Tests of the `tTcut` method

December 22, 2021

Contents

1 Model 1

The first model that I have implemented is the Standard Model with a single generation of leptons (electron + neutrino) and no QCD. I will first try to describe my implementation step-by-step, including the complications that I encountered. I will add a few comments here and there, some of which you may find to be irrelevant/trivial, but which are useful at least for my understanding:

- The zero-temperature implementation has been performed through `FeynRules`. The electron and the neutrino have been declared with tildes, in order to trick MO into considering them as Z_2-odd particles.
- For the moment, I have only implemented the SM in its broken phase. The computations that follow are performed using the broken SM, even if the effects that we're studying are taking place at temperatures which are higher than the EW phase transition temperature. This is to be discussed (eventually).
- The implementation has been performed in Unitary gauge. I can change this quite easily, if we think it's useful.
- I have chosen as free parameters: g_1, g_2, v, λ and y_e, *i.e.* the two gauge couplings, the Higgs vev, the Higgs quartic coupling and the electron Yukawa coupling. This is the only way that I could think of in order to incorporate, eventually, temperature-dependent masses through the vev. This means that I had to rewrite the SM implementation, which is why things took a bit longer than I initially expected. The values of these parameters (except for v) have been taken to be equal to their values at m_Z.
- For some weird reason, `FeynRules` fails to detect that diagonalisations have been done correctly. This means that I get extra vertices which, however, *are* zero (I verified this by hand by taking the expressions of the coefficients of the "fishy" terms, basically for some reason `FeynRules` fails to detect that $g_2 s_w - g_1 c_w = 0$). Once I generated `CalcHEP` model files, I removed these extra vertices by hand. I hope the remaining vertices are not affected (I'm not very good at reading `CalcHEP` vertex syntax).

- One issue was that, given the particle content, the neutrino (the "dark matter candidate") remains massless (which would mean that Ωh^2 makes no sense). In order to circumvent this, and following Sasha's suggestion, I gave the neutrino a fake mass of 1 keV (in the `CalcHEP` model file)[1] [2] [3].
- Following our discussions, I also fixed the electron mass to 511 keV. I don't believe this matters at all, but I did.
- From the initial tests we came to realise that the dominant contribution comes from Z decays (or, eventually, annihilations mediated by a Z boson). In order to eliminate these contributions, I deleted the Zee and $Z\nu\nu$ vertices from the model by hand (I just removed the vertices from `lgrng1.mdl`).

The relevant masses have been defined as:

$$\begin{aligned} m_Z &= \frac{1}{2}\sqrt{g_1^2 + g_2^2}v \\ m_W &= \frac{1}{2}g_2 v \\ m_h &= \sqrt{2\lambda}v \\ \mu &= \sqrt{\lambda v^2} \end{aligned} \tag{1}$$

In passing, I also found it useful to define $s_w^2 = g_1^2/(g_1^2 + g_2^2)$.

The first step was to try and reproduce the naive behaviour by running MO as it stands, with no temperature-dependent masses and no cut. In order to isolate the contribution, I employed

```
omegaFi22 = darkOmegaFi22(TR, "~e+, ~e- -> ~ve, ~NE", 0, 1, & err);
```

The result I obtain is

$$\Omega h^2\big|_{\mathrm{bare}} = 3.390 \times 10^{15} \tag{2}$$

Yes, it's a ridiculous value, as we would expect when we try to do freeze-in with electroweak-order couplings. The absolute value is irrelevant, what I'm interested in are the relative values among different methods.

Next, I played around with the `tTcut` method by changing `tTcut` in `freezein.c` (default value was zero). The results I obtained are

$$\begin{aligned} \Omega h^2\big|_{\mathrm{tTcut}=1} &= 2.415 \times 10^{12} \\ \Omega h^2\big|_{\mathrm{tTcut}=0.77} &= 3.722 \times 10^{12} \\ \Omega h^2\big|_{\mathrm{tTcut}=0.62} &= 5.154 \times 10^{12} \end{aligned} \tag{3}$$

[1] Comment: This means that if the common lore is correct, *i.e.* that in IR-dominated freeze-in through scattering dark matter production takes place mostly around $T \sim m_{\mathrm{DM}}/3$, most of dark matter should be produced at a temperature around 0.3 keV.

