

Equations of Motion for Dark Matter–Dark Energy Interactions

Using xPand for choosing a metric and the Newtonian gauge

In this notebook we derive all equations to 0th and 1st order of dark energy in the form of a scalar field ϕ that interacts with dark matter with mass $m = m(\phi)$. The equations are implemented in the Cosmic Linear Anisotropy Solving System (CLASS) in the Newtonian gauge.

See <https://github.com/kabeleh/iDM/> for the implementation into the CLASS source code. In particular, the comprehensive GitHub Wiki explains technical details.

General Setup

Mathematica 14.3 introduced commands that are already assigned in xAct. The following lines allow xAct to use its own definition.

```
In[1]:= Unprotect [ Commutator, Anticommutator ] ;
```

```
ClearAll [ Commutator, Anticommutator ] ;
```

```
Remove [ Commutator, Anticommutator ] ;
```

```
In[4]:= << xAct`xPand`
```

xPand.nb

Package xAct`xPerm` version 1.2.3, {2015, 8, 23}
Copyright (C) 2003–2020, Jose M. Martin–Garcia, under the General Public License.
Connecting to external linux executable...
Connection established.

Package xAct`xTensor` version 1.2.0, {2021, 10, 17}
Copyright (C) 2002–2021, Jose M. Martin–Garcia, under the General Public License.

Package xAct`xPert` version 1.0.6, {2018, 2, 28}
Copyright (C) 2005–2020, David Brizuela, Jose M. Martin–Garcia and Guillermo A. Mena Marugan, under the General Public License.
** Variable \$PrePrint assigned value ScreenDollarIndices
** Variable \$CovDFormat changed from Prefix to Postfix
** Option AllowUpperDerivatives of ContractMetric changed from False to True
** Option MetricOn of MakeRule changed from None to All
** Option ContractMetrics of MakeRule changed from False to True

Package xAct`xPand` version 0.4.4, {2025, 4, 1}
Copyright (C) 2012–2025, Cyril Pitrou, Xavier Roy and Obinna Umeh under the General Public License.

These packages come with ABSOLUTELY NO WARRANTY; for details type Disclaimer []. This is free software,
and you are welcome to redistribute it under certain conditions. See the General Public License for details.

```
In[5]:= SetOptions [ AutomaticRules, Verbose → False ] ;  
$DefInfoQ = False;
```

```
In[7]:= $ConformalTime = True; ( * To match CLASS notation * )  
$FirstOrderVectorPerturbations = True; ( *Default=True* )  
$FirstOrderTensorPerturbations = True; ( *Default=True* )
```

```
In[10]:= ( * The first order perturbation ( of the metric ) is the total perturbation. Therefore we need  
to put higher order perturbations of the metric to 0, but also the spatial part of the velocity.* )  
BackgroundFieldMethod = True;
```

Creation of Manifold and Metric

We choose a 4-dimensional manifold M , define a 4d metric $g_{\alpha\beta}$, the covariant derivative ∇ with respect to this metric, as well as a flat Friedmann–Lemaître Spacetime with metric-perturbation h :

```

DefManifold [ M, 4 ( *d* ), {  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\mu$ ,  $\nu$ ,  $\lambda$ ,  $\sigma$  } ];

DefMetric [ -1, g [ - $\alpha$ , - $\beta$  ], CD, { ";", " $\overline{\nabla}$ " }, PrintAs  $\rightarrow$  " $\overline{g}^{(4D)}$ " ];

SetSlicing [ g, n, h, cd, { " | ", " $\overline{D}$ " }, "FLFlat" ];

** MakeRule: Potential problems moving indices on the LHS.

```

Definition of Auxiliary Functions

To facilitate the computations, we define some functions to express the results in a more human-readable form.

In[14]:=

```

org [ expr_ ] := NoScalar@Collect [ ContractMetric [ expr ], $PerturbationParameter, ToCanonical ]

collect [ expr_ ] := NoScalar@Collect [ expr, $PerturbationParameter, Identity ]

```

In[16]:=

```

MyxPand [ expr_, gauge_, order_ ] := ToxPand [ expr, dg, u, du, h, gauge, order ]

```

In[17]:=

```

( *

MyxPand does the following:

    expr

    Conformal [ g,gah2 ] [ % ]

    ExpandPerturbation@Perturbed [ %,1 ] / / org ( *1 is the order* )

    SplitPerturbations [ %,SplitMetric [ g,dg,h,"AnyGauge" ],h ] ( *"AnyGauge" is the chosen gauge* )

* )

```

Definition of Perturbations

We set g to be the metric to be used and h to be its perturbation.

