Abstract Governing Equations for Dark Energy–Dark Matter Interactions

Using xPert

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, where the dark matter mass $m = m(\phi)$.
See https://github.com/kabeleh/iDM/ for the implementation into the
Cosmic Linear Anisotropy Solving System (CLASS) source code. In particular, the comprehensive GitHub Wiki explains technical details.

Load xPert package

Mathematica 14.3 introduces 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`xPert`
```

Package xAct`xPerm` version 1.2.3, {2015, 8, 23}

Connecting to external linux executable...

Connection established.

Package xAct`xTensor` version 1.2.0, {2021, 10, 17}

Package xAct`xPert` version 1.0.6, {2018, 2, 28}

** 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

In[5]:= **$DefInfoQ = False;**

In[6]:= **$PrePrint = ScreenDollarIndices;**

Derive abstract equations

We define a 4D manifold *M*

In[7]:= **DefManifold[M, 4, IndexRange[a, n]]**

and an abstract metric *g*

In[8]:= **DefMetric[-1, background[-a, -b], cd, PrintAs → "g"]**

with the perturbations of the metric stored as *h*

In[9]:= **DefMetricPerturbation[background, pert, ϵ, PrintAs → "h"]**

Lagrangian

In our model of DM–DE interactions, the DM mass m depends on the scalar field ϕ. The Lagrangian of our model that otherwise corresponds to ΛCDM contains the terms $L \supset -\frac{\nabla^{\mu}\phi\nabla_{\mu}\phi}{2} - V(\phi) - \rho_{\mathrm{dm}}$. In the following, we set up our system to make computations using this Lagrangian, including perturbations.

In[10]:=

```
DefTensor[sf[], M, PrintAs → "ϕ"] (* our scalar field *)
DefTensorPerturbation[pertsf[LI[order]], sf[], M, PrintAs → "δϕ"] (*The scalar field perturbations*)
```

In[12]:=

```
DefConstantSymbol[massP, PrintAs → "M_p"] (*The Planck mass*)
DefScalarFunction[mf, PrintAs → "m"] (*The DM mass,
defined as a scalar function to realise the ϕ dependence*)
DefScalarFunction[V] (*The scalar field potential*)
```

In[15]:=

```
DefTensor[nf[], M, PrintAs → "n"] (*The DM number density*)
DefTensorPerturbation[pertnf[LI[order]], nf[], M, PrintAs → "δn"]
 (*The DM number density perturbation*)
```

Now, we have all ingredients to define our Lagrangian as $L = \sqrt{-g}\left(-\frac{\nabla^{\mu}\phi\nabla_{\mu}\phi}{2} - V(\phi) - \rho_{\rm dm}\right)$:

In[17]:=

```
L = Sqrt[-Detbackground[]] (-1/2 cd[-b][sf[]] × cd[b][sf[]] - V[sf[]] - nf[] × mf[sf[]])
```

Out[17]=

$$\sqrt{-\tilde{\tilde{g}}}\left(-m[\phi]\,n - V[\phi] - \frac{1}{2}\left(\nabla_b\phi\right)\left(\nabla^b\phi\right)\right)$$

The variation of the Lagrangian is given by:

In[18]:=

```
varL = L // Perturbation // ExpandPerturbation // ContractMetric // ToCanonical
```

Out[18]=

$$-\frac{1}{2}\sqrt{-\tilde{\tilde{g}}}\,m[\phi]\,n\,h^{1b}{}_{b} - \sqrt{-\tilde{\tilde{g}}}\,m[\phi]\,\delta n^{1} - \frac{1}{2}\sqrt{-\tilde{\tilde{g}}}\,h^{1b}{}_{b}\,V[\phi] - \sqrt{-\tilde{\tilde{g}}}\left(\nabla_b\phi\right)\delta\phi^{1;b} +$$
$$\frac{1}{2}\sqrt{-\tilde{\tilde{g}}}\,h^{1}{}_{ba}\left(\nabla^a\phi\right)\left(\nabla^b\phi\right) - \frac{1}{4}\sqrt{-\tilde{\tilde{g}}}\,h^{1a}{}_{a}\left(\nabla_b\phi\right)\left(\nabla^b\phi\right) - \sqrt{-\tilde{\tilde{g}}}\,n\,\delta\phi^{1}\,m'[\phi] - \sqrt{-\tilde{\tilde{g}}}\,\delta\phi^{1}\,V'[\phi]$$

Equation of Motion

The equations of motion are derived from the Lagrangian as $0 = \frac{\delta L}{\delta\phi(t,x,y,z)} - \partial_{\mu}\frac{\delta L}{\delta\left(\partial_{\mu}\phi(t,x,y,z)\right)}$.

Background

The background equation of motion is the modified Klein–Gordon equation $0 = \nabla_{\mu}\nabla^{\mu}\phi - V' - \rho'_{\rm dm}$, which can either be obtained by taking the derivatives with respect to the background, such that $0 = \frac{\delta L^{(0)}}{\delta\overline{\phi}(t)} - \partial_t\frac{\delta L^{(0)}}{\delta\left(\partial_t\overline{\phi}(t)\right)}$, which yields

In[19]:=

