

xPand: an algorithm for perturbing homogeneous cosmologies

Cyril Pitrou^{1,2}, Xavier Roy³ and Obinna Umeh³

¹ Institut d'Astrophysique de Paris, Université Pierre & Marie Curie—Paris VI, CNRS-UMR 7095, 98 bis, Bd Arago, F-75014 Paris, France

² Institut Lagrange de Paris, Sorbonne Universités, 98 bis Bd Arago, F-75014 Paris, France

³ Astrophysics, Cosmology and Gravity Centre (ACGC), and Department of Mathematics and Applied Mathematics, University of Cape Town, Rondebosch 7701, South Africa

E-mail: pitrou@iap.fr

Received 1 March 2013, in final form 18 June 2013

Published 12 July 2013

Online at stacks.iop.org/CQG/30/165002

Abstract

In this paper, we develop in detail a fully geometrical method for deriving perturbation equations about a spatially homogeneous background. This method relies on the $3 + 1$ splitting of the background space–time and does not use any particular set of coordinates: it is implemented in terms of geometrical quantities only, using the tensor algebra package *xTensor* in the *xAct* distribution along with the extension for perturbations *xPert*. Our algorithm allows one to obtain the perturbation equations for all types of homogeneous cosmologies, up to any order and in all possible gauges. As applications, we recover the well-known perturbed Einstein equations for Friedmann–Lemaître–Robertson–Walker cosmologies up to second order and for Bianchi I cosmologies at first order. This work paves the way to the study of these models at higher order and to that of any other perturbed Bianchi cosmologies, by circumventing the usually too cumbersome derivation of the perturbed equations.

PACS numbers: 02.70.Wz, 98.80.Jk, 98.80.–k

(Some figures may appear in colour only in the online journal)

Introduction

Cosmological perturbation theory constitutes the cornerstone of our current understanding of the origin, evolution and formation of large-scale structures. The evolution history of perturbations is written about a fixed homogeneous and isotropic Friedmann–Lemaître–Robertson–Walker (FLRW) background space–time, and the interpretation of cosmological observations (such as WMAP, QUIET and ACT [1–3]), within this model, converge toward a coherent and unified picture of the Universe. This picture is likely to get even clearer when the next generation of large-array cosmological observations (such as EUCLID and SKA)

becomes operational. It is widely expected that these observations will generate large amount of data that will provide a per cent level accuracy for the cosmological parameters.

Cosmological perturbation equations are simple and straightforward to derive at linear order, but they are inadequate for understanding the late-time evolution of the Universe, precisely when the nonlinear gravitational effects, carrying the information of the physics of current interest, come into play. Going beyond first order is a difficult task, and in some cases it becomes extremely arduous to even perform a coordinate or gauge transformation at nonlinear order by hand. To the best of our knowledge, there is no available easy-to-use software designed for cosmology and capable of deriving all equations of motion for perturbed variables. The closely related available option is the GRTensor package [4], which runs on *Maple* or *Mathematica*. However, the outputs generated at linear order are already a bit complicated to understand, let alone its outputs at nonlinear orders, as it relies exclusively on a properly defined set of background coordinates each time it acts on a perturbed variable.

To fill up this gap, we have developed an algebra package for cosmological perturbation theory, called *xPand* [5], which uses the tools of the tensor algebra package *xTensor* and an extension for perturbation, *xPert* [6, 7]. The *xTensor* and *xPert* packages are part of the *xAct* distribution [6] that run on *Mathematica* and they are available under the General Public License. *xPert* is specifically designed to perform perturbations on arbitrary background space–times [8, 7], but it lacks the features for specializing to a specific background space–time and a specific form for the metric perturbations, as is needed in the case of cosmology. In [9, 10], these packages were used to study perturbations about a spherically symmetric space–time, more precisely around a Schwarzschild solution of the Einstein field equations. The *xPand* package now allows one to derive, in a simple and user-friendly manner, all the necessary equations for cosmological perturbation variables, around any homogeneous background space–times, at any order and in any gauge. Specifically, the available types of cosmologies cover the Minkowski, FLRW (flat and curved), and Bianchi space–times, and the available predefined gauge are: general gauge (no gauge choice), comoving gauge, flat gauge, isodensity gauge, Newtonian gauge and synchronous gauge.

The paper is organized as follows. In section 1, we provide a general overview of the mathematical framework on which *xPert* is built. In section 2, we detail the $3 + 1$ splitting of the background manifold into a family of homogeneous hypersurfaces orthogonal to a fundamental observer’s velocity. In section 3, we decompose the perturbed metric with respect to this foliation and define the scalar, vector and tensor perturbations. Each of these sections is supplemented by a presentation of the associated implementation in *xPand*, by means of several detailed examples. Finally, we summarize and discuss in section 4 the features and performances of the package.

1. Perturbations around a general space–time

In this section we briefly review the algorithm of *xPert* [7], which constitutes the basis of our method. For more details about perturbation theory in the context of cosmology, we refer the reader to, e.g., [14, 15].

1.1. General framework

Let us consider in what follows a background manifold $\overline{\mathcal{M}}$ along with its perturbed manifold \mathcal{M} . Both are related by means of a diffeomorphism $\phi : \overline{\mathcal{M}} \rightarrow \mathcal{M}$. Tensorial quantities are thus transported from one manifold to the other with the help of the pull-back ϕ^* , the push-forward

ϕ_* , and their respective inverses. The metric of the perturbed manifold relates to that of the background as

$$\phi^*(g) = \bar{g} + \Delta[\bar{g}] = \bar{g} + \sum_{n=1}^{\infty} \frac{\Delta^n[\bar{g}]}{n!}. \quad (1)$$

Here and in the sequel, we use boldface symbols for tensorial quantities, an over-bar for background quantities⁴, and the notation $\Delta[\bar{T}]$ (resp. $\Delta^n[\bar{T}]$) for the total (resp. n th order) perturbation of a tensor T . One may prefer to write Δ_ϕ instead of Δ , as the definition of the perturbations depends on the diffeomorphism ϕ , that is on the choice of the gauge. We however choose to omit this reference for the sake of clarity, and in order not to burden the notation unnecessarily, we shall moreover use the short-hand: $T = \phi^*(T)$, for any perturbed quantity.

Unless otherwise specified, when we write down the components of a tensor, these should be understood as expressed in a general basis⁵ (this holds equally for the background and perturbed tensors, and for the perturbations). Since all perturbation orders live on the background manifold, as they are the result of the pull-back of a tensorial quantity living on the perturbed manifold, we shall raise and lower indices using the background metric. We have for instance

$${}^{(n)}h^{\mu\nu} = \bar{g}^{\mu\rho} \bar{g}^{\nu\sigma} {}^{(n)}h_{\rho\sigma}, \quad (2)$$

for the n th order metric perturbations ${}^{(n)}h \equiv \Delta^n[\bar{g}]$.