[2] Comment: Could this non-gauge-invariant mass cause any trouble *in this example*?

[3] Comment: Sasha is not responsible if the fake mass makes no sense, his proposal was to assign "a small mass". I randomly chose 1 keV. Sasha also proposed me another solution in order to just calculate the yield Y (apparently, there's a relevant function in `freezein.c`). I chose to work with a publicly available routine.

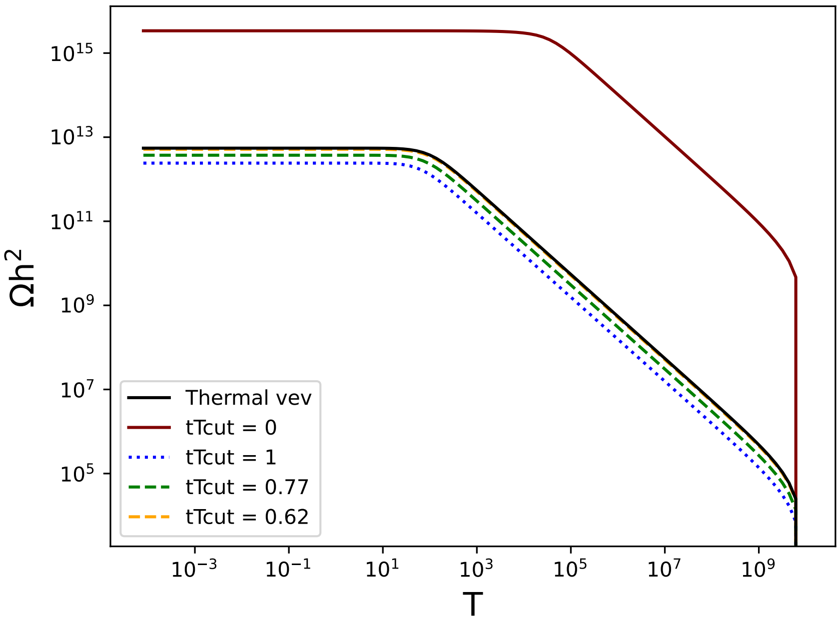

Figure 1: Comparison of the T-dependence of different methods, as described in the text.

The value 0.62 for `tTcut` will be explained shortly, whereas the 0.77 one is kept for comparison. These values are ~ 1 order of magnitude smaller than the ones which I obtained in the version of the model including the Z boson, which demonstrates that most of the effects in my initial tests came from this channel. In passing, I note that the "no T-dependence, no cut, including the Z" results were slightly larger (but within the same order-of-magnitude) than what I got in (2) (concretely, I had obtained $\Omega h^2\big|_{\text{bare}} = 3.390 \times 10^{15}$), which shows that if no corrections were applied, the t-channel contribution got comparable to the s-channel one. The t-channel results agree more or less, at least within orders of magnitude, and it seems that something has definitely changed wrt the most "naive" approach[4].

My last test was to incorporate a temperature-dependent vev, which also induces temperature-dependent masses for the gauge bosons (and, for that matter, the Higgs - which does not enter in any place). To this goal, I worked in the following way:

- Define the T-dependent part of the vev, $v^T(T)$ through the T-dependent part of the W boson thermal mass, $m_W^T(T)$ as

$$v^T(T) = \frac{2}{g_2} m_W^T(T) \tag{4}$$

[4]Comment: The way I did things, every time I wanted to change `tTcut` I did that in `freezein.c`, then re-made my `main.c`. Perhaps we should consider including `tTcut` as a parameter in `main.c`.

The latter is given by

$$m_W^T{}^2(T) = \frac{1}{6}g_2^2T^2\left(N + N_S + N_F/2\right) = \frac{11}{12}g_2^2T^2 \tag{5}$$

$$\Rightarrow v^T(T) = 2\sqrt{\frac{11}{12}}T \tag{6}$$

- Then, define the total "thermal vev" as:

$$v(T) = \sqrt{v^2 + v^{T\ 2}(T)} \tag{7}$$

- Finally, we write the full W boson mass as

$$m_W(T) = \frac{1}{2}g_2 v(T) \tag{8}$$

and similarly for the other particles.