In[18]:=

```

DefMetricFields [ g, dg, h ]

```

Equations of Motion

Definition of fields and functions:

In[19]:=

```

DefProjectedTensor [ sf [ ], h, PrintAs  $\rightarrow$  " $\Phi$ " ] ( * The scalar field  $\phi$  * )

DefProjectedTensor [ nf [ ], h, PrintAs  $\rightarrow$  "n" ] ( * The dark matter particle number * )

```

In[21]:=

```

DefScalarFunction [ V ] ( * The scalar field  $V(\phi)$  * )

DefScalarFunction [ m ] ( * The dark matter mass  $m(\phi)$  * )

```

In[23]:=

```

DefConstantSymbol [ massP, PrintAs  $\rightarrow$  " $M_p$ " ] ( * The Planck mass * )

```

Background Equation of Motion of Scalar Field

The zeroth order equation of motion for the scalar field is derived from the Lagrangian

$$L = \sqrt{-g} \left(-\frac{\nabla^\mu \phi \nabla_\mu \phi}{2} - V(\phi) - \rho_{\text{dm}} \right) \text{ by computing } 0 = \frac{\delta L}{\delta \phi(t, x, y, z)} - \partial_\mu \frac{\delta L}{\delta (\partial_\mu \phi(t, x, y, z))} \text{ in xPert.nb. This yields the modified Klein-Gordon equation } 0 = \nabla_\mu \nabla^\mu \phi - V' - \rho'_{\text{dm}}.$$

In[24]:=

```
eom0 = CD[α][CD[-α][sf[LI[0], LI[0]]] - D[V[sf[LI[0], LI[0]]], sf[LI[0], LI[0]]] -
nf[LI[0], LI[0]] × D[m[sf[LI[0], LI[0]]], sf[LI[0], LI[0]]]
```

Out[24]=

```
∇α∇αΦ - n m'[Φ] - V'[Φ]
```

Being the equation of motion, this expression vanishes. To simplify other equations, we create a rule from this.

In[25]:=

```
scfeom = MakeRule[{CD[β][CD[-β][sf[]]],
Derivative[1][m][sf[LI[0], LI[0]]] × nf[] + Derivative[1][V][sf[LI[0], LI[0]]]}
```

Out[25]=

```
{HoldPattern[∇β∇βΦ] :=> Module[{ }, n m'[Φ] + V'[Φ]], HoldPattern[∇β∇βΦ] :=> Module[{ }, n m'[Φ] + V'[Φ]]}
```

We can test the rule and see that the equation of motion vanishes:

In[26]:=

```
eom0 /. scfeom
```

Out[26]=

```
0
```

Metric-Dependent Klein-Gordon Equation

In our chosen flat background metric in conformal time the equation of motion of the scalar field is the Klein-Gordon equation

In[27]:=

```
xeom0 = MyxPand[eom0 ah[LI[0], LI[0]]^2, "AnyGauge", 0]
```

** Warning: The perturbed velocity, or the fields required to

parameterize the splitting of matter fields perturbations splitting were not previously defined **

** DefMatterFields is called to build the perturbation of the vector field and the projected fields needed for future splitting **

Out[27]=

```
-2 H φ̇ - φ̈ - a2 n m'[Φ] - a2 V'[Φ]
```

To implement this into CLASS, we must make three changes: First, express this 2nd order ODE into a system of two 1st order ODEs. Second, express $\mathcal{H} = H/a$, since H is in cosmic time in CLASS. Third, express the time derivatives as derivatives wrt the log of the scale factor a , i.e. $\frac{d}{dt} = \frac{H a}{d \log a} \frac{d}{d \log a}$. This yields:

$$\begin{aligned} \phi' &= \dot{\phi} / a H \\ H a \dot{\phi} &= -2 H a \dot{\phi} - a^2 (\rho' + V') \\ \Rightarrow \dot{\phi}' &= -2 \dot{\phi} - (a/H) (\rho' + V') \end{aligned}$$

The first-order perturbations also lead to equations of motion. We derive the abstract expression in xPert.nb and obtain