1.2. Expansion of the curvature tensors

1.2.1. Mathematical framework. The inverse of the metric tensor is given by the relation

$$g^{-1} = (\bar{g} + \Delta[\bar{g}])^{-1}. \quad (3)$$

Expanding it into

$$g^{-1} = \bar{g}^{-1} \sum_{m=0}^{\infty} (-1)^m (\bar{g}^{-1} \Delta[\bar{g}])^m, \quad (4)$$

and making use of the definition ${}^{(n)}h = \Delta^n[\bar{g}]$, we obtain the n th order perturbation of g^{-1} :

$$\Delta^n[(\bar{g}^{-1})^{\mu\nu}] = \sum_{(k_i)} (-1)^m \frac{n!}{k_1! \dots k_m!} {}^{(k_m)}h^{\mu\zeta_m} {}^{(k_{m-1})}h_{\zeta_m}^{\zeta_{m-1}} \dots {}^{(k_2)}h_{\zeta_3}^{\zeta_2} {}^{(k_1)}h_{\zeta_2}^{\nu}, \quad (5)$$

where the sum $\sum_{(k_i)}$ runs over the 2^{n-1} sorted partitions of n for $m \leq n$ positive integers, such that $k_1 + \dots + k_m = n$. Note that $\Delta^n[(\bar{g}^{-1})^{\mu\nu}] \neq {}^{(n)}h^{\mu\nu} = \bar{g}^{\mu\rho} \bar{g}^{\nu\sigma} {}^{(n)}h_{\rho\sigma}$ (e.g., we have at first order: $\Delta^1[(\bar{g}^{-1})^{\mu\nu}] = -{}^{(1)}h^{\mu\nu}$).

By means of relation (5), we can express the perturbation of the connection coefficients as [7]

$$\Delta^n[\bar{\Gamma}^{\rho}_{\mu\nu}] = \sum_{(k_i)} (-1)^{m+1} \frac{n!}{k_1! \dots k_m!} {}^{(k_m)}h^{\rho\zeta_m} {}^{(k_{m-1})}h_{\zeta_m}^{\zeta_{m-1}} \dots {}^{(k_2)}h_{\zeta_3}^{\zeta_2} {}^{(k_1)}h_{\zeta_2\mu\nu}, \quad (6)$$

where the last term of the right-hand side is defined by

$${}^{(n)}h_{\rho\mu\nu} = \frac{1}{2} (\bar{\nabla}_\nu {}^{(n)}h_{\rho\mu} + \bar{\nabla}_\mu {}^{(n)}h_{\rho\nu} - \bar{\nabla}_\rho {}^{(n)}h_{\mu\nu}). \quad (7)$$

⁴ This convention differs from the one adopted in [7]. We however opt for this choice as it reflects the standard usage in cosmological perturbation theory.

⁵ We use Greek letters ($\alpha, \beta, \mu, \nu, \rho, \dots$) for space-time indices.

The perturbation of the Riemann tensor is given in all generality by

$$\Delta^n [\bar{R}_{\mu\nu\rho}{}^\sigma] = \bar{\nabla}_\nu (\Delta^n [\bar{\Gamma}^\sigma{}_{\mu\rho}]) - \sum_{k=1}^{n-1} \binom{n}{k} \Delta^k [\bar{\Gamma}^\zeta{}_{\nu\rho}] \Delta^{n-k} [\bar{\Gamma}^\sigma{}_{\zeta\mu}] - (\mu \leftrightarrow \nu), \quad (8)$$

and for a connection compatible with the metric, we have, from equation (6),

$$\begin{aligned} \Delta^n [\bar{R}_{\mu\nu\rho}{}^\sigma] = & \sum_{(k_i)} (-1)^m \frac{n!}{k_1! \dots k_m!} \left(\binom{k_m}{\zeta_m} h^{\sigma\zeta_m} \dots \binom{k_2}{\zeta_3} h_{\zeta_3}{}^{\zeta_2} \bar{\nabla}_\mu \binom{k_1}{\zeta_2} h_{\zeta_2\rho\nu} \right. \\ & + \sum_{s=2}^m \binom{k_m}{\zeta_m} h^{\sigma\zeta_m} \dots \binom{k_{s+1}}{\zeta_{s+2}} h_{\zeta_{s+2}}{}^{\zeta_{s+1}} \binom{k_s}{\zeta_s} h_{\zeta_s\zeta_{s+1}\mu} \binom{k_{s-1}}{\zeta_{s-1}} h^{\zeta_s\zeta_{s-1}} \dots \binom{k_2}{\zeta_3} h_{\zeta_3}{}^{\zeta_2} \binom{k_1}{\zeta_2} h_{\zeta_2\nu\rho} \left. \right) \\ & - (\mu \leftrightarrow \nu). \end{aligned} \quad (9)$$

The symbol $(\mu \leftrightarrow \nu)$ denotes the repetition of the preceding expression with indices μ and ν exchanged. The perturbation of the Ricci tensor is simply obtained by contracting the second and fourth indices of $\Delta^n [\bar{R}_{\mu\nu\rho}{}^\sigma]$ in the previous expression, and the perturbation of the Ricci scalar, $\bar{R} = \bar{g}^{\rho\sigma} \bar{R}_{\rho\sigma}$, is written:

$$\Delta^n [\bar{R}] = \sum_{k=0}^n \binom{n}{k} \Delta^k [\bar{g}^{\rho\sigma}] \Delta^{n-k} [\bar{R}_{\rho\sigma}]. \quad (10)$$

At last, the perturbation of the Einstein tensor is expressed according to

$$\Delta^n [\bar{G}_{\mu\nu}] = \Delta^n [\bar{R}_{\mu\nu}] - \frac{1}{2} \sum_{k=0}^n \sum_{j=0}^k \frac{n!}{k! j! (n-j-k)!} \binom{j}{\mu\nu} \Delta^k [\bar{g}^{\rho\sigma}] \Delta^{n-j-k} [\bar{R}_{\rho\sigma}]. \quad (11)$$

1.2.2. Implementation in xPert. All the perturbative expansions expounded above are implemented in the package *xPert* [7]. For completeness, we here briefly review its main commands. The package can be loaded by evaluating

```
In[1] := <<xAct'xPert'
        (Version and copyright messages)
```

We first define the four-dimensional manifold M with abstract indices $\{\alpha, \beta, \mu, \nu, \lambda, \sigma\}$:

```
In[2] := DefManifold[ M, 4, {\alpha, \beta, \mu, \nu, \lambda, \sigma} ];
```

and then we define the ambient metric g of negative signature, along with its associated covariant derivative CD :

```
In[3] := DefMetric[ -1, g[-\alpha, -\beta], CD, {";", "\bar{\nabla}"}, PrintAs->'g'];
```

where M and g respectively correspond to \bar{M} and \bar{g} . Several tensors related to this metric are automatically defined at the same time (e.g., all the curvature tensors). Note that in *xTensor*, the covariant indices of a tensor are represented with a minus sign ($g[-\alpha, -\beta]$ means $\bar{g}_{\alpha\beta}$), while the latter is omitted for contravariant indices ($g[\alpha, \beta]$ means $\bar{g}^{\alpha\beta}$).