In order to implement this method, I edited `vars1.mdl` and `func1.mdl`: I removed `vev` from the free parameters and I instead added an entry in `func1.mdl` as

$$v(T) = \sqrt{246^2 + \left(2\sqrt{\frac{11}{12}}T\right)^2} \tag{9}$$

I used this expression following our discussions. This is where the choice `tTcut`$= 0.62$ comes from, by the way: I used the coefficient of the T-dependent part of m_W, i.e. the coefficient of $m_W(T)$, which is $\sqrt{\frac{11}{12}}g_2 = 0.62$. A more complete calculation should combine the broken and the unbroken SM phases. As I've written before, I believe that with MO as it stands it is possible to combine the two contributions and I'd like to try this one out too. In any case, through this method I obtain

$$\Omega h^2\big|_{\rm tTcut=0,\ T-dependent\ vev,\ single\ phase} = 5.480 \times 10^{12} \tag{10}$$

i.e. within 10% wrt the tTcut method. As far as I'm concerned, this is good enough (if we're aiming for a 1% accuracy ok, let's discuss it, on my side I'm ok with a 10% one in this context).

In Figure 1 I compare the T-evolution of Ωh^2 between the different methods: no-cut-no-thermal-vev, thermal vev and two values of `tTcut`. In all cases, the "reheating temperature" (*i.e.* the temperature at which we start counting) has been set to 10^{10} GeV. In all cases, I obtained a `kin22 problem` warning.

As a last test for this model, I wanted to see if our results are now insensitive to the reheating temperature. The results I obtained are the following

- Through the method of the T-dependent vev

$$\begin{aligned} \Omega h^2\big|_{\rm T-dependent\ vev,\ T_R=10^{10}GeV} &= 5.480 \times 10^{12} \\ \Omega h^2\big|_{\rm T-dependent\ vev,\ T_R=10^{5}GeV} &= 5.475 \times 10^{12} \\ \Omega h^2\big|_{\rm T-dependent\ vev,\ T_R=10^{3}GeV} &= 4.937 \times 10^{12} \\ \Omega h^2\big|_{\rm T-dependent\ vev,\ T_R=10^{2}GeV} &= 1.724 \times 10^{12} \end{aligned} \tag{11}$$

- Through the `tTcut` method

$$\Omega h^2\big|_{\rm tTcut=0.62,\ T_R=10^{10}GeV} = 5.154\times 10^{12} \tag{12}$$
$$\Omega h^2\big|_{\rm tTcut=0.62,\ T_R=10^{5}GeV} = 5.149\times 10^{12}$$
$$\Omega h^2\big|_{\rm tTcut=0.62,\ T_R=10^{3}GeV} = 4.645\times 10^{12}$$
$$\Omega h^2\big|_{\rm tTcut=0.62,\ T_R=10^{2}GeV} = 1.629\times 10^{12}$$

i.e. all in good agreement and all fairly insensitive to T_R, unless the latter starts approaching m_W. For comparison, the results that I obtain using no T-dependence and no `tTcut` read as follows

$$\Omega h^2\big|_{\rm tTcut=0.,\ T_R=10^{10}GeV} = 3.390\times 10^{15} \tag{13}$$
$$\Omega h^2\big|_{\rm tTcut=0.,\ T_R=10^{5}GeV} = 2.421\times 10^{15}$$
$$\Omega h^2\big|_{\rm tTcut=0.,\ T_R=10^{3}GeV} = 3.623\times 10^{13}$$
$$\Omega h^2\big|_{\rm tTcut=0.,\ T_R=10^{2}GeV} = 2.243\times 10^{12}$$

i.e. they vary much more dramatically.

In order for everyone to be able to check/reproduce all this, I'm adding to the `wiki` the following model files:

- The `FeynRules` implementation (which includes the "fishy" terms).
- The zero-temperature `CalcHEP` model files (used to obtain results with the `tTcut` method (`SMforUVbehaviourNoThermal.zip`).
- The `CalcHEP` model files with a T-dependent vev (`SMforUVbehaviour.zip`).
- The zero-temperature `CalcHEP` model files (used to obtain results with the `tTcut` method without the Z boson (`SMforUVbehaviourNoThermalNoZ.zip`).
- The `CalcHEP` model files with a T-dependent vev without the Z boson (`SMforUVbehaviourNoZ.zip`).

Note that in order to use these different implementations, you have to create different `micrOMEGAs` projects.

References