeze all interactions are cleared away
219 if T_initial > CONST.lambda_QCD:
220     universe.evolve(CONST.lambda_QCD, export=False)
221     universe.QCD_transition(hadrons=[charged_pion, neutral_pion],
222                             quarkic_interactions=sterile_quark,
223                             hadronic_interactions=HNL_hadrons_interactions,
224                             secondary_interactions=Secondary_interactions_mesons
225                         )
226     universe.evolve(T_freeze, export=False)
227     universe.interactions = tuple()
228     universe.evolve(T_final)
229 else:
230     universe.evolve(T_freeze, export=False)
231     universe.interactions = tuple()
232     universe.evolve(T_final)
```

1.4 Computational schemes

1.4.1 Decoupling temperatures

The decoupling temperature in the code is actually the temperature from where on the Boltzmann equation will be solved. Before decoupling, (almost) each particle species has an equilibrium distribution, which is the initial condition needed when solving the Boltzmann equation. Even if a particle is not decoupled yet, the result will be the same as when using equilibrium physics, given that all relevant interactions are included.

1.4.1.1 HNLs

As shown before, the mixing angle can be taken as the one in vacuum. The decoupling temperature is then estimated as

$$T_{\text{dec}} = \begin{cases} 1.5 \cdot T_{\text{rel}} & \text{if } 1.5 \cdot T_{\text{rel}} > 1.5 \cdot M_N \\ 1.5 \cdot M_N & \text{if } M_N \leq 1.5 \cdot T_{\text{rel}} \leq 1.5 \cdot M_N \\ M_N & \text{if } 1.5 \cdot T_{\text{rel}} < M_N \end{cases} \tag{1.1}$$

This temperature is also taken as the initial temperature of the simulation.

1.4.1.2 Active neutrinos

The decoupling temperature of active neutrinos is theoretically estimated to be ~ 1.5 MeV. In the code, the default decoupling temperature is set to 5 MeV, in order to account for the contribution of HNLs to the cosmological expansion and its energy injection.

1.4.1.3 Photons and electrons

These particles are always in equilibrium and the 'decoupling temperature' is set to 0 MeV.

1.4.1.4 Muons

There are a number of rates that have to be compared here:

- Γ_{decay} with $\Gamma_{\mu\mu\to\gamma\gamma}$:

$$3 \cdot 10^{-16}\ \text{MeV} = \Gamma_{\text{decay}} = \Gamma_{\mu\mu\to\gamma\gamma} \approx \frac{\alpha^2}{m_\mu^2}\left(\frac{m_\mu T}{2\pi}\right)^{\frac{3}{2}} e^{-\frac{m_\mu}{T}} \Longrightarrow T \approx 4.6\ \text{MeV}. \tag{1.2}$$

- Γ_{decay} with $\Gamma_{\mu\nu_\mu\to \text{e}\nu_\text{e}}$:

$$\begin{aligned}
\Gamma_{\mu\nu_\mu\to \text{e}\nu_\text{e}} &= n_{\nu_\mu}\sigma v \\
&= \frac{3}{4}\frac{\zeta(3)}{\pi^2}T^3 \int \frac{1}{64\pi^2}\frac{|\mathcal{M}|^2}{\sqrt{(P_\mu \cdot P_{\nu_\mu})^2 - m_\mu^2 m_{\nu_\mu}^2}}\frac{1}{\sqrt{s}}\,|\mathbf{p}_e|\,\mathrm{d}\Omega \\
&// \ s = \left(P_\mu + P_{\nu_\mu}\right)^2 \gg m_e^2 \ // \\
&// \ |\mathcal{M}|^2 = 64 G_\text{F}^2 (P_\mu \cdot P_{\nu_\mu})(P_e \cdot P_{\nu_e}) \ // \\
&// \ |\mathbf{p_e}| = \frac{1}{2\sqrt{s}}\sqrt{s^2 - 2s(m_e^2 + m_{\nu_e}^2) + (m_e^2 - m_{\nu_e}^2)^2} \approx \frac{\sqrt{s}}{2} \ // \\
&// \ \mathbf{p}_e + \mathbf{p}_{\nu_e} = \mathbf{0} \ // \\
&// \ E_e \approx |\mathbf{p}_e| \ // \\
&\approx \frac{3}{8}\frac{\zeta(3)G_\text{F}^2}{\pi^4}T^3 \int (E_e E_{\nu_e} - \mathbf{p}_e \cdot \mathbf{p}_{\nu_e})\,\mathrm{d}\Omega \\
&\approx \frac{3}{8}\frac{\zeta(3)G_\text{F}^2}{\pi^4}T^3 \int 2\,|\mathbf{p_e}|^2\,\mathrm{d}\Omega \\
&\approx \frac{3}{8}\frac{\zeta(3)G_\text{F}^2}{\pi^4}T^3 2 \cdot \frac{s}{4} \cdot 4\pi \\
&= \frac{3}{4}\frac{\zeta(3)G_\text{F}^2}{\pi^3}T^3 \left(P_\mu + P_{\nu_\mu}\right)^2 \\
&// \ \text{Lab frame @ T} \approx 1\ \text{MeV}: \ |\mathbf{p}_\mu| \approx 0 \ // \\
&// \ \left|\mathbf{p}_{\nu_\mu}\right| \approx \pi T \ // \\
&\approx \frac{3}{4}\frac{\zeta(3)G_\text{F}^2}{\pi^3}T^3 \left(\left(m_\mu + \left|\mathbf{p}_{\nu_\mu}\right|\right)^2 - \left|\mathbf{p}_{\nu_\mu}\right|^2\right) \\
&\approx \frac{3}{4}\frac{\zeta(3)G_\text{F}^2}{\pi^3}T^3 \left(m_\mu^2 + 2\pi m_\mu T\right)
\end{aligned} \tag{1.3}$$

Equating this temperature to the muon decay rate gives $T \approx 15$ MeV.