```
VarD[sf[], cd][L] / Sqrt[-Detbackground[]] // Simplify
```

Out[19]=

$$\nabla_a\nabla^a\phi - n\,m'[\phi] - V'[\phi]$$

or by taking the derivatives with respect to the first order perturbations, such that

$$0 = \frac{\delta L^{(1)}}{\delta(\delta\phi(t,x,y,z))} - \partial_\mu \frac{\delta L^{(1)}}{\delta\left(\partial_\mu \delta\phi(t,x,y,z)\right)}$$, which yields

In[20]:=

```
eom0 = VarD[pertsf[LI[1]], cd][varL] / Sqrt[-Detbackground[]] /.
    delta[-LI[1], LI[1]] → 1 // Simplification
```

Out[20]=

$$\nabla_a \nabla^a \phi - n\, m'[\phi] - V'[\phi]$$

First-Order Perturbations

The latter approach can be used for any higher order of perturbations, e.g. for the first order perturbations we use the second order Lagrangian

In[21]:=

```
varL2 = ToCanonical[ContractMetric[ExpandPerturbation[Perturbation[L, 2]]]] /
    Sqrt[-Detbackground[]] // Simplification
```

Out[21]=

$$\frac{1}{8}\Big(2\, m[\phi] \Big(2 n\, h^{1}{}_{ba}\, h^{1ba} - 2 n\, h^{2b}{}_{b} - 2 h^{1b}{}_{b}\left(n\, h^{1a}{}_{a} + 2\,\delta n^{1}\right) - 4\,\delta n^{2} + n\left(h^{1a}{}_{a}\right)^{2}\Big) - 4 h^{1a}{}_{a}\, h^{1b}{}_{b}\, V[\phi] - 4 h^{2b}{}_{b}\, V[\phi] +$$
$$2\left(h^{1a}{}_{a}\right)^{2} V[\phi] - 8\,\delta\phi^{1}{}_{;b}\,\delta\phi^{1;b} - 8 h^{1a}{}_{a}\left(\nabla_b \phi\right)\delta\phi^{1;b} - 8\left(\nabla_b\phi\right)\delta\phi^{2;b} - 8 h^{1}{}_{ac}\, h^{1}{}_{b}{}^{c}\left(\nabla^a\phi\right)\left(\nabla^b\phi\right) +$$
$$4 h^{2}{}_{ba}\left(\nabla^a\phi\right)\left(\nabla^b\phi\right) + 2 h^{1}{}_{ac}\, h^{1ac}\left(\nabla_b\phi\right)\left(\nabla^b\phi\right) - 2 h^{1a}{}_{a}\, h^{1c}{}_{c}\left(\nabla_b\phi\right)\left(\nabla^b\phi\right) - 2 h^{2a}{}_{a}\left(\nabla_b\phi\right)\left(\nabla^b\phi\right) +$$
$$\left(h^{1a}{}_{a}\right)^{2}\left(\nabla_b\phi\right)\left(\nabla^b\phi\right) + 4 h^{1}{}_{ba}\Big(h^{1ba}\, V[\phi] + \left(\nabla^a\phi\right)\left(4\,\delta\phi^{1;b} + h^{1c}{}_{c}\left(\nabla^b\phi\right)\right)\Big) - 8 n\, h^{1b}{}_{b}\,\delta\phi^{1}\, m'[\phi] -$$
$$16\,\delta n^{1}\,\delta\phi^{1}\, m'[\phi] - 8 n\,\delta\phi^{2}\, m'[\phi] - 8 h^{1b}{}_{b}\,\delta\phi^{1}\, V'[\phi] - 8\,\delta\phi^{2}\, V'[\phi] - 8 n\left(\delta\phi^{1}\right)^{2} m''[\phi] - 8\left(\delta\phi^{1}\right)^{2} V''[\phi]\Big)$$

and take the derivative with respect to the scalar field perturbation to get the first order equation of motion

In[22]:=