Upon defining the perturbations dg of the metric g with the command:

```
In[4] := DefMetricPerturbation[ g, dg, \epsilon ];
```

where ϵ is the perturbative parameter to be used in the expansions, it becomes feasible to evaluate the perturbation of any tensor associated with the metric. For instance, the perturbation at first order of the Ricci scalar is simply obtained by evaluating

```

In[5] := ExpandPerturbation@Perturbed[ RicciScalarCD[], 1 ]//,
        ContractMetric//ToCanonical
Out[5] := R[\tilde{\nabla}] - \varepsilon dg^{1\alpha\beta} R[\tilde{\nabla}]_{\alpha\beta} + \varepsilon \tilde{\nabla}_\beta \tilde{\nabla}_\alpha dg^{1\alpha\beta} - \varepsilon \tilde{\nabla}_\beta \tilde{\nabla}^\beta dg^{1\alpha}{}_\alpha

```

The functions `ExpandPerturbation` and `Perturbed` are used to evaluate the perturbation of any expression up to any order (here at first order), the function `ContractMetric` removes the background metric tensor through contraction on dummy indices (i.e. repeated indices), and the function `ToCanonical` simplifies the result, gathering together the terms which are equal up to symmetries. Further details can be found in [7].

1.3. Conformal transformation

In *xTensor*, the *first* metric defined on the manifold is the one that is used to raise and lower any tensor indices. For our purpose, we have chosen it to be the *conformal* metric \tilde{g} (cf input `In[3]`), which is different from, but conformally related to, the background metric of the physical space–time. This choice ensures that the conventional way of moving the indices of perturbed fields (equation (2)) is well satisfied within our algorithm. We however need, now, to relate the tensorial quantities one considers in perturbation theory, namely those living on the background manifold of the physical space–time, to those we have defined or shall define on the conformal background manifold M .

We detail this important point in the rest of this section.

1.3.1. Mathematical framework. Let us denote by \tilde{g} the metric of the physical space–time, and by \bar{g} its background value. The metrics \tilde{g} and \bar{g} are related by the conformal transformation

$$\tilde{g}_{\mu\nu} = a^2 \bar{g}_{\mu\nu}, \quad (\tilde{g}^{-1})^{\mu\nu} = a^{-2} (\bar{g}^{-1})^{\mu\nu}, \quad (\tilde{g}^{-1})^{\mu\rho} \tilde{g}_{\rho\nu} = \delta^\mu{}_\nu, \quad (12)$$

with a being the scale factor of the background space–time⁶. Substituting into the first expression of (12) the perturbative expansion (1) for \bar{g} and its counterpart for \tilde{g} , we extend the conformal transformation to the background:

$$\tilde{\bar{g}}_{\mu\nu} = a^2 \bar{g}_{\mu\nu}, \quad (\tilde{\bar{g}}^{-1})^{\mu\nu} = a^{-2} \bar{g}^{\mu\nu}, \quad (13)$$

and to the perturbed level:

$${}^{(n)}\tilde{h}_{\mu\nu} = a^2 {}^{(n)}h_{\mu\nu}, \quad \text{with} \quad {}^{(n)}\tilde{h}^{\mu\nu} = \bar{g}^{\mu\rho} \bar{g}^{\nu\sigma} {}^{(n)}h_{\rho\sigma}. \quad (14)$$

The associated Levi-Civita connections $\tilde{\nabla}$ and ∇ , on the one hand, and $\tilde{\bar{\nabla}}$ and $\bar{\nabla}$, on the other, are related by⁷

$$\tilde{\nabla}_\mu \omega_\nu = \nabla_\mu \omega_\nu - C^\rho{}_{\mu\nu} \omega_\rho, \quad \tilde{\bar{\nabla}}_\mu \omega_\nu = \bar{\nabla}_\mu \omega_\nu - \bar{C}^\rho{}_{\mu\nu} \omega_\rho, \quad (15)$$

for any 1-form ω . Using equations (12) and (13), we can write the quantities $C^\rho{}_{\mu\nu}$ and $\bar{C}^\rho{}_{\mu\nu}$ as

$$C^\rho{}_{\mu\nu} = 2\delta^\rho{}_{(\mu} \nabla_{\nu)} \ln a - g_{\mu\nu} \nabla^\rho \ln a, \quad \bar{C}^\rho{}_{\mu\nu} = 2\delta^\rho{}_{(\mu} \bar{\nabla}_{\nu)} \ln a - \bar{g}_{\mu\nu} \bar{\nabla}^\rho \ln a, \quad (16)$$

⁶ We stress again that the first metric defined on M is the one that is used to raise and lower any tensor indices. Defining another metric, say f , on the same manifold from the function `DefMetric` actually creates two objects: (i) the tensor f , with internal notation \mathfrak{f} , and (ii) the tensor f^{-1} , with internal notation `Inv[\mathfrak{f}]`. These have the following properties:

$$f^{\mu\nu} = \bar{g}^{\mu\rho} \bar{g}^{\nu\sigma} f_{\rho\sigma}, \quad (f^{-1})^{\mu\nu} = \bar{g}^{\mu\rho} \bar{g}^{\nu\sigma} (f^{-1})_{\rho\sigma}, \quad (f^{-1})^{\mu\rho} f_{\rho\nu} = \delta^\mu{}_\nu,$$

which explains the notation we use in equation (12).

⁷ By definition the scale factor used in the transformation (12) is not to be perturbed. Hence we have: $\tilde{\bar{g}} = \tilde{\bar{g}}$, and thus: $\tilde{\bar{\nabla}} = \tilde{\bar{\nabla}}$.

with $\nabla^\rho = g^{\rho\sigma} \nabla_\sigma$ and $\bar{\nabla}^\rho = \bar{g}^{\rho\sigma} \bar{\nabla}_\sigma$, and where the parentheses indicate symmetrization over the indices enclosed. We can then formulate the n th order perturbation of $C^\rho{}_{\mu\nu}$ as⁸

$$\Delta^n [\bar{C}^\rho{}_{\mu\nu}] = \sum_{k=0}^n \frac{n!}{k!(n-k)!} \{k\} h_{\mu\nu} \{n-k\} h^{\rho\sigma} \bar{\nabla}_\sigma \ln a. \quad (17)$$

With the help of the two previous relations, we are now able to provide the correspondence we seek.

For instance, the Riemann tensor associated with the metric \tilde{g} is given by (see appendix D of [16])

$$\begin{aligned} \tilde{R}_{\mu\nu\rho}{}^\sigma &= \bar{R}_{\mu\nu\rho}{}^\sigma - 2\bar{\nabla}_{[\mu} \bar{C}^\sigma{}_{\nu]\rho} + 2\bar{C}^\zeta{}_{\rho[\mu} \bar{C}^\sigma{}_{\nu]\zeta} \\ &= \bar{R}_{\mu\nu\rho}{}^\sigma + 2\delta^\sigma{}_{[\mu} \bar{\nabla}_{\nu]} \bar{\nabla}_\rho \ln a - 2\bar{g}_{\rho[\mu} \bar{\nabla}_{\nu]} \bar{\nabla}^\sigma \ln a \\ &\quad - 2\delta^\sigma{}_{[\mu} \bar{\nabla}_{\nu]} \ln a \bar{\nabla}_\rho \ln a + 2\bar{g}_{\rho[\mu} \bar{\nabla}_{\nu]} \ln a \bar{\nabla}^\sigma \ln a - 2\bar{g}_{\rho[\mu} \delta^\sigma{}_{\nu]} \bar{\nabla}^\zeta \ln a \bar{\nabla}_\zeta \ln a, \end{aligned} \quad (18)$$

where the brackets indicate anti-symmetrization over the indices enclosed. Perturbing this expression and using equation (17), we can finally relate $\Delta^n [\tilde{R}_{\mu\nu\rho}{}^\sigma]$ to $\Delta^n [\bar{R}_{\mu\nu\rho}{}^\sigma]$ and recover the usual quantities studied in perturbation theory.