This means that muon decay dominates for temperatures below 4.6 MeV. This is taken as the decoupling temperature of muons. Muon annihilation is neglected. After decoupling, the

distribution function will fall off as

$$f_\mu = f_\mu(T_{\rm dec}) e^{-\frac{t-t_{\rm dec}}{\tau_\mu}} \tag{1.4}$$

1.4.1.5 Pions

A rough estimation can be made for the decoupling temperature of pions, if one considers $\pi^+\pi^- \leftrightarrow \pi^0\pi^0$ to be the dominant reaction that keeps them in equilibrium. A Lagrangian for such interaction can be written as

$$\mathcal{L} \sim \frac{1}{f_\pi^2}(\partial_\mu \pi)\pi(\partial^\mu \pi)\pi \tag{1.5}$$

The reaction rate can then be estimated as

$$\Gamma = n_\pi \sigma v \sim \left(\frac{m_\pi T}{2\pi}\right)^{\frac{3}{2}} e^{-\frac{m_\pi}{T}} \frac{m_\pi^4}{f_\pi^4 T^2}\sqrt{\frac{T}{m_\pi}} \tag{1.6}$$

Equating this rate to the pion decay rate, gives a decoupling temperature of ~ 7 MeV. Bear in mind that this is a very rough estimation. Their distribution function after decoupling falls off like in Eq. 1.4. Since pions have such a small lifetime, the exponent makes their distribution function essentially zero. Therefore, all pions available in the system during BBN temperatures originate from HNL decay. Same story holds for muons.

1.4.1.6 Other mesons

Other mesons (starting from eta meson with mass ~ 550 MeV) will be very non-relativistic at BBN temperatures and the Boltzmann suppression factor will dominate. The distribution functions are therefore taken to be zero. This can be done by setting the variable `thermal_dyn` to `False`. For example, the eta meson can be defined as

```
eta = Particle(**SMP.hadrons.eta, thermal_dyn=False)
```

All mesons available in the system are thus solely due to HNL decay. The decoupling temperature is chosen $T = 7$ MeV, a bit before active neutrino decoupling, to account for heat-up of the plasma from subsequent decays.

1.4.2 Grid parameters

The grid class has two parameters that can be adjusted, the maximum momentum of the grid `MAX_MOMENTUM` and the number of samples `MOMENTUM_SAMPLES`. The default values, set in the environment module, are `MAX_MOMENTUM` = 100 MeV and `MOMENTUM_SAMPLES` = 400. This gives a resolution of 4 samples/MeV. This resolution is chosen in order to have a convergence of the ^{4}He abundance in SBBN. In simulations involving HNLs, there is a module available (kinematics module) that provides suitable grid parameters for all particle species for given HNL mass and mixing angle. These grid parameters account for resolving changes in distribution functions due to three- and four-particle interactions.

1.4.2.1 HNLs

The maximum momentum is determined when the initial distribution function drops below $f_{\min} = 10^{-100}$ and is given by

$$y_{\max} = \sqrt{\left((aT)_{\text{ini}} \ln\left(\frac{1}{f_{\min}}\right) - 1\right)^2 - m_N^2 a_{\text{ini}}^2}\ , \tag{1.7}$$

with $(aT)_{\text{ini}} = 1$ MeV and a_{ini} the initial scale factor.

There are two cases that must be considered when estimating the grid resolution.
HNLs decouple non-relativistically:

$$\begin{aligned} f_N &= \frac{1}{e^{\frac{\sqrt{m_N^2+p^2}}{T}} + 1} \\ &\quad // \text{ After decoupling : } f(t,p) = f\left(\frac{a}{a_{\text{dec}}}p\right) = f\left(\frac{y}{a_{\text{dec}}}\right) \ // \\ &= \frac{1}{e^{\frac{\sqrt{m_N^2+\frac{y^2}{a_{\text{dec}}^2}}}{T_{\text{dec}}}} + 1} \approx \frac{1}{e^{\frac{m_N}{T_{\text{dec}}}+\frac{y^2}{2m_N a_{\text{dec}}^2 T_{\text{dec}}}} + 1} \\ &\approx e^{-\frac{m_N}{T_{\text{dec}}}-\frac{y^2}{2m_N a_{\text{dec}}^2 T_{\text{dec}}}} \end{aligned} \tag{1.8}$$

$$\langle y^2 \rangle = \frac{\int \mathrm{d}y y^4 e^{-\frac{y^2}{2m_N a_{\text{dec}}^2 T_{\text{dec}}}}}{\int \mathrm{d}y y^2 e^{-\frac{y^2}{2m_N a_{\text{dec}}^2 T_{\text{dec}}}}} = 3m_N a_{\text{dec}}^2 T_{\text{dec}} \approx 3m_N \frac{a^2 T^2}{T_{\text{dec}}} \tag{1.9}$$

$$\langle y \rangle = \frac{\int \mathrm{d}y y^3 e^{-\frac{y^2}{2m_N a_{\text{dec}}^2 T_{\text{dec}}}}}{\int \mathrm{d}y y^2 e^{-\frac{y^2}{2m_N a_{\text{dec}}^2 T_{\text{dec}}}}} = \sqrt{\frac{8m_N a_{\text{dec}}^2 T_{\text{dec}}}{\pi}} \approx aT\sqrt{\frac{8m_N}{\pi T_{\text{dec}}}} \tag{1.10}$$

$$\Delta y = \sqrt{\langle y^2\rangle - \langle y\rangle^2} = aT\sqrt{\left(3-\frac{8}{\pi}\right)\frac{m_N}{T_{\rm dec}}} \tag{1.11}$$

In the non-relativistic case, $T_{\rm dec} < m_N$ and therefore $\Delta y \gtrsim aT\sqrt{\left(3-\frac{8}{\pi}\right)} \approx 0.67aT$.