In[28]:=

```
eoms1 = (2 * CD[-α] [CD[α] [sf[LI[1], LI[0]]]] +
  CD[α] [sf[LI[0], LI[0]]] * (CD[-α] [dg[LI[1], β, -β]] - 2 * CD[-β] [dg[LI[1], -α, β]]) -
  2 * dg[LI[1], α, β] * CD[-β] [CD[-α] [sf[LI[0], LI[0]]]] +
  dg[LI[1], α, -α] * (CD[-β] [CD[β] [sf[LI[0], LI[0]]]] - nf[LI[0], LI[0]] *
    Derivative[1][m] [sf[LI[0], LI[0]]] - Derivative[1][V] [sf[LI[0], LI[0]]]) -
  2 * (nf[LI[1], LI[0]] * Derivative[1][m] [sf[LI[0], LI[0]]] +
    Scalar[sf[LI[1], LI[0]]] * (nf[LI[0], LI[0]] * Derivative[2][m] [sf[LI[0], LI[0]]] +
      Derivative[2][V] [sf[LI[0], LI[0]]]) ) / . scfeom
```

Out[28]=

$$2 \left(\bar{\nabla}_\alpha \bar{\nabla}^{\alpha(1)} \Phi \right) + (\bar{\nabla}^\alpha \Phi) \left(\bar{\nabla}_\alpha \delta g^{1\beta}_\beta - 2 \left(\bar{\nabla}_\beta \delta g^{1\beta}_\alpha \right) \right) - 2 \delta g^{1\alpha\beta} (\bar{\nabla}_\beta \bar{\nabla}_\alpha \Phi) - 2 \left(\binom{(1)}{n} m'[\Phi] + \binom{(1)}{\Phi} (n m''[\Phi] + V''[\Phi]) \right)$$

where we used the zeroth order equation of motion to simplify the expression. This is the perturbed Klein-Gordon equation $0 = \left(-\rho''_{\text{DM}} - V'' + \square \right) \delta\phi - m' \cdot \delta n + \phi^{i\mu} \cdot \left(\frac{1}{2} h^\nu_{\nu;\mu} - h^\nu_{\mu;\nu} \right) - h^{\mu\nu} \phi_{;\nu\mu}$, with δn the dark matter number density perturbation, $\phi^{i\mu} = \nabla^\mu \phi$, $\delta \phi_{;\mu}^\mu = \square \delta\phi$, and $h_{\mu\nu}$ the metric perturbation. As a next step, we express this for the Newtonian Gauge as

In[29]:=

```
xeoms1 = FullSimplify[
  MyxPand[eoms1, "NewtonGauge", 0] /. {sf[LI[0], LI[2]] -> -2 Hh[LI[0], LI[0]] * sf[LI[0], LI[1]] -
    ah[LI[0], LI[0]]^2 * nf[LI[0], LI[0]] * Derivative[1][m] [sf[LI[0], LI[0]]] -
    ah[LI[0], LI[0]]^2 * Derivative[1][V] [sf[LI[0], LI[0]]]} / /
  Simplification] * Scalar[ah[LI[0], LI[0]]]^2 / (-2)
```

Out[29]=

$$2 \mathcal{H} \left(\binom{(1)}{\Phi} \right) + \binom{(1)}{\Phi}'' - \dot{\Phi} \left(\binom{(1)}{\Phi}' + 3 \binom{(1)}{\psi}' \right) - \bar{D}_\alpha \bar{D}^\alpha \binom{(1)}{\Phi} +$$

$$\binom{(1)}{a}^2 \left(\binom{(1)}{n} m'[\Phi] + 2 \binom{(1)}{\Phi} V'[\Phi] + n \left(2 \binom{(1)}{\Phi} m'[\Phi] + \binom{(1)}{\Phi} m''[\Phi] \right) + \binom{(1)}{\Phi} V''[\Phi] \right)$$

In[30]:=

```
(* /. {sf[LI[0], LI[2]] -> -2 Hh[LI[0], LI[0]] sf[LI[0], LI[1]] - ah[LI[0], LI[0]]^2 nf[LI[0], LI[0]]
  Derivative[1][m] [sf[LI[0], LI[0]]] - ah[LI[0], LI[0]]^2 Derivative[1][V] [sf[LI[0], LI[0]]] }
```

enforces the expansion of ϕ by using the background EoM. This is needed to implement the expression into CLASS and using the quantities V' and ρ' instead of ϕ'' .*)