1.3.2. Implementation in xPand. The *xTensor* package provides the tools to define a metric conformally related to another, thanks to the option `ConformalTo` of the function `DefMetric`. We have encapsulated this in *xPand* within the function `DefConformalMetric`, which furthermore ensures the transitivity of several conformal transformations.

Let us load the package *xPand*:

```
In[6] := <<xAct 'xPand'
-----
Package xAct'xPand' version 0.4.0, {2013,02,08}
(CopyRight (C) 2012-2013, Cyril Pitrou, Xavier Roy and Obinna Umeh under
the GPL.)
```

By evaluating the command

```
In[7] := DefConformalMetric[ g, a ];
```

we define the scalar factor `a[]` and the metric `ga2` conformally related to `g` through `a` (`ga2` thus corresponds to the background metric \tilde{g} of the physical space–time). To obtain the expression of any tensorial quantities living on the manifold described by `ga2` in terms of those defined on `M`, one then simply has to use the *xPand* function `Conformal`.

For instance, we obtain for the Riemann tensor associated with `ga2` the expression

⁸ The background and perturbed connections, on the one hand, and the background and conformal connections, on the other hand, are respectively related by

$$\nabla_\mu \omega_\nu = \bar{\nabla}_\mu \omega_\nu - \Delta[\bar{\Gamma}^\rho{}_{\mu\nu}] \omega_\rho, \quad \tilde{\nabla}_\mu \tilde{\omega}_\nu = \bar{\nabla}_\mu \tilde{\omega}_\nu - \bar{C}^\rho{}_{\mu\nu} \tilde{\omega}_\rho,$$

for any 1-form ω . Performing a conformal transformation on the former expression and perturbing the latter respectively yields

$$\begin{aligned} \tilde{\nabla}_\mu \tilde{\omega}_\nu &= \tilde{\bar{\nabla}}_\mu \tilde{\omega}_\nu - \Delta[\tilde{\bar{\Gamma}}^\rho{}_{\mu\nu}] \tilde{\omega}_\rho = \tilde{\bar{\nabla}}_\mu \tilde{\omega}_\nu - \bar{C}^\rho{}_{\mu\nu} \tilde{\omega}_\rho - \Delta[\tilde{\bar{\Gamma}}^\rho{}_{\mu\nu}] \tilde{\omega}_\rho, \\ \tilde{\nabla}_\mu \tilde{\omega}_\nu &= \nabla_\mu \tilde{\omega}_\nu - C^\rho{}_{\mu\nu} \tilde{\omega}_\rho = \bar{\nabla}_\mu \tilde{\omega}_\nu - \Delta[\bar{\Gamma}^\rho{}_{\mu\nu}] \tilde{\omega}_\rho - C^\rho{}_{\mu\nu} \tilde{\omega}_\rho. \end{aligned}$$

These relations can only be compatible if

$$\Delta[\tilde{\bar{\Gamma}}^\rho{}_{\mu\nu}] - \Delta[\bar{\Gamma}^\rho{}_{\mu\nu}] = \Delta[\bar{C}^\rho{}_{\mu\nu}].$$

It can be checked directly from equations (6), (14) and (17) that this is indeed the case. This shows the equivalence between the transformations $\bar{\nabla} \rightarrow \nabla \rightarrow \tilde{\bar{\nabla}}$ and $\bar{\nabla} \rightarrow \tilde{\bar{\nabla}} = \tilde{\bar{\nabla}} \rightarrow \tilde{\bar{\nabla}}$. The latter approach actually proves itself to be faster within *xPand*. It is therefore the one that we have coded in the function `ToxPand` (see section 4.2).

$$\begin{aligned}
In[8] &:= \text{Conformal}[g, ga2][\text{RiemannCD}[-\alpha, -\beta, -\mu, \nu]] \\
Out[8] &:= R[\bar{\nabla}]_{\alpha\beta\mu}{}^\nu - \delta_\beta{}^\nu \bar{g}_{\alpha\mu} \frac{\bar{\nabla}_\lambda a \bar{\nabla}^\lambda a}{a^2} + \delta_\alpha{}^\nu \bar{g}_{\beta\mu} \frac{\bar{\nabla}_\lambda a \bar{\nabla}^\lambda a}{a^2} + 2\delta_\beta{}^\nu \frac{\bar{\nabla}_\alpha a \bar{\nabla}_\mu a}{a^2} \\
&\quad - 2\delta_\alpha{}^\nu \frac{\bar{\nabla}_\beta a \bar{\nabla}_\mu a}{a^2} - \delta_\beta{}^\nu \frac{\bar{\nabla}_\mu \bar{\nabla}_\alpha a}{a} + \delta_\alpha{}^\nu \frac{\bar{\nabla}_\mu \bar{\nabla}_\beta a}{a} - 2\bar{g}_{\beta\mu} \frac{\bar{\nabla}_\alpha a \bar{\nabla}^\nu a}{a^2} \\
&\quad + 2\bar{g}_{\alpha\mu} \frac{\bar{\nabla}_\beta a \bar{\nabla}^\nu a}{a^2} + \bar{g}_{\beta\mu} \frac{\bar{\nabla}^\nu \bar{\nabla}_\alpha a}{a} - \bar{g}_{\alpha\mu} \frac{\bar{\nabla}^\nu \bar{\nabla}_\beta a}{a}
\end{aligned}$$

which coincides with equation (18).

The conformal transformation may as well be performed on quantities that are not related to a metric. For a general tensor, it is defined as

$$\tilde{T}^{\mu_1 \dots \mu_p}{}_{\nu_1 \dots \nu_q} = a^{q-p+W(T)} T^{\mu_1 \dots \mu_p}{}_{\nu_1 \dots \nu_q}, \quad (19)$$

where $W(T)$ is the *conformal weight* of the tensor T . The default value of W is chosen to be zero in order to leave the norm invariant under a conformal transformation:

$$\begin{aligned}
In[9] &:= \text{DefTensor}[W[-\alpha], M]; \\
In[10] &:= \text{Conformal}[g, ga2][W[-\alpha]] \\
Out[10] &:= a W_\alpha \\
In[11] &:= \text{Conformal}[g, ga2][W[\alpha]] \\
Out[11] &:= \frac{W^\alpha}{a}
\end{aligned}$$

The conformal weight can however be modified for each tensor with the *xPand* function `ConformalWeight`. This can be of use for instance to preserve the geodesic character of light-like vectors⁹:

$$\begin{aligned}
In[12] &:= \text{DefTensor}[k[-\alpha], M]; \\
In[13] &:= \text{ConformalWeight}[k] \hat{=} -1; \\
In[14] &:= \text{ConformalWeight}[k[-\alpha]] \\
Out[14] &:= 0 \\
In[15] &:= \text{ConformalWeight}[k[\alpha]] \\
Out[15] &:= -2 \\
In[16] &:= \text{Conformal}[g, ga2][k[\alpha] \text{CD}[-\alpha] @ k[-\beta]] \\
Out[16] &:= \frac{k^\alpha \bar{\nabla}_\alpha k_\beta}{a^2} - k_\alpha k^\alpha \frac{\bar{\nabla}_\beta a}{a^3}
\end{aligned}$$

So far, by applying `Conformal` then `ExpandPerturbation@Perturbed` on a given expression defined on M , one obtains the perturbation of its conformal transformation in terms of the tensors defined on M , the metric g , its perturbations dg , the connection $\bar{\nabla}$, and the scale factor a . To end up with the usual expressions of perturbation theory, one needs to perform a $3 + 1$ splitting of the background manifold (section 2), then decompose each perturbed fields into its spatial and temporal parts and finally parameterize the perturbations of the metric (section 3).