HNLs decouple relativistically:

$$\begin{aligned} f_N &= \frac{1}{e^{\frac{\sqrt{m_N^2+p^2}}{T}}+1} \\ &\text{// After decoupling //} \\ &= \frac{1}{e^{\frac{\sqrt{m_N^2+\frac{y^2}{a_{\rm dec}^2}}}{T_{\rm dec}}}+1} \approx \frac{1}{e^{\frac{y}{a_{\rm dec}T_{\rm dec}}}+1} \end{aligned} \tag{1.12}$$

$$\langle y^2\rangle = \frac{\int {\rm d}y \frac{y^4}{e^{\frac{y}{a_{\rm dec}T_{\rm dec}}}+1}}{\int {\rm d}y \frac{y^2}{e^{\frac{y}{a_{\rm dec}T_{\rm dec}}}+1}} = 15\frac{\zeta(5)}{\zeta(3)}a_{\rm dec}^2T_{\rm dec}^2 \approx 15\frac{\zeta(5)}{\zeta(3)}a^2T^2 \tag{1.13}$$

$$\langle y\rangle = \frac{\int {\rm d}y \frac{y^3}{e^{\frac{y}{a_{\rm dec}T_{\rm dec}}}+1}}{\int {\rm d}y \frac{y^2}{e^{\frac{y}{a_{\rm dec}T_{\rm dec}}}+1}} = \frac{7\pi^4}{180\zeta(3)}a_{\rm dec}T_{\rm dec} \approx \frac{7\pi^4}{180\zeta(3)}aT \tag{1.14}$$

$$\Delta y = \sqrt{\langle y^2\rangle - \langle y\rangle^2} = aT\sqrt{15\frac{\zeta(5)}{\zeta(3)} - \left(\frac{7\pi^4}{180\zeta(3)}\right)^2} \approx 1.73aT \tag{1.15}$$

Since the initial value of aT is $a_{\rm ini}T_{\rm ini} = 1$ MeV and only grows during the evolution of the Universe, the minimum values are

$$\Delta y_{N,\rm min} \approx \begin{cases} 0.67 \text{ MeV}, & \text{Non} - \text{relativistic decoupling} \\ 1.73 \text{ MeV}, & \text{Relativistic decoupling} \end{cases} \tag{1.16}$$

In the code, the resolution is taken 0.67 MeV/step for all cases.

1.4.2.2 Mesons

The maximum momentum is taken as

$$y_{\rm max} \approx \begin{cases} 100 \text{ MeV} & \text{if } m_N < m_{\rm pion} \\ 10a_{\rm max} \text{ MeV} & \text{if } m_N - m_{\rm meson} < 0.35 \text{ MeV} \\ a_{\rm max}\sqrt{m_N^2 - m_{\rm meson}^2} & \text{otherwise} \end{cases} \tag{1.17}$$

with $a_{\rm max} = 10$. This corresponds to a temperature of $\mathcal{O}(0.1)$ MeV. This is rather a conservative approach, considering that HNLs that have not decayed yet at such temperatures are almost

certainly excluded. The value 0.35 MeV takes into account that some mesons (e.g. pions) are present in the system at high temperatures and with high momenta. For such cases the maximum momentum should be high enough to sample the distribution function properly up to $f(y_{\max}) \sim 10^{-40}$.

Mesons are created from HNL two-body decays. The center-of-mass momentum of both particles A and B in the reaction $N \to A + B$ is given by

$$p_{\rm CM}^2 = \frac{\left(m_N^2 - m_A^2 - m_B^2\right)^2 - 4m_A^2 m_B^2}{4m_N^2} \tag{1.18}$$

The maximum and minimum energy of particle i ($i = A, B$) in the lab frame can be obtained by a Lorentz transformation:

$$E_{i,\rm lab,max} = \gamma \left(E_{i,\rm CM} + \beta p_{\rm CM}\right) \tag{1.19}$$

$$E_{i,\rm lab,min} = \gamma \left(E_{i,\rm CM} - \beta p_{\rm CM}\right) \tag{1.20}$$

$$\beta = \frac{p_N}{\sqrt{m_N^2 + p_N^2}} \quad \gamma = \frac{1}{\sqrt{1-\beta^2}}$$

Assuming HNLs are non-relativistic at BBN temperatures, the resolution can be roughly estimated as

$$\begin{aligned}
\Delta p_i &= p_{i,\rm lab,max} - p_{i,\rm lab,min} \\
&= \sqrt{\left(\gamma \left(E_{i,\rm CM} + \beta p_{\rm CM}\right)\right)^2 - m_i^2} - \sqrt{\left(\gamma \left(E_{i,\rm CM} - \beta p_{\rm CM}\right)\right)^2 - m_i^2} \\
&\approx \sqrt{\left(E_{i,\rm CM} + \frac{\Delta p_N}{m_N} p_{\rm CM}\right)^2 - m_i^2} - \sqrt{\left(E_{i,\rm CM} - \frac{\Delta p_N}{m_N} p_{\rm CM}\right)^2 - m_i^2} \\
\Delta y_{\rm meson} &\approx \sqrt{\left(a E_{\rm meson,CM} + \frac{\Delta y_N}{m_N} p_{\rm CM}\right)^2 - a^2 m_{\rm meson}^2} \\
&\quad - \sqrt{\left(a E_{\rm meson,CM} - \frac{\Delta y_N}{m_N} p_{\rm CM}\right)^2 - a^2 m_{\rm meson}^2} \,,
\end{aligned} \tag{1.21}$$

where a can be taken $a \sim \mathcal{O}(1)$, when the temperature is around 1 MeV. Plugging in different values for a does not seem to change the result significantly.

There is a different calculation that must be done that involves the required resolution for meson decay, when they are non-relativistic. It follows a similar procedure as for HNLs above:

$$f_{\rm m} = \frac{1}{e^{\frac{E_{\rm m} - \mu}{T}} - 1} \approx e^{-\frac{m_{\rm m} - \mu}{T}} e^{-\frac{p^2}{2 m_{\rm m} T}} \tag{1.22}$$

$$\langle y^2 \rangle = 3 m_{\rm m} a^2 T \tag{1.23}$$

$$\langle y \rangle = a \sqrt{\frac{8 m_{\rm m} T}{\pi}} \tag{1.24}$$

$$\Delta y = \sqrt{\left(3 - \frac{8}{\pi}\right) m_{\rm m} a^2 T} \gtrsim 0.67 \sqrt{m_{\rm m} a \text{ MeV}} \tag{1.25}$$

The scale factor a here is $\mathcal{O}(1)$, which makes this resolution larger than the one in Eq. 1.21.

Different interactions use different grid points (see below). For three-particle interactions the code uses the resolution in Eq. 1.21 divided by 3. Subsequent interactions of mesons can involve three-body decays, which use a default resolution of 0.25 MeV/step.

1.4.2.3 Active neutrinos

The maximum momentum is chosen to be

$$y_{\rm max} = E_N = \sqrt{y_N^2 + m_N^2 a_{\rm max}^2} \approx m_N a_{\rm max} \ , \tag{1.26}$$

where $a_{\rm max} = 10$.