In order to implement this into CLASS, we have to express this second-order ODE as a system of two first-order ODEs. Furthermore, the quantity $\dot{\phi}$ is not present in CLASS. This metric perturbation is related to the shear. DE-DM interactions could however induce anisotropic stress, since the equation of state will not be $w = -1$ and we have interactions and a coupling. To take this into account, we introduce the scalar field momentum $p = \delta\phi - \phi \dot{\phi}$. The change in momentum wrt time is $\dot{p} = \delta\dot{\phi} - \dot{\phi} \dot{\phi} - \phi \ddot{\phi}$. In this, we can insert our first-order equation of motion that gives us $\delta\ddot{\phi}$. We then implement the following system of equations in CLASS:

$$\dot{\delta\phi} = p + \phi \dot{\phi} \dots$$

$$\dot{p} = \delta\phi - \dot{\phi} \dot{\phi} - \phi \dot{\phi}$$

$$= -2 H p - (a^2 V'' + a^2 m'' n - \partial_i \partial^i) \delta\phi - a^2 m' \delta n + 3 \dot{\Psi} \dot{\phi} - a^2 \phi (V' + m' n)$$

We explain some technical details about the implementations on the GitHub Wiki.

Particle Number Conservation

The total number of particles is assumed to be conserved. This is also the case in CLASS. Therefore, no change to the CLASS source code is necessary. The particle number conservation $\nabla_\mu (n u^\mu) = 0$ is a useful quantity to simplify equations. Therefore, we create a corresponding rule

In[31]:=

```
numcons = MakeRule [ { CD [ α ] [ nf [ ] ] × u [ - α ] , - nf [ ] × CD [ α ] [ u [ - α ] ] } ]
```

Out[31]=

```
{ HoldPattern [ ( ∇̄^α n ) u_α ] → Module [ { β } , - n ( ∇̄^β u_β ) ] , HoldPattern [ ( ∇̄_α n ) u^α ] → Module [ { β } , - n ( ∇̄^β u_β ) ] }
```

Expressed in our metric, the change in particle number is given by

In[32]:=

```
partcons = MyxPand [ CD [ β ] [ nf [ ] × u [ - β ] ] , "AnyGauge" , 0 ]
```

Out[32]=

$$\frac{3 \mathcal{H} n}{a} + \frac{\dot{n}}{a}$$

Particle Number Perturbation

While it is true that the particle number is conserved at the background level, there is still a perturbation to the particle number, which leads to an equation of motion at the perturbation level:

In[33]:=

```
partconspt = Simplify [ MyxPand [ CD [ β ] [ nf [ ] × u [ - β ] ] , "NewtonGauge" , 1 ] , partcons == 0 ]
```

Out[33]=

$$\frac{\epsilon \left(3 \mathcal{H} \left(\overset{(1)}{n} \right) + \overset{(1)}{n}' + n \left(-3 \left(\overset{(1)}{\psi} \right)' + \bar{\mathcal{D}}_\alpha \bar{\mathcal{D}}^\alpha \overset{(1)}{v}_u \right) \right)}{a}$$

Since there are no terms that depend on the scalar field, this corresponds to the CDM case and is already appropriately taken care of in CLASS.

Stress–Energy Tensor

The stress-energy tensor was derived from the action $(T^{\mu\nu} = \frac{2}{\sqrt{-g}} \frac{\delta(\sqrt{-g} L)}{\delta g_{\mu\nu}})$ using xPert.

In[34]:=

```
stressenergy = CD [ - β ] [ sf [ ] ] * CD [ - α ] [ sf [ ] ] -
```

```
1 / 2 g [ - α , - β ] ( CD [ - γ ] [ sf [ ] ] * CD [ γ ] [ sf [ ] ] + 2 V [ sf [ ] ] ) + m [ sf [ ] ] × nf [ ] × u [ - α ] × u [ - β ]
```

Out[34]=

$$m[\Phi] n u_\alpha u_\beta + (\bar{\nabla}_\alpha \Phi) (\bar{\nabla}_\beta \Phi) - \frac{1}{2} \bar{g}^{(4D)}_{\alpha\beta} (2V[\Phi] + (\bar{\nabla}_\gamma \Phi) (\bar{\nabla}^\gamma \Phi))$$

Einstein Source Equations

From the stress–energy tensor, we can obtain the Einstein source terms for the scalar field and the dark matter term. We have to split the tensor into the time–time, the space–time, and the spatial part, to read off ρ and p for the scalar field ϕ as well as for DM. However, CLASS uses the formalism by Ma & Bertschinger, i.e. we have to obtain the Einstein source terms from the traceless stress–energy tensor $\Sigma_\mu^\nu = T_\mu^\nu - \delta_\mu^\nu T_\rho^\rho / 3$.