⁹ We will make use of this prescription in a future version of *xPand* to implement the derivation of the (perturbed) null geodesic equation.

2. 3 + 1 splitting of the background manifold

2.1. Induced metric

The assumption that the background space–time possesses a set of (three-dimensional) homogeneous surfaces provides a natural choice for the 3+1 slicing. We foliate the background manifold by means of this family, and we denote by $\bar{\mathbf{n}}$ the unit time-like vector ($\bar{n}^\mu \bar{n}_\mu = -1$) normal to it. The metric of $\bar{\mathcal{M}}$ is decomposed as

$$\bar{g}_{\mu\nu} = \bar{h}_{\mu\nu} - \bar{n}_\mu \bar{n}_\nu, \quad \text{with} \quad \bar{h}_{\mu\nu} \bar{n}^\mu = 0 \quad \text{and} \quad \bar{h}^\mu{}_\rho \bar{h}^\rho{}_\nu = \bar{h}^\mu{}_\nu, \quad (20)$$

where $\bar{\mathbf{h}}$ represents the induced metric of the spatial hypersurfaces¹⁰.

The acceleration of the so-called Eulerian observers satisfies in all generality [12]

$$\bar{a}_\mu = \bar{n}^\rho \bar{\nabla}_\rho \bar{n}_\mu = \frac{\bar{D}_\mu \bar{\alpha}}{\bar{\alpha}}, \quad (21)$$

with $\bar{\alpha}$ being the lapse function. \bar{D} stands for the connection of the three-surfaces associated with $\bar{\mathbf{h}}$ ($\bar{D}_\rho \bar{h}_{\mu\nu} = 0$), and it is related to the four-covariant derivative as

$$\bar{D}_\rho T_{\mu_1 \dots \mu_p} = \bar{h}^\sigma{}_\rho \bar{h}^{\nu_1}{}_{\mu_1} \dots \bar{h}^{\nu_p}{}_{\mu_p} \bar{\nabla}_\sigma T_{\nu_1 \dots \nu_p}, \quad (22)$$

for any spatial tensor field¹¹. Since the lapse is homogeneous in the configuration at stake, the acceleration vanishes and the observers are in geodesic motion. We can therefore label each hypersurface by their proper time η and write $\bar{n}_\mu = -\bar{\nabla}_\mu \eta$. In addition, $\bar{\mathbf{n}}$ being hypersurface-forming by construction, its vorticity vanishes; this property yields

$$\bar{\omega}_{\mu\nu} = \bar{h}^\rho{}_\mu \bar{h}^\sigma{}_\nu \bar{\nabla}_{[\rho} \bar{n}_{\sigma]} = 0 \quad \Leftrightarrow \quad \bar{\nabla}_{[\mu} \bar{n}_{\nu]} = 0, \quad (23)$$

where the equivalence stems from the null acceleration. For comprehensive reviews on the 3 + 1 formalism, we refer the reader to, e.g., [12, 13].

2.2. Extrinsic curvature

Another tensor we shall make use of is the symmetric extrinsic curvature tensor, which characterizes the way the three surfaces are embedded into the background manifold. It satisfies the relation

$$\bar{K}_{\mu\nu} = \bar{h}^\rho{}_\mu \bar{h}^\sigma{}_\nu \bar{\nabla}_\rho \bar{n}_\sigma, \quad (24)$$

where we have chosen a positive sign for the right-hand side¹². From the decomposition (20) along with the vanishing of the acceleration $\bar{\mathbf{a}}$ and the unitarity of $\bar{\mathbf{n}}$, we can reformulate expression (24) as

$$\bar{K}_{\mu\nu} = \bar{\nabla}_\mu \bar{n}_\nu. \quad (25)$$

Since the volume expansion of the background space–time is entirely contained in the scale factor a , and owing to the conformal transformation (13), the trace of the extrinsic curvature vanishes: $\bar{K}^\mu{}_\mu = 0$. As a result, we have for general Bianchi cosmologies: $\bar{K}_{\mu\nu} = \bar{\sigma}_{\mu\nu}$, with $\bar{\sigma}_{\mu\nu}$ being the shear of the Eulerian observers; and for FLRW cosmologies: $\bar{K}_{\mu\nu} = 0$.

¹⁰ For the sake of clarity, let us note that ${}^{(n)}\mathbf{h}$ are *not* the perturbations of $\bar{\mathbf{h}}$. From the definition of ${}^{(n)}\mathbf{h}$ together with relation (20), we can actually relate them as $\Delta^n[{}^{(n)}\mathbf{h}] = {}^{(n)}\mathbf{h} - \Delta^n[\bar{\mathbf{n}} \otimes \bar{\mathbf{n}}]$.

¹¹ We recall that the operator \bar{D} loses its character of derivative when it is applied to non-spatial tensors. More precisely, we are not allowed to use the Leibniz rule anymore, as one can realize upon writing for instance

$$\begin{aligned} \bar{D}_\rho(\psi \bar{T}_{\mu_1 \dots \mu_p}) &= \psi \bar{D}_\rho \bar{T}_{\mu_1 \dots \mu_p} + \bar{h}^{\nu_1}{}_{\mu_1} \dots \bar{h}^{\nu_p}{}_{\mu_p} \bar{T}_{\nu_1 \dots \nu_p} \bar{D}_\rho \psi \\ &\neq \psi \bar{D}_\rho \bar{T}_{\mu_1 \dots \mu_p} + \bar{T}_{\mu_1 \dots \mu_p} \bar{D}_\rho \psi, \end{aligned}$$

for any scalar field ψ .

¹² This convention does not affect the 3 + 1 Einstein equations as written in terms of the kinematical quantities of the Eulerian observers.