```
eoms1 =
  Simplify[VarD[pertsf[LI[1]], cd][varL2] /. {delta[-LI[1], LI[1]] → 1, delta[-LI[2], LI[1]] → 0,
        delta[-LI[1], LI[2]] → 0}, eom0 == 0] // ToCanonical // Simplification // FullSimplify
```

Out[22]=

$$2\,\delta\phi^{1;a}{}_{;a} + h^{1b}{}_{b;a}\left(\nabla^a\phi\right) - 2\Big(\left(\nabla^a\phi\right) h^{1}{}_{a}{}^{b}{}_{;b} + h^{1ab}\left(\nabla_b\nabla_a\phi\right) + \delta n^{1}\, m'[\phi] + \left(\delta\phi^{1}\right)\left(n\, m''[\phi] + V''[\phi]\right)\Big)$$

We used the zeroth-order equation of motion (eom0==0) to simplify the expression. The resulting first-order equation of motion is $\mathbf{0} = \left(-\rho''_{\mathrm{DM}} - V'' + \Box\right)\delta\phi - m' \cdot \delta n + \phi^{;\mu} \cdot \left(\frac{1}{2} h^{\nu}_{\ \nu;\mu} - h^{\nu}_{\ \mu;\nu}\right) - h^{\mu\nu}\,\phi_{;\nu\mu}$, with δn the dark matter number density perturbation, $\phi^{;\mu} = \nabla^\mu \phi$, $\delta\phi^{;\mu}_{;\mu} = \Box\,\delta\phi$, and $h_{\mu\nu}$ the metric perturbation.

Particle Number Conservation

The Cosmic Linear Anisotropy Solving System (CLASS) uses the perfect fluid approximation and the formalism by Ma & Bertschinger (arXiv:astro-ph/9506072), which includes the four-velocity u_μ, and assumes that the number of dark matter particles is conserved, i.e. $\nabla_\mu (n\, u^\mu) = 0$. We turn this into a rule that can later be applied to simplify equations.

In[23]:=

```
DefTensor[myu[-a], M, PrintAs → "u"] (* The four-velocity u_μ *)
DefTensorPerturbation[pertu[LI[order], -a], myu[-a], M, PrintAs → "δu"]
(* The perturbations of the four-velocity *)
```

In[25]:=

```
numcons = MakeRule[{cd[a][nf[]] × myu[-a], -nf[] × cd[a][myu[-a]]}]
```

Out[25]=

$\{\text{HoldPattern}[(\nabla^{\underline{a}} n)\, u_{\underline{a}}] \mapsto \text{Module}[\{b\}, -n\, u_b{}^{;b}],\ \text{HoldPattern}[(\nabla_{\underline{a}} n)\, u^{\underline{a}}] \mapsto \text{Module}[\{b\}, -n\, u_b{}^{;b}]\}$

Stress–Energy Tensor

General Setup

In[26]:=

```
DefTensor[myt[-a, -b], M, PrintAs → "T"] (* We define the stress-energy tensor T *)
DefTensorPerturbation[pertT[LI[order], -a, -b], myt[-a, -b], M, PrintAs → "δT"]
(* We define the stress-energy perturbations δT *)
```

Obtaining the Stress–Energy Tensor

The stress–energy tensor is obtained from the Lagrangian: $T^{\mu\nu} = \dfrac{2}{\sqrt{-g}} \dfrac{\delta\left(\sqrt{-g}\ L\right)}{\delta g_{\mu\nu}}$

In[28]:=