For active neutrinos, three-particle reactions require grids with a much higher resolution than four-particle reactions to resolve the collision integral. The resolution for four-particle reactions is chosen to be 0.125 MeV/step if the mass difference between HNL and muon is smaller than 10 MeV and 0.25 MeV/step otherwise (default value). These values are empirically determined, so as to give reasonable abundances of neutrinos created in such reactions.

There are two cases that must be considered when estimating the grid resolution for three-particle reactions.

- Neutrinos created from HNL decay.

  Equation 1.21 can be used, where the mass is set to zero:

  $$\Delta y_\nu \approx 2 \frac{\Delta y_N}{m_N} p_{\rm CM} \approx\approx \frac{1.34}{m_N} p_{\rm CM} \text{ MeV} \tag{1.27}$$

- Neutrinos created from meson decay.

  The result here is similar to above, but with the resolution in Eq. 1.25 substituted (since it is the minimum resolution required to resolve meson decay properly):

  $$\Delta y_\nu \approx 2 \frac{\Delta y_{\rm m}}{m_{\rm m}} p_{\rm CM} \approx 1.34 p_{\rm CM} \sqrt{\frac{a \text{ MeV}}{m_{\rm m}}} \tag{1.28}$$

  For example, if the meson were a charged pion, the resolution would be $\Delta y_\nu \approx 3.4\sqrt{a}$, with $a \sim \mathcal{O}(1)$.

1.4.2.4 Leptons

The maximum momentum is similar to the meson case:

$$y_{\max} \approx \begin{cases} m_N a_{\max} & \text{if } \texttt{std = True} \\ 100 \text{ MeV} & \text{if } m_N < m_{\text{muon}} \\ 10 a_{\max} \text{ MeV} & \text{if } m_N - m_{\text{muon}} < 0.35 \text{ MeV} \\ a_{\max} \sqrt{m_N^2 - m_{\text{muon}}^2} & \text{otherwise} \end{cases} \tag{1.29}$$

with $a_{\max} = 10$ and the parameter `std` is set to `True` for electrons (default set to `False`).

For electrons the grid resolution is 0.125 MeV/step if the mass difference between HNL and muon is smaller than 10 MeV and 0.25 MeV/step otherwise. For muons the result is similar to Eq. 1.21:

$$\Delta y_{\text{meson}} = \text{Min} \Bigg[0.25 \text{ MeV}, \tag{1.30}$$

$$\sqrt{\left(a E_{\text{muon,CM}} + \frac{\Delta y_{\text{pion}}}{m_{\text{pion}}} p_{\text{CM}} \right)^2 - a^2 m_{\text{muon}}^2} - \sqrt{\left(a E_{\text{muon,CM}} - \frac{\Delta y_{\text{pion}}}{m_{\text{pion}}} p_{\text{CM}} \right)^2 - a^2 m_{\text{muon}}^2} \Bigg] , \tag{1.31}$$

where Δy_{pion} is taken from Eq. 1.25 and $a \sim \mathcal{O}(1)$.

1.4.3 Grid cut-offs

Using the particle grids such as they are when calculating collision integrals will be computationally costly. There are a few ways to speed up the simulation by introducing cut-offs in the grids.

1.4.3.1 Three-particle reactions

In general, three-particle collision integrals are computed quickly. The location of the peak of the collision integral is known for creation reactions; it lies between the momenta given in Eqs.

1.19 and 1.20 (in comoving coordinates):

$$y_{\text{lab,min}} = \sqrt{\left(\gamma\left(aE_{\text{CM}} - a\beta p_{\text{CM}}\right)\right)^2 - m^2a^2} \tag{1.32}$$

$$y_{\text{lab,max}} = \sqrt{\left(\gamma\left(aE_{\text{CM}} + a\beta p_{\text{CM}}\right)\right)^2 - m^2a^2} \tag{1.33}$$

$$\beta = \frac{y_X}{\sqrt{m_X^2a^2 + y_X^2}} \ , \tag{1.34}$$

where X is the decaying particle. Since species X has a momentum distribution, β is not necessarily the same in the equations above. There can be particles X with $\beta = p_{\text{CM}}/E_{\text{CM}}$, such that massive decay products can have $y_{\text{lab,min}} = 0$. This is roughly also true for massless decay products products at early times, when the mass term in 1.34 is negligible.
In the code, the collision integral is only computed for grid points between these two momenta. For all other points, it is evaluated to zero.

1.4.3.2 Four-particle reactions

Four-particle reactions take up most of the computation time. It is therefore important to reduce the number of grid points for which the four-particle collision integral must be computed. The basic idea used in the code is that HNL decay products are treated as in SBBN, as long as the HNL energy is below the default maximum grid momentum of 100 MeV. The minimum momentum is always 0 MeV.

- Creation reactions $1 + 2 + 3 \leftarrow 4$

$$y_{\text{max}} = \text{Max}\left[\ 100\ \text{MeV},\ \sqrt{\left(\sqrt{y_{4,\text{max}} + m_4^2a^2} - m_2a - m_3a\right)^2 - m_1^2a^2}\ \right] \ , \tag{1.35}$$

  where $y_{4,\text{max}}$ is the maximum momentum of the grid of particle 4.

- Scattering and decay reactions

  For HNL scattering and decay:

$$y_{\text{max}} = y_{\text{max,grid}} \tag{1.36}$$

  For scattering and decay of HNL decay product i:

$$y_{\text{max}} = \text{Max}\left[\ 100\ \text{MeV},\ \sqrt{\frac{y_{N,\text{max}}^2}{10^2} + m_N^2a^2 - m_i^2a^2}\ \right] \tag{1.37}$$

$y_{N,\max}$ is divided by 10, because HNLs are non-relativistic at temperatures of interest. If there are no HNLs involved in a simulation (such as SBBN simulation), then $y_{\max} = 100$ MeV.

Of course, once $y_{\max}$ exceeds the maximum momentum of the grid, the latter is used.

1.4.4 Interpolation of collision integral

A particle that participates in three-particle reactions is required to have a grid with high resolution. If the same particle also participates in four-particle reactions, then using the same grid points would be computationally expensive, since four-particle reactions require a relatively low grid resolution. For this purpose, an interpolation mechanism is implemented that takes a subset of the grid with a resolution of ∼0.25 MeV/step and computes the collision integral only for this subset. Then it uses linear interpolation to obtain the collision integral for all other points of the original grid. This mechanism also works in conjunction with the grid cut-offs discussed above. **Note**: Cubic interpolation does not give proper results for high-res grids.