2.3. Curvature tensors

The splitting of the four-Riemann tensor can be constructed from its different projections onto the spatial slices and the congruence of the observers. It is written as

$$\begin{aligned} \bar{R}_{\mu\nu\rho\sigma} = & \bar{R}_{\mu\nu\rho\sigma} + 2\bar{K}_{\mu[\rho}\bar{K}_{\sigma]\nu} - 4(\bar{D}_{[\mu}\bar{K}_{\nu][\rho})\bar{n}_{\sigma]} - 4(\bar{D}_{[\rho}\bar{K}_{\sigma][\mu})\bar{n}_{\nu]} + 4\bar{n}_{[\mu}\bar{K}_{\nu]}^{\xi}\bar{K}_{\xi[\rho}\bar{n}_{\sigma]} \\ & + 4\bar{n}_{[\mu}\bar{K}_{\nu][\rho}\bar{n}_{\sigma]}, \end{aligned} \quad (26)$$

where $\bar{R}_{\mu\nu\rho\sigma}$ stands for the three-Riemann curvature of the hypersurfaces. The over-dot indicates the covariant derivative along the world-lines of the observers (for any tensor field T , we have $\dot{T}_{\mu_1\dots\mu_p} = \bar{n}^\rho\bar{\nabla}_\rho T_{\mu_1\dots\mu_p}$). The purely spatial projection of (26) only calls upon the first two terms, and it drives the Gauss relation

$$\bar{h}^\varphi_\mu\bar{h}^\nu_\nu\bar{h}^\xi_\rho\bar{h}^\zeta_\sigma\bar{R}_{\varphi\nu\xi\zeta} = \bar{R}_{\mu\nu\rho\sigma} + 2\bar{K}_{\mu[\rho}\bar{K}_{\sigma]\nu}. \quad (27)$$

The three-space and one-time projection gives, from the next two terms, the Codazzi relation

$$\bar{h}^\varphi_\mu\bar{h}^\nu_\nu\bar{h}^\xi_\rho\bar{n}^\zeta\bar{R}_{\varphi\nu\xi\zeta} = \bar{D}_\mu\bar{K}_{\nu\rho} - \bar{D}_\nu\bar{K}_{\mu\rho}, \quad (28)$$

and the last non-null projection (two-space and two-time) provides, from the last two terms, an evolution equation for the extrinsic curvature. These decompositions can be performed in *xTensor* with the function `GaussCodazzi`.

For FLRW space-times, the curvature tensors of the hypersurfaces cast the form

$$\bar{R}_{\mu\nu\rho\sigma} = 2k\bar{h}_{\rho[\mu}\bar{h}_{\nu]\sigma}, \quad \bar{R}_{\mu\nu} = 2k\bar{h}_{\mu\nu}, \quad \bar{R} = 6k, \quad (29)$$

with k being the curvature parameter (equal to zero for flat FLRW cosmologies). The corresponding expressions for general Bianchi space-times are more involved, as they require the introduction of the constants of structures. We detail their derivation in appendix B.

2.4. Derivatives

In order to achieve the 3 + 1 splitting, we are left with the decomposition of the covariant derivative $\bar{\nabla}$ in terms of the induced derivative \bar{D} . For general spatial tensors (namely, for spatial tensors defined within $\bar{\mathcal{M}}$ or defined within \mathcal{M} then mapped onto $\bar{\mathcal{M}}$), the relation between the two derivatives reads

$$\bar{\nabla}_\rho T_{\mu_1\dots\mu_p} = -\bar{n}_\rho\dot{T}_{\mu_1\dots\mu_p} + \bar{D}_\rho T_{\mu_1\dots\mu_p} + \sum_{i=1}^p \bar{n}_{\mu_i}\bar{K}^\sigma_{\rho}\bar{T}_{\mu_1\dots\mu_{i-1}\sigma\mu_{i+1}\dots\mu_p}. \quad (30)$$

Even though our formalism is purely geometrical, we aim at eventually providing, for the perturbations, partial differential equations with respect to the proper time η of the Eulerian observers. When considering the four-dimensional basis built to address the Bianchi classification (refer to appendix B), the Lie derivative along the direction of \bar{n} precisely comes down to ∂_η . It is accordingly more appropriate for our purpose to use the Lie derivative rather than the dot derivative.

The relation between $\mathcal{L}_{\bar{n}}$ and $\bar{n}^\rho\bar{\nabla}_\rho$ is written as¹³

$$\mathcal{L}_{\bar{n}}T_{\mu_1\dots\mu_p} = \dot{T}_{\mu_1\dots\mu_p} + \sum_{i=1}^p \bar{K}^\sigma_{\mu_i}T_{\mu_1\dots\mu_{i-1}\sigma\mu_{i+1}\dots\mu_p}, \quad (31)$$

and provides us with the following reformulation of (30):

$$\bar{\nabla}_\rho T_{\mu_1\dots\mu_p} = -\bar{n}_\rho\mathcal{L}_{\bar{n}}T_{\mu_1\dots\mu_p} + \bar{D}_\rho T_{\mu_1\dots\mu_p} + 2\sum_{i=1}^p \bar{n}_{(\mu_i}\bar{K}^\sigma_{\rho)}T_{\mu_1\dots\mu_{i-1}\sigma\mu_{i+1}\dots\mu_p}. \quad (32)$$

¹³ Note that for a spatial tensor T , the quantity $\mathcal{L}_{\bar{n}}T_{\mu_1\dots\mu_p}$ is also spatial.

Finally, the last expression we shall need is the commutation rule between the derivatives $\mathcal{L}_{\bar{n}}$ and \bar{D} . For general spatial tensors, it is given by

$$\mathcal{L}_{\bar{n}}(\bar{D}_\rho T_{\mu_1 \dots \mu_p}) = \bar{D}_\rho(\mathcal{L}_{\bar{n}} T_{\mu_1 \dots \mu_p}) + \sum_{i=1}^p (\bar{h}^{\sigma\zeta} \bar{D}_\zeta \bar{K}_{\rho\mu_i} - \bar{D}_\rho \bar{K}_{\mu_i}{}^\sigma - \bar{D}_{\mu_i} \bar{K}_\rho{}^\sigma) T_{\mu_1 \dots \mu_{i-1} \sigma \mu_{i+1} \dots \mu_p}, \quad (33)$$

where we have made use of relations (22) and (28) for its derivation.

2.5. Implementation in *xPand*

The 3 + 1 splitting of the background manifold is performed by the *xPand* function `SetSlicing`. It can be applied to the following spatially homogeneous cosmologies: ‘Minkowski’, ‘FLFlat’, ‘FLCurved’, ‘BianchiI’, ‘BianchiA’, ‘BianchiB’ and ‘Anisotropic’.

2.5.1. Construction of the spatial hypersurfaces. From the ambient metric g , `SetSlicing` first defines the unit normal vector n , the induced metric h of the hypersurfaces, the associated covariant derivative cd , the associated scale factor $a[h]$ and the conformal metric $gah2$. It then specifies the expressions of the intrinsic and extrinsic curvature tensors according to the type of cosmologies chosen by the user (hence, the geometry of the model is fully described).

Let us illustrate with the cosmologies ‘FLCurved’ and ‘BianchiA’ (for the latter most sophisticated case, refer to appendix B). With the evaluation

```
In[17] := SetSlicing[ g, n, h, cd, {‘|’, ‘D’}, ‘FLCurved’ ];
```

the extrinsic curvature $K[h][-\alpha, -\beta]$ is set to zero, and the expressions of the three-curvature tensors are implemented following relations (29). For instance, we have for the three-Ricci tensor

```
In[18] := Riccicd[-α, -β]
Out[18] := 2 h̄_αβ k
```

For ‘BianchiA’ cosmologies,

```
In[19] := SetSlicing[ g, nA, hA, cdA, {‘|’, ‘D’}, ‘BianchiA’ ];
```

`SetSlicing` defines the spatial constants of structure C^k_{ij} (see appendix B) and then constructs a function allowing one to express the intrinsic curvature tensors in terms of them. For the three-Ricci tensor, we then have

```
In[20] := RiccicdA[-α, -β] // ToConstantsOfStructure[hA]
Out[20] := 1/4 C_α^λμ C_βλμ - 1/2 C_λβμ C_α^λμ - 1/2 C_α^λμ C_μβλ + 1/2 C_αβ^λ C_λμ^μ + 1/2 C_βα^λ C_λμ^μ + 1/2 C_αβ^λ C_λμ^μ
```

which is equivalent to equation (B.21)¹⁴. The constants of structure can be further expanded, following the usual Schücking, Kundt and Behr (SKB) decomposition [17], by means of the *xPand* function `ToBianchiType[hA]`.