First, we compute the trace of the stress–energy tensor:

```
In[35]:= stresstrace = g [ α , β ] stressenergy
```

```
Out[35]= 
$$\bar{g}^{(4\text{D})\alpha\beta} \left( m[\Phi] n_{\alpha} u_{\beta} + (\bar{\nabla}_{\alpha}\Phi) (\bar{\nabla}_{\beta}\Phi) - \frac{1}{2} \bar{g}^{(4\text{D})}_{\alpha\beta} \left( 2V[\Phi] + (\bar{\nabla}_{\gamma}\Phi) (\bar{\nabla}^{\gamma}\Phi) \right) \right)$$

```

Then, we compute the traceless stress–energy tensor:

```
In[36]:= stresstraceless = stressenergy - (1 / 3) g [ -α , -β ] stresstrace
```

```
Out[36]= 
$$m[\Phi] n_{\alpha} u_{\beta} + (\bar{\nabla}_{\alpha}\Phi) (\bar{\nabla}_{\beta}\Phi) - \frac{1}{2} \bar{g}^{(4\text{D})}_{\alpha\beta} \left( 2V[\Phi] + (\bar{\nabla}_{\gamma}\Phi) (\bar{\nabla}^{\gamma}\Phi) \right) - \frac{4}{3} \left( m[\Phi] n_{\alpha} u_{\beta} + (\bar{\nabla}_{\alpha}\Phi) (\bar{\nabla}_{\beta}\Phi) - \frac{1}{2} \bar{g}^{(4\text{D})}_{\alpha\beta} \left( 2V[\Phi] + (\bar{\nabla}_{\gamma}\Phi) (\bar{\nabla}^{\gamma}\Phi) \right) \right)$$

```

Next, we need to raise one of the indices. Finally, we can split the traceless stress–energy tensor into the time–time, the time–space, and the spatial components, which yields the pressure, density, and flux terms of the scalar field and dark matter:

```
In[37]:= VisualizeTensor[MyxPanda[g [ λ , β ] stresstraceless], "NewtonGauge", 1, h]
```

```
Out[37]=
```

	n	h
n	$\frac{1}{3} m[\Phi] n + \frac{\dot{\phi}^2}{6a^2} + \frac{V[\Phi]}{3} + \epsilon \left(\frac{1}{3} m[\Phi] \left({}^{(1)}n \right) + \frac{\dot{\phi} \left({}^{(1)}\dot{\phi} \right)}{3a^2} - \frac{\dot{\phi}^2 \left({}^{(1)}\phi \right)}{3a^2} + \frac{1}{3} n \left({}^{(1)}\Phi \right) m'[\Phi] + \frac{1}{3} \left({}^{(1)}\Phi \right) V'[\Phi] \right)$	$\epsilon \left(-\frac{\left({}^{(1)}B \right) \dot{\phi}^2}{3a^2} + \frac{1}{3} m[\Phi] n \left({}^{(1)}v_u^{\lambda} \right) - \frac{\dot{\phi} \left(\bar{D}^{\lambda} \left({}^{(1)}\phi \right) \right)}{3a^2} + \frac{1}{3} m[\Phi] n \left(\bar{D}^{\lambda} \left({}^{(1)}v_u \right) \right) \right)$
h	$\epsilon \left(-\frac{1}{3} \left({}^{(1)}B_{\alpha} \right) m[\Phi] n - \frac{1}{3} m[\Phi] n \left({}^{(1)}v_{u_{\alpha}} \right) + \frac{\dot{\phi} \left(\bar{D}_{\alpha} \left({}^{(1)}\phi \right) \right)}{3a^2} - \frac{1}{3} m[\Phi] n \left(\bar{D}_{\alpha} \left({}^{(1)}v_u \right) \right) \right)$	$-\frac{\bar{h}_{\alpha}^{\lambda} \dot{\phi}^2}{6a^2} + \frac{1}{3} \bar{h}_{\alpha}^{\lambda} V[\Phi] + \epsilon \left(-\frac{\bar{h}_{\alpha}^{\lambda} \dot{\phi} \left({}^{(1)}\dot{\phi} \right)}{3a^2} + \frac{\bar{h}_{\alpha}^{\lambda} \dot{\phi}^2 \left({}^{(1)}\phi \right)}{3a^2} + \frac{1}{3} \bar{h}_{\alpha}^{\lambda} \left({}^{(1)}\Phi \right) V'[\Phi] \right)$

Density

In the time–time component T_0^0 we find the dark matter density $\rho_{DM} = m(\phi) n$ as well as the scalar field density $\rho_{\phi} = \frac{1}{3} \left(\frac{\dot{\phi}^2}{a^2} + V(\phi) \right)$. The dark matter density perturbation is given by $\delta\rho_{DM} = \frac{1}{3} \left(m \delta_n + n m' \delta\phi \right)$.