1.4.5 Neglecting interactions

Collision integrals are not computed, when either one of the following conditions is met:

- If for at least one particle on both sides of the reaction $1 + 2 \leftrightarrow 3 + 4$

$$\frac{n}{n_{\text{initial}}} \leq 10^{-10} \left(\frac{a_{\text{initial}}}{a_{\max}} \right)^3 , \tag{1.38}$$

  with n the number density, n_{initial} the initial number density, a_{initial} the initial scale factor and $a_{\max} = 10$. The ratio of scale factor is to compensate for the expansion of the Universe. This condition can be satisfied when, for example, electrons and positrons annihilate, such that the rate of the reaction $\nu + e \leftrightarrow \nu + e$ becomes very low.

- If for scattering reactions involving HNLs the temperature drops below

$$T \leq \begin{cases} T_{\text{dec}}/15 & \text{relativistic decoupling} \\ 1 \text{ MeV} & \text{otherwise} \end{cases}$$

  where T_{dec} is the HNL decoupling temperature. This is done for all particles in such reactions, since the reaction rate $\Gamma_N \sim |\theta_\alpha|^2 \Gamma_\nu \leq \Gamma_\nu$. This means that scattering reactions involving HNLs decouple before neutrino decoupling. The validity of these conditions are discussed in Appendix subsection **??**.

- If in the creation reaction $1 + 2 + 3 \leftarrow 4$ particle 4 has already decayed. Particle 4 has decayed if the condition in Eq. 1.38 is satisfied. Here, this condition only holds if particle 4 is an HNL.
- If in the creation reaction $1 + 2 + 3 \leftarrow 4$ particle 4 is an unstable HNL decay product that has not been created yet. The number of particles created is tracked by a parameter called `num_creation`. This condition applies if, for example, particle 4 is a muon that decouples at $T = 4.6$ MeV, which is below the active neutrino decoupling temperature.
- If for HNL decay reactions the temperature is higher than the HNL mass (see **??**).

1.4.6 Multiple particles of same species

If there are particles of the same species involved in one side of a reaction, then a separate collision integral for each of these particles must be computed. However, since the collision integrals are the same, they can just be added and the result is an overall multiplicative factor. This scheme is only used when

- particle 4 in the reaction $1 + 2 + 3 \leftarrow 4$ is an HNL. For example, $\nu_e + \nu_e + \overline{\nu_e} \leftarrow N$ gives an overall factor of 3: two from this reaction and one from the charge conjugated channel.
- particle 4 in the reaction $1 + 2 + 3 \leftarrow 4$ is not a Majorana particle and particle 1, 2 or 3 is not charged. For example, both reactions $\pi^0 + \pi^0 + \pi^0 \leftarrow \eta$ and $\pi^+ + \pi^+ + \pi^- \leftarrow K^+$ give an overall factor of 3 for the collision integral of the pions, while $\pi^+ + \pi^- + \pi^0 \leftarrow \eta$ does not, because there is no charge conjugated channel.
- particle 3 in the reaction $1 + 2 \leftarrow 3$ is not a Majorana particle and particle 1 or 2 is not charged. For example, the reaction $\pi^0 + \pi^0 \leftarrow K_S^0$ gives a factor of 2 while $\pi^+ + \pi^- \leftarrow K_S^0$ does not.

1.4.7 Collision integral kind

Each reaction comes with a collision integral that has a specific form of the functional $F[f]$ (Eq. **??**), which is governed by a parameter called `kind`. It can have the following values: `Full`, `F_1`, `F_f`, `F_creation`, `F_decay`, `F_1_vacuum_decay` and `F_f_vacuum_decay`. The different values of this parameter will be explained through examples.

- Reaction $1 + 2 \leftrightarrow 3 + 4$ with

$$F[f] = F_{\text{full}} = f_3 f_4 (1 - f_1)(1 - f_2) - f_1 f_2 (1 - f_3)(1 - f_4) \tag{1.39}$$
$$= F_{\text{creation}} + f_1 F_{\text{decay}}$$
$$F_{\text{creation}} = f_3 f_4 (1 - f_1)(1 - f_2) \tag{1.40}$$
$$F_{\text{decay}} = -f_2 (1 - f_3)(1 - f_4) \tag{1.41}$$

  which can also be rewritten as

$$F[f] = F_{\text{full}} = f_3 f_4 (1 - f_2) - f_1 \left[f_3 f_4 (1 - f_2) + f_2 (1 - f_3)(1 - f_4) \right] \tag{1.42}$$
$$= F_1 + f_1 F_f$$
$$F_1 = f_3 f_4 (1 - f_2) \tag{1.43}$$
$$F_f = -\left[f_3 f_4 (1 - f_2) + f_2 (1 - f_3)(1 - f_4) \right] \tag{1.44}$$

  Thus, `Full` returns Eq. 1.39, `F_creation` returns Eq. 1.40 and `F_decay` returns Eq. 1.41, `F_1` returns Eq. 1.43 and `F_f` returns Eq. 1.44.

- Reaction $1 \rightarrow 2 + 3 + 4$ with

$$F[f] = -f_1 (1 - f_2)(1 - f_3)(1 - f_4) \tag{1.45}$$
$$= f_1 F_{\text{decay}}$$
$$F_{\text{decay}} = -(1 - f_2)(1 - f_3)(1 - f_4) \tag{1.46}$$

  In vacuum the Fermi blocking factors can be neglected.

$$F[f] = -f_1 = f_1 F_{\text{vacuum,decay}} \tag{1.47}$$
$$F_{\text{vacuum,decay}} = -1 \tag{1.48}$$

  This is from the point of view of particle 1. From the point of view of particle 4 one has

$$F[f] = f_1 (1 - f_2)(1 - f_3)(1 - f_4) = F_{\text{creation}} \tag{1.49}$$

  In vacuum the Fermi blocking factors can again be neglected.

$$F_{\text{vacuum,decay}} = f_1 \tag{1.50}$$

  Thus, `F_f_vacuum_decay` returns Eq. 1.48 and `F_1_vacuum_decay` returns Eq. 1.50.