We summarize in table 1 the different evaluations performed by `SetSlicing` with respect to the type of cosmologies.

¹⁴ Equation (B.21) is recovered by making use of the Jacobi identity (B.5) on `Out[20]`. This latter relation is not implemented, as it implies a symmetry among several terms that *xTensor* does not yet handle. However, once the constants of structure are expanded according to the parameterization (B.8), the Jacobi identity reduces to the constraint (B.9), which is automatically applied in *xPand*.

Table 1. Evaluations performed by `SetSlicing` for the extrinsic curvature, the three-curvature tensors and the constants of structure, according to the type of homogeneous cosmologies. In terms of g (or, equivalently, \bar{g}), the dynamics of ‘Minkowski’ and ‘FLFlat’ space-times are identical. The difference lies in the conformal transformation (13): for the former models, the scale factor is set to 1. For ‘BianchiA’, ‘BianchiB’ and ‘Anisotropic’ space-times, the three-curvature tensors can be formulated in terms of the constants of structure by means of the function `ToConstantsOfStructure[]` and, for the two first models, can be further expanded with the help of `ToBianchiType[]`. The ‘Anisotropic’ space-time is used for hypersurfaces whose dimension differs from three (for these models the SKB decomposition does not apply).

| Space-time | Extrinsic curvature | Three-curvature tensors | Constants of structure |
|-------------|---------------------|-------------------------|------------------------------------------------------------|
| Minkowski | Null | Null | Null |
| FLFlat | Null | Null | Null |
| FLCurved | Null | Equations (29) | $C^k_{ij} = 2\sqrt{k} \bar{h}^{mk} \epsilon_{mij}$ |
| BianchiI | $\bar{K}_{\mu\nu}$ | Null | Null |
| BianchiA | $\bar{K}_{\mu\nu}$ | Equations (B.20)–(B.22) | $C^k_{ij} = \epsilon_{ijm} N^{mk}$ |
| BianchiB | $\bar{K}_{\mu\nu}$ | Equations (B.20)–(B.22) | $C^k_{ij} = \epsilon_{ijm} N^{mk} + 2A_{[i} \delta^k_{j]}$ |
| Anisotropic | $\bar{K}_{\mu\nu}$ | Equations (B.20)–(B.22) | C^k_{ij} |

2.5.2. *Derivatives.* For anisotropic cosmologies, `SetSlicing` creates a set of automatic rules to handle the action of the derivatives on the three-curvature tensors. The aim is to obtain a formulation in terms of the extrinsic curvature tensor and/or the constants of structure, which fully identify the geometry of the model. Note that such rules are not necessary for isotropic models: the expressions of the intrinsic curvature tensors are simple enough to let `xTensor` perform the evaluation.

The action of the derivative $\bar{\nabla}$ is implemented following equation (32), that of \bar{D} following equation (B.14) together with (B.16), and the application of $\mathcal{L}_{\bar{n}}$ on a three-curvature tensor automatically calls the function `ToConstantsOfStructure[]` so as to use the property (B.19). Regarding the three-covariant derivative, we have for instance

$$\begin{aligned}
In[21] &:= cdA[-\alpha] @ RiccidA[-\mu, -\nu] \\
Out[21] &:= \frac{1}{2} C_{\alpha\nu}^{\beta} R[\bar{D}]_{\mu\beta} - \frac{1}{2} C_{\alpha\nu}^{\beta} R[\bar{D}]_{\mu\beta} + \frac{1}{2} C_{\nu\alpha}^{\beta} R[\bar{D}]_{\mu\beta} + \frac{1}{2} C_{\alpha\mu}^{\beta} R[\bar{D}]_{\nu\beta} - \frac{1}{2} C_{\alpha\mu}^{\beta} R[\bar{D}]_{\nu\beta} \\
&\quad + \frac{1}{2} C_{\mu\alpha}^{\beta} R[\bar{D}]_{\nu\beta}
\end{aligned}$$

Lastly, `SetSlicing` constructs an automatic rule to perform the commutation (33) for any expression. This serves to make sure that the Lie derivative will first act on a given tensor, so as to recover the usual formulation of the perturbed equations¹⁵. For ‘FLCurved’ cosmologies, we have

$$\begin{aligned}
In[22] &:= DefTensor[V[-\alpha], M, OrthogonalTo->\{n[\alpha]\}, ProjectedWith->\{h[\alpha, -\beta]\}]; \\
In[23] &:= LieD[n[\mu]] @ cd[-\nu] @ V[-\alpha] \\
Out[23] &:= \bar{D}_{\nu} \mathcal{L}_{\bar{n}} V_{\alpha}
\end{aligned}$$

and for ‘BianchiA’,

$$In[24] := DefTensor[TA[-\alpha], M, OrthogonalTo->\{nA[\alpha]\}, ProjectedWith->\{hA[\alpha, -\beta]\}, PrintAs->'T'];$$

¹⁵ In line of this comment, let us note that `SetSlicing` also creates *internal* rules for the commutation of several \bar{D} . These will enforce the appearance of Laplacians, and move a covariant derivative closer to a tensor when a divergence is present.

$$\begin{aligned} In[25] &:= \text{LieD}[\text{nA}[\mu]] @ \text{cdA}[-v] @ \text{TA}[-\alpha] \\ Out[25] &:= \mathcal{C}^{\lambda}_{\nu}{}^{\beta} K_{\alpha\lambda} T_{\beta} - \mathcal{C}_{\alpha\nu}{}^{\lambda} K^{\beta}_{\lambda} T_{\beta} - \mathcal{C}_{\nu\alpha}{}^{\lambda} K^{\beta}_{\lambda} T_{\beta} + \mathcal{C}^{\lambda}_{\alpha}{}^{\beta} K_{\nu\lambda} T_{\beta} + \bar{D}_{\nu} \mathcal{L}_{\bar{n}\bar{A}} T_{\alpha} \end{aligned}$$

where to obtain $Out[25]$, expression (B.14) has been automatically applied to the extrinsic curvature tensor.

3. Perturbed fields

We now present the splitting of the perturbed fields into their spatial and temporal parts, the scalar–vector–tensor (SVT) decomposition of their projected components and finally the parameterization of the metric perturbations.