The scalar field density perturbation is $\delta\rho_\phi = \frac{1}{3} \left(\frac{\dot{\phi} \delta\dot{\phi}}{a^2} - \frac{\dot{\phi}^2 \dot{\phi}}{a^2} + \delta\phi V' \right)$, which can be expressed in terms of the earlier introduced scalar field momentum $p = \dot{\phi} - \phi \dot{\phi}$ as $\delta\rho_\phi = \frac{1}{3} \left(\frac{\dot{\phi} p}{a^2} + \delta\phi V' \right)$.

Pressure

From the spatial part T_i^j we obtain the scalar field pressure $P_\phi = \frac{1}{3} \left(\frac{\dot{\phi}^2}{2a^2} - V \right)$ and its perturbation

$$\delta P_\phi = \frac{1}{3} \left(\frac{\dot{\phi} p}{a^2} - \delta\phi V' \right).$$

Energy Flux

From the space-time part $T_i^0 = -T_0^i$ we obtain the energy flux. Since neither derivatives of $m(\phi)$ nor DE-DM coupling terms appear, this is already accounted for in CLASS.

Conservation Equation

The conservation equation of the stress-energy tensor $\nabla_\mu T^{\mu\nu} = 0 = \nabla_\mu \left(\frac{2}{\sqrt{-g}} \frac{\delta(\sqrt{-g} L)}{\delta g_{\mu\nu}} \right)$ yields

In[38]:=

```
teom = CD[α] [stressenergy] / m[sf[]] / nf[] /. numcons /. sfeom / /
```

Simplification /. numcons /. sfeom / / Simplification

Out[38]=

$$u^\alpha \left(\bar{\nabla}_\alpha u_\beta \right) + \frac{\left(u^\alpha u_\beta \left(\bar{\nabla}_\alpha \Phi \right) + \bar{\nabla}_\beta \Phi \right) m'[\Phi]}{m[\Phi]} + \frac{\left(-\bar{g}^{(4)D} \right)_{\beta\alpha} \left(\bar{\nabla}^\alpha \Phi \right) + \bar{\nabla}_\beta \Phi \right) V'[\Phi]}{m[\Phi] (n)}$$

Since no vector perturbation are implemented in CLASS, we can use this equation as a stepping stone for the velocity dispersion evolution.

Velocity Dispersion Evolution

Taking the derivative of this equation yields the velocity dispersion evolution

$$0 = \partial_\eta \theta + H \theta + \square \phi + \frac{m'}{m} \left(\theta \dot{\phi} + \square \delta\phi \right)$$

In[39]:=

```
MyxPand[CD[β] [teom] × ah[LI[0], LI[0]] ^2, "NewtonGauge", 1] /. numcons /. sfeom / / Simplification
```

Out[39]=

$$\epsilon \left(\mathcal{H} \left(\bar{\mathcal{D}}_\alpha \bar{\mathcal{D}}^{\alpha(1)} v_u \right) + \bar{\mathcal{D}}_\alpha \bar{\mathcal{D}}^{\alpha(1)} v_u + \bar{\mathcal{D}}_\alpha \bar{\mathcal{D}}^{\alpha(1)} \phi + \frac{\left(\bar{\mathcal{D}}_\alpha \bar{\mathcal{D}}^{\alpha(1)} \Phi + \dot{\phi} \left(\bar{\mathcal{D}}_\alpha \bar{\mathcal{D}}^{\alpha(1)} v_u \right) \right) m'[\Phi]}{m[\Phi]} \right)$$

Since this is already a first-order perturbation equation, we have everything we need at this point to implement it into CLASS. We refer to the GitHub Wiki and the CLASS source code for the details about the technical implementation. We note that $\bar{\mathcal{D}}_\alpha \bar{\mathcal{D}}^{\alpha(1)} v_u$ is called θ in CLASS, and $\bar{\mathcal{D}}^{\alpha(1)} \phi$ is Ψ in CLASS.