Collision integrals with `kind` set to `F_decay`, `F_f_vacuum_decay` or `F_f` will be multiplied by the distribution function later on in the code (it is not done so already when the collision integral itself is computed).
Collision integrals for unstable HNL decay products have their `kind` parameter set to either `F_decay` (default) or `F_f_vacuum_decay`.

1.4.8 Collision integral bounds

The bounds of collision integrals are rather complicated. The main idea is that energy-momentum conservation is used to obtain these bounds whenever possible. In most cases, however, using only energy-conservation is the way to go, to prevent everything from becoming an even bigger mess. This is not really a big deal; it just means that the region integrated is larger than the one allowed by energy-momentum conservation. Of course, outside this latter region the integrand is evaluated to zero. Most bounds are obtained by using Mathematica and will therefore not be derived, merely stated. In what follows: $\widetilde{m} = ma$.

1.4.8.1 Three-particle reactions

- Creation reactions $1 + 2 \leftrightarrow 3$

  If $y_1 = 0$:

$$y_2 = \sqrt{\frac{\left(\widetilde{m}_3^2 - \widetilde{m}_1^2 - \widetilde{m}_2^2\right)^2 - 4\widetilde{m}_1^2\widetilde{m}_2^2}{4\widetilde{m}_3^2}} \tag{1.51}$$

  If $m_1 = 0$:

$$A = \frac{\widetilde{m}_2^4 + \widetilde{m}_3^4 - 2\widetilde{m}_2^2\left(\widetilde{m}_3^2 + 2y_1^2\right)}{4y_1\left(\widetilde{m}_2^2 - \widetilde{m}_3^2\right)}$$

$$y_{2,\min} = \mathrm{Max}\left[A, -A\right] \tag{1.52}$$

$$y_{2,\max} = \sqrt{\left(\sqrt{y_{3,\max}^2 + \widetilde{m}_3^2} - \sqrt{y_1^2 + \widetilde{m}_1^2}\right)^2 - \widetilde{m}_2^2}\ , \tag{1.53}$$

  where $y_{3,\max}$ is the maximum momentum of the grid of particle 3 (to make the system consistent).

  If $m_1 \neq 0$:

$$B = \frac{-y_1\left(\widetilde{m}_3^2 - \widetilde{m}_1^2 - \widetilde{m}_2^2\right) + \sqrt{\left(y_1^2 + \widetilde{m}_1^2\right)\left(\widetilde{m}_2^4 + \left(\widetilde{m}_3^2 - \widetilde{m}_1^2\right)^2 - 2\widetilde{m}_2^2\left(\widetilde{m}_1^2 + \widetilde{m}_3^2\right)\right)}}{2\widetilde{m}_1^2}$$

$$C = \frac{y_1\left(\widetilde{m}_3^2 - \widetilde{m}_1^2 - \widetilde{m}_2^2\right) + \sqrt{\left(y_1^2 + \widetilde{m}_1^2\right)\left(\widetilde{m}_2^4 + \left(\widetilde{m}_3^2 - \widetilde{m}_1^2\right)^2 - 2\widetilde{m}_2^2\left(\widetilde{m}_1^2 + \widetilde{m}_3^2\right)\right)}}{2\widetilde{m}_1^2}$$

$$y_{2,\min} = \mathrm{Max}\left[B, -B\right] \tag{1.54}$$

$$y_{2,\max} = \mathrm{Min}\left[C,\ \sqrt{\left(\sqrt{y_{3,\max}^2 + \widetilde{m}_3^2} - \sqrt{y_1^2 + \widetilde{m}_1^2}\right)^2 - \widetilde{m}_2^2}\right] \tag{1.55}$$

- Decay reactions $1 + 2 \leftrightarrow 3$

  If $y_1 = 0$, Eq. 1.51 is used. Else

$$y_{2,\min} = \mathrm{Max}\left[C, -C\right] \tag{1.56}$$

$$y_{2,\max} = B \tag{1.57}$$

1.4.8.2 Four-particle reactions

- Decay reactions $1 \leftrightarrow 2 + 3 + 4$

$$y_{2,\min} = 0 \tag{1.58}$$

$$y_{2,\max} = \sqrt{\left(\sqrt{y_1^2 + \widetilde{m}_1^2} - \widetilde{m}_3 - \widetilde{m}_4\right)^2 - \widetilde{m}_2^2} \tag{1.59}$$

$$y_{3,\min} = 0 \tag{1.60}$$

$$y_{3,\max} = \sqrt{\left(\sqrt{y_1^2 + \widetilde{m}_1^2} - \sqrt{y_2^2 + \widetilde{m}_2^2} - \widetilde{m}_4\right)^2 - \widetilde{m}_3^2}\ , \tag{1.61}$$

  where y_2 lies between $y_{2,\min}$ and $y_{2,\max}$.

- Creation reactions $1 + 2 + 3 \leftrightarrow 4$

$$y_{2,\min} = 0 \tag{1.62}$$

$$y_{2,\max} = \sqrt{\left(\sqrt{y_{4,\max}^2 + \widetilde{m}_4^2} - \sqrt{y_1^2 + \widetilde{m}_1^2} - \widetilde{m}_3\right)^2 - \widetilde{m}_2^2} \tag{1.63}$$

$$y_{3,\min} = \sqrt{\left(\widetilde{m}_4 - \sqrt{y_1^2 + \widetilde{m}_1^2} - \sqrt{y_2^2 + \widetilde{m}_2^2}\right)^2 - \widetilde{m}_3^2} \tag{1.64}$$

$$y_{3,\max} = \sqrt{\left(\sqrt{y_{4,\max}^2 + \widetilde{m}_4^2} - \sqrt{y_1^2 + \widetilde{m}_1^2} - \sqrt{y_2^2 + \widetilde{m}_2^2}\right)^2 - \widetilde{m}_3^2}\ , \tag{1.65}$$

  where y_2 lies between $y_{2,\min}$ and $y_{2,\max}$ and $y_{4,\max}$ is the maximum momentum of the grid of particle 4 (to make the system consistent). Equation 1.64 only holds if $\widetilde{m}_4 \geq \widetilde{E}_1 + \widetilde{E}_2 + \widetilde{m}_3$, otherwise it is 0.