```
set[-a, -b] =
  2 (-VarD[pert[LI[1], a, b], cd][varL] / Sqrt[-Detbackground[]] /. delta[-LI[1], LI[1]] → 1 //
        SeparateMetric[background] // RicciToEinstein) //
     Expand // ContractMetric // ToCanonical
```

Out[28]=

$g_{ab}\, m[\phi]\, n + g_{ab}\, V[\phi] - (\nabla_a \phi)(\nabla_b \phi) + \frac{1}{2} g_{ab} (\nabla_c \phi)(\nabla^c \phi)$

which has the familiar form $T_{\mu u} = \nabla_\mu \phi \nabla_\nu \phi - \frac{1}{2} g_{\mu\nu} \left(\nabla^\gamma \phi \nabla_\gamma \phi + 2\, V\,(\phi)\right) - g_{\mu\nu}\, \rho.$

To be aligned with the choice of CLASS to use the formalism by Ma & Bertschinger and to make use of the perfect fluid approximation, we replace $-g_{\mu\nu}$ with $u_\mu\, u_\nu$ for the DM term and find $T_{\mu\nu} = \nabla_\mu \phi \nabla_\nu \phi - \frac{1}{2} g_{\mu\nu} \left(\nabla^\gamma \phi \nabla_\gamma \phi + 2\, V\,(\phi)\right) + \rho\, u_\mu\, u_\nu$

In[29]:=

```
myt[-a, -b] = cd[-a][sf[]] × cd[-b][sf[]] -
   background[-a, -b] / 2 (cd[c][sf[]] × cd[-c][sf[]] + 2 V[sf[]]) +
   myu[-a] × myu[-b] × mf[sf[]] × nf[]
```

Out[29]=

$m[\phi]\, u_a\, u_b\, n + (\nabla_a \phi)(\nabla_b \phi) - \frac{1}{2} g_{ab} \left(2 V[\phi] + (\nabla_c \phi)(\nabla^c \phi)\right)$

Velocity Dispersion Evolution

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

In[30]:=

```
DefTensor[eomt0[-b], M]
eomt0[-b] =
  FullSimplify[cd[a][myt[-a, -b]], {eom0 == 0, eoms1 == 0}] /. numcons // Expand //
     ContractMetric // ToCanonical // Simplification
```

Out[31]=

$$m[\phi]\, u^a\, n\, u_{b;a} + \left(\nabla_a \nabla^a \phi\right)\left(\nabla_b \phi\right) + u^a\, u_b\, n\left(\nabla_a \phi\right) m'[\phi] - \left(\nabla_b \phi\right) V'[\phi]$$

respectively $0 = u^\mu \nabla_\mu u_\nu + \frac{m'}{m}\left(\nabla_\nu \phi + u_\nu u_\mu \nabla^\mu \phi\right)$ by dividing by $m\, n$ and using the background EoM $\ddot{\phi} - V' = n\, m'$.

In CLASS, such vector equations are not implemented. We can obtain a scalar equation by taking the derivative of this equation, which yields the velocity dispersion evolution equation

In[32]:=

```
Simplify[cd[b][eomt0[-b]] /. numcons // Expand // ContractMetric // ToCanonical //
    Simplification, {eom0 == 0, eoms1 == 0}]
```

Out[32]=

$$\left(\nabla_a \nabla_b \nabla^b \phi\right)\left(\nabla^a \phi\right) + \left(\nabla_a \nabla^a \phi\right)\left(\nabla_b \nabla^b \phi\right) + m[\phi]\left(u^a\, u^b{}_{;a}\left(\nabla_b n\right) + u^a\, n\, u^b{}_{;a;b} + n\, u_{b;a}\, u^{a;b}\right) +$$
$$u^a\, n\left(\nabla_a \phi\right) u^b{}_{;b}\, m'[\phi] + 2\, u^a\, n\, u^b{}_{;a}\left(\nabla_b \phi\right) m'[\phi] + u^a\, u^b\left(\nabla_a n\right)\left(\nabla_b \phi\right) m'[\phi] +$$
$$u^a\, u^b\, n\left(\nabla_b \nabla_a \phi\right) m'[\phi] - \left(\nabla_a \nabla^a \phi\right) V'[\phi] + u^a\, u^b\, n\left(\nabla_a \phi\right)\left(\nabla_b \phi\right) m''[\phi] - \left(\nabla_a \phi\right)\left(\nabla^a \phi\right) V''[\phi]$$

As we will see in xPand.nb, this equation is already a first-order perturbation equation. Therefore, we do not need higher-order terms from the stress–energy tensor.

Einstein Source Equation

The time–time, time–space, and spatial parts of the stress–energy tensor yield the Einstein source equations. This requires us to choose a metric, which we will do in the next steps in the xPand.nb.