- Scattering reactions $1 + 2 \leftrightarrow 3 + 4$

$$y_{2,\min} = \sqrt{\left(\widetilde{m}_3 + \widetilde{m}_4 - \sqrt{y_1^2 + \widetilde{m}_1^2}\right)^2 - \widetilde{m}_2^2} \tag{1.66}$$

$$y_{2,\max} = \mathrm{Max}\left[y_{2,\mathrm{grid,max}},\ 3 \cdot y_{2,\min}\right] \ , \tag{1.67}$$

where $y_{2,\mathrm{grid,max}}$ is the maximum momentum of the grid of particle 2. Equation 1.66 only holds if $\widetilde{m}_3 + \widetilde{m}_4 \geq \widetilde{E}_1 + \widetilde{m}_2$, otherwise it is 0. Equation 1.67 is based on trail-and-error, which explains it unconventional form.

If $y_1 \neq 0$:

$$y_{3,\min} = 0 \tag{1.68}$$

$$y_{3,\min} = \sqrt{\left(\sqrt{y_1^2 + \widetilde{m}_1^2} + \sqrt{y_2^2 + \widetilde{m}_2^2} - \widetilde{m}_4\right)^2 - \widetilde{m}_3^2} \tag{1.69}$$

If $y_1 = 0$, the bounds are derived from both energy and momentum conservation instead of energy conservation only. The discussion of these bounds is skipped, since they are convoluted and only used for one grid point.

1.4.9 The 5-step computation scheme

Each simulation step consists of five parts:

1. The current regime of each particle is determined and its parameter set is updated with the latest cosmological and thermodynamical variables, such as temperature, energy density and pressure.

2. The interactions are initialized, which means that if a particle is decoupled, the particles' `collision_integrals` array is loaded with currently active 'integral' objects. Each such object holds the following information: the integral type (three- or four-particle collision integral), for which particle this integral is meant, the reaction of interest (each element of the 'reaction' object contains the corresponding particle, the side and index of the particle in the reaction and whether it is actually a particle or anti-particle), the matrix element of the reaction and the collision integral kind. In every 'reaction' object (list) the first entry is the particle for which the collision integral will be calculated.

3. The collision integrals are computed. When particles are added to the `Universe` class, they will be sorted by descending mass. The computation of the collision integrals follows the same order.

4. The distribution functions are updated, using the Adams-Moulton implicit method of order 5. Neutrino oscillations − if enabled − will be taken into account here.

5. The terms in the temperature evolution equation are calculated and the temperature along with a number of other cosmological variables are updated.

1.4.10 Unstable HNL decay products

SM particles that have a very small lifetime follow a more involved routine after they decouple. There is a parameter `fast_decay` that is set to `True` for these particles. Their distribution function is divided in two parts:

- Background distribution given by Eq. 1.4. Particles for which `thermal_dyn` is set to `False` have a distribution that is just zero.
- Contribution from HNL decay and other meson decays.

Some of the unstable particles will first thermalize with the plasma before they decay (muons, charged pions and so on). These particles are identified with the parameter `thermalization`. There are three stages at which the distribution function must be computed:

1. Before creation of particle. The distribution is just the background distribution.

2. After creation of particle. The distribution function is given by

$$f_{\text{creation}} = f_{\text{background}} + I_{\text{creation}} \Delta t \tag{1.70}$$

3. (Only if `thermalization = True`) After thermalization of particle. The distribution is given by

$$f_{\text{thermalization}} = f_{\text{background}} + \frac{\Delta n}{(2\pi \widetilde{m} a T)^{\frac{3}{2}}} e^{-\frac{y^2}{2\widetilde{m} a T}} \tag{1.71}$$

The second term on the right-hand side is obtained by

$$\int \mathrm{d}^3 y I_{\text{creation}} = \Delta n = \int \mathrm{d}^3 y e^{-\frac{m-\mu}{T}} e^{-\frac{y^2}{2\widetilde{m}aT}} = (2\pi\widetilde{m}aT)^{\frac{3}{2}} e^{-\frac{m-\mu}{T}}$$
$$f_{\text{equilibrium}} = e^{-\frac{m-\mu}{T}} e^{-\frac{y^2}{2\widetilde{m}aT}} = \frac{\Delta n}{(2\pi\widetilde{m}aT)^{\frac{3}{2}}} e^{-\frac{y^2}{2\widetilde{m}aT}} \tag{1.72}$$

Notice that the background distribution is omitted when Δn is computed, which is justified due to the strong exponential decay of the background distribution.

For each reaction in which the decaying particle is a muon or meson, the dynamical equilibrium mechanism is applied to compute the collision integral I_{creation} (also for active neutrinos). Consider the reaction

$$N \xrightarrow{(1)} \pi^+ \xrightarrow{(2)} \mu^+ \xrightarrow{(3)} \nu$$

The number of charged pions created during decay (1) is known, since that collision integral is just a standard one. This value Δn is then stored in a parameter called `num_creation`. This is also the same amount of charged pions that decay into muons (times the branching ratio). Recall that the collision integrals are computed for particles in order of descending mass. So, when it is the muon's turn in (2), the code knows how many muons should be created in this reaction. It then introduces a scaling factor α in Eq. 1.70 such that

$$\alpha \int I_{\text{creation}} \Delta t = \Delta n \cdot \text{BR} \Longrightarrow \alpha = \frac{\Delta n \cdot \text{BR}}{\int I_{\text{creation}} \Delta t} \tag{1.73}$$

where BR is the branching ratio of the decay/creation channel. Differences in degrees of freedom (collision integral is divided by the d.o.f.) and any other additional factors are already taken into account here. The distribution function after creation will therefore be

$$f_{\text{creation}} = f_{\text{background}} + \alpha I_{\text{creation}} \Delta t \tag{1.74}$$

The piece of code that computes the scaling α for creation reactions can be seen below. An almost identical procedure is followed for decay reactions of unstable particles to obtain I_{decay}. The total collision integral is equal to

$$I_{\text{total}} = \begin{cases} I_{\text{creation}} + \frac{f_{\text{thermalization}} - f_{\text{creation}}}{\Delta t} + I_{\text{decay}} & \text{if } \texttt{thermalization = True} \\ I_{\text{creation}} + I_{\text{decay}} & \text{otherwise} \end{cases} \tag{1.75}$$

```
if reaction_type.type(self.reaction, reaction_type.CREATION) and hasattr(self.reaction[-1].
    specie, 'fast_decay'):
  # Join symbols of created particles together in order to choose the right branching ratio
  sym = ''.join([item.specie.symbol for item in self.reaction[:-1]])
  for key in self.reaction[-1].specie.BR:
    if Counter(sym) == Counter(key):
      BR = self.reaction[-1].specie.BR[key]

  # Number of created particles ('proportional' is more accurate)
  # 'integral' is divided by dof before, which is why it must be multiplied by dof here
  dof = self.particle.dof if self.particle.majorana else self.particle.dof / 2
  created = simps(integral * constant * dof * grid.TEMPLATE**2, grid.TEMPLATE)

  # If there are no particles created through this channel, return zero
  if created == 0.:
    if self.kind in [CollisionIntegralKind.Full, CollisionIntegralKind.Full_vacuum_decay]\
    or hasattr(self.particle, 'fast_decay'):
      return numpy.zeros(len(integral)), numpy.zeros(len(integral))
    return numpy.zeros(len(integral))

  # num_creation does not depend on the d.o.f. of the decaying particle
  scaling = BR * self.reaction[-1].specie.num_creation / created
  integral *= scaling
```

1.4.11 QCD phase transition

The QCD phase transition is approximated as an instantaneous event that happens at $T = \Lambda_{\rm QCD}$. Current value of the phase transition temperature is set to $\Lambda_{\rm QCD} = 150$ MeV. The phase transition sequence is a follows:

1. The last stored values of cosmological parameters are removed from the data array, since they correspond to a temperature below $\Lambda_{\rm QCD}$.

2. The entropic degrees of freedom of relativistic particles before the phase transition is computed by

$$g_{s,\rm before} = \sum_{\rm bosons} g_b + \frac{7}{8} \sum_{\rm fermions} g_f \tag{1.76}$$

   Only particles in equilibrium are taken into account here, since entropy is separately conserved for the thermal bath and the decoupled species.

3. Quarks and gluons are removed from the system and replaced by hadrons (in this case mesons).

4. The entropic degrees of freedom of relativistic particles after phase transition is computed accordingly to Eq. 1.76.

5. The temperature is assumed to be unaffected by the phase transition. This means that alongside aT, the scale factor a and the dimensionful scale factor x change in the same way:

$$(aT)_{\text{after}} = (aT)_{\text{before}} \left(\frac{g_{s,\text{before}}}{g_{s,\text{after}}} \right)^{\frac{1}{3}} \tag{1.77}$$

$$a_{\text{after}} = a_{\text{before}} \left(\frac{g_{s,\text{before}}}{g_{s,\text{after}}} \right)^{\frac{1}{3}} \tag{1.78}$$

$$x_{\text{after}} = x_{\text{before}} \left(\frac{g_{s,\text{before}}}{g_{s,\text{after}}} \right)^{\frac{1}{3}} \tag{1.79}$$

6. All other cosmological variables, like Hubble parameter and total energy density, are updated and the temperature is set equal to $T = \Lambda_{\text{QCD}}$.

7. All particles are updated to the latest parameters.

8. Quarkic interactions are removed from the system and replaced by hadronic interactions and, if applicable, their subsequent interactions.

1.4.12 Additional modification to the modified KAWANO code

In the modified KAWANO code there is a variable called hv that stores the ratio of the baryon energy density to T^3:

$$hv = \frac{\rho_b}{T^3} \approx \frac{M_u n_b}{T^3} = \frac{\eta n_\gamma M_u}{T^3} = 3.3683 \cdot 10^4 \eta \ \text{ grams} \,, \tag{1.80}$$

where M_u is the atomic mass unit in grams. The initial value of this quantity is written in terms of the final value of the baryon-to-photon ratio ($\eta_{\text{final}} \sim 6 \cdot 10^{-10}$):

$$(hv)_{\text{ini}} = 3.3683 \cdot 10^4 \eta_{\text{ini}} = 3.3683 \cdot 10^4 \eta_{\text{final}} \frac{\eta_{\text{ini}}}{\eta_{\text{final}}} \tag{1.81}$$

In SBBN:

$$\frac{\eta_{\text{ini}}}{\eta_{\text{final}}} = \frac{\frac{n_{b,\text{ini}}}{n_{\gamma,\text{ini}}} \frac{a_{\text{ini}}^3}{a_{\text{ini}}^3}}{\frac{n_{b,\text{final}}}{n_{\gamma,\text{final}}} \frac{a_{\text{final}}^3}{a_{\text{final}}^3}} = \frac{n_{\gamma,\text{final}} a_{\text{final}}^3}{n_{\gamma,\text{ini}} a_{\text{ini}}^3} = \left(\frac{(aT)_{\text{final}}}{(aT)_{\text{ini}}} \right)^3 = \frac{11}{4} = 2.75 \tag{1.82}$$

And therefore

$$(hv)_{\text{ini}} = 3.3683 \cdot 10^4 \cdot \eta_{\text{final}} \cdot 2.75 \tag{1.83}$$

The evolution of hv would then be computed in KAWANO as in SBBN. This is what happens in the previous version of the modified KAWANO code.
However, in this way KAWANO does not account for the fact that HNLs can heat up the plasma when they decay and change the final value of aT. The evolution of hv can therefore be different than that in SBBN. So, the line in Eq. 1.83 is modified to

$$hv = 3.3683 \cdot 10^4 \cdot \eta_{\text{final}} \cdot \left(\frac{(aT)}{(aT)_{\text{ini}}} \right)^3 , \tag{1.84}$$

where the aT's are computed in pyBBN and passed on to KAWANO.

An additional perk that comes with the updated KAWANO is that the final baryon-to-photon ratio η does not need to be adjusted manually anymore to give the correct final value $\eta_{\text{final}} \sim 6{\cdot}10^{-10}$. It all happens automatically now.

Bibliography

[1] O. Ruchayskiy and A. Ivashko, *Restrictions on the lifetime of sterile neutrinos from primordial nucleosynthesis*, JCAP **1210** (2012) 014, [1202.2841].