3.1. Projected components and scalar–vector–tensor decomposition

Any perturbed quantity lying within \mathcal{M} and mapped onto the background manifold can be decomposed by means of the normal vector \bar{n} and the induced metric \bar{h} . For a rank-2 covariant tensor T , we have for instance,

$$T_{\mu\nu} = \bar{n}_{\mu}\bar{n}_{\nu}(\bar{n}^{\rho}\bar{n}^{\sigma}T_{\rho\sigma}) + 2\bar{n}_{(\mu}(\bar{n}^{\rho}\bar{h}^{\sigma}_{\nu)}T_{\rho\sigma}) + (\bar{h}^{\rho}_{\mu}\bar{h}^{\sigma}_{\nu}T_{\rho\sigma}). \quad (34)$$

The terms inside parentheses hence constructed respectively define a scalar, a spatial vector and a spatial tensor. As is customary in perturbation theory, we shall then employ the SVT decomposition to further expand the two last projected quantities [18]. Let us briefly recall the form of this decomposition for later needs: a spatial vector U_{μ} is split into a scalar part S and a vector part V_{μ} as

$$U_{\mu} = \bar{D}_{\mu}S + V_{\mu}, \quad \text{with} \quad \bar{D}^{\mu}V_{\mu} = 0, \quad (35)$$

and a symmetric spatial tensor $H_{\mu\nu}$ is decomposed into two scalar parts S_1 and S_2 , a vector part V_{μ} and a tensor part $T_{\mu\nu}$ as

$$H_{\mu\nu} = \bar{D}_{\mu}\bar{D}_{\nu}S_1 + S_2\bar{h}_{\mu\nu} + \bar{D}_{(\mu}V_{\nu)} + T_{\mu\nu}, \quad \text{with} \quad \begin{cases} \bar{D}^{\mu}V_{\mu} = 0, \\ \bar{D}^{\mu}T_{\mu\nu} = 0 \end{cases} \text{ and } T^{\mu}_{\mu} = 0. \quad (36)$$

3.2. Perturbations of the metric

The SVT decomposition of the metric perturbations yields the general expressions¹⁶

$$\bar{n}^{\rho}\bar{n}^{\sigma}\langle^{(n)}h_{\rho\sigma} = -2\langle^{(n)}\phi, \quad (37)$$

$$\bar{n}^{\rho}\bar{h}^{\sigma}_{\nu}\langle^{(n)}h_{\rho\sigma} = -\bar{D}_{\nu}\langle^{(n)}B - \langle^{(n)}B_{\nu}, \quad (38)$$

$$\bar{h}^{\rho}_{\mu}\bar{h}^{\sigma}_{\nu}\langle^{(n)}h_{\rho\sigma} = 2(\bar{D}_{\mu}\bar{D}_{\nu}\langle^{(n)}E + \bar{D}_{(\mu}\langle^{(n)}E_{\nu)}) + \langle^{(n)}E_{\mu\nu} - \langle^{(n)}\psi\bar{h}_{\mu\nu}). \quad (39)$$

Four of the ten degrees of freedom of $\langle^{(n)}h$ are carried by the scalars $\langle^{(n)}\phi$, $\langle^{(n)}\psi$, $\langle^{(n)}E$ and $\langle^{(n)}B$, four are encoded in the vectors $\langle^{(n)}E_{\mu}$ and $\langle^{(n)}B_{\nu}$, and two are contained in the tensor $\langle^{(n)}E_{\mu\nu}$. Note that some of these fields are required to vanish for specific gauge choices (see, e.g., [19] for a comprehensive review).

¹⁶ Note that for Bianchi space–times it is more convenient to modify this decomposition by replacing the last term of equation (39) with $\langle^{(n)}\psi 3\bar{K}_{\mu\nu}/\bar{K}^{\alpha}_{\alpha}$ [25]. This boils down to a redefinition of $\langle^{(n)}E$, $\langle^{(n)}E_{\mu}$, $\langle^{(n)}E_{\mu\nu}$ and $\langle^{(n)}\psi$, and it is this parameterization that we have chosen in *xPand*.

3.3. Implementation in xPand

3.3.1. *Construction of SVT quantities.* The construction of spatial tensors satisfying the SVT properties is performed by the *xPand* function `DefProjectedTensor`. By default, these tensors are defined on both the background and perturbed manifolds. For instance, we have

```
In[26] := DefProjectedTensor[ U[-α,-β], h ];
In[27] := n[-α] U[α, β]
Out[27] := 0
In[28] := U[α, -α]
Out[28] := 0
In[29] := Perturbation[ U[-α,-β], 1 ]
Out[29] := Δ[Uαβ]
```

To relax one or several default properties, the user has to modify the optional arguments `TensorProperties`, set by default to `{‘SymmetricTensor’, ‘Traceless’, ‘Transverse’}`, and `SpaceTimesOfDefinition`, evaluated by default as `{‘Background’, ‘Perturbed’}`:

```
In[30] := UndefTensor[ U ];
In[31] := DefProjectedTensor[ U[-α,-β], h, TensorProperties->
    {‘SymmetricTensor’, ‘Transverse’},
    SpaceTimesOfDefinition->{‘Background’}];
In[32] := U[α,-α]
Out[32] := Uαα
In[33] := Perturbation[U[-α, -β], 1]
Out[33] := 0
```

3.3.2. *Comments on the label-indices.* In the internal notation, *xPert* attaches a label-index to the metric perturbations in order to denote their order. The first-order perturbation $dg_{\alpha\beta}^1$ (cf output *Out[5]*) is hence stored as `dg[LI[1], -α,-β]`. In *xPand*, we employ the same notation for spatial tensors defined with `DefProjectedTensor`, and we moreover attach a second label-index to indicate the number of Lie derivatives along \bar{n} acting on them. Thus, we have for instance

```
In[34] := DefProjectedTensor[V[-α], h]
In[35] := LieD[n[μ]]@V[LI[1], LI[0], -α]
Out[35] := (1)Vα'
In[36] := V[LI[1], LI[1], -α]
Out[36] := (1)Vα'
```

`DefProjectedTensor` constructs a set of rules to automatically allocate label-indices to a tensor written without. Hence, `V[-α]` is converted to `V[LI[0], LI[0], -α]`, while `V[LI[p], -α]` is converted into `V[LI[p], LI[0], -α]`, for any perturbation order `p`.

Note that *xTensor* interprets a tensor with label-indices as a tensor by itself, whatever the meaning of the label-indices. Its indices are therefore raised and lowered by the ambient metric in the usual way: `g[α,β] V[LI[1],LI[1],-α]` yields `V[LI[1], LI[1], β]`. While this is obviously mathematically correct for conformal isotropic manifolds¹⁷, it is no longer true for

¹⁷ In such cases, `g[α,β] V[LI[1],LI[1],-α]` corresponds to $g^{\alpha\beta} \mathcal{L}_{\bar{n}} V_{\alpha}$ which is equal to $\mathcal{L}_{\bar{n}} V^{\beta}$, as the extrinsic curvature vanishes. This last expression indeed corresponds to `V[LI[1], LI[1], β]`.

anisotropic ones. For these latter models, the second label-index can be interpreted as Lie derivatives *only* when the tensor is with indices *down*. To avoid confusion, we modify the output when the tensor indices are up. For instance, we have for ‘‘BianchiA’’ cosmologies

```
In[37] := DefProjectedTensor[VA[-α], hA]
In[38] := VA[LI[1], LI[1], -α]
Out[38] := (1)Vα'
In[39] := VA[LI[1], LI[1], α]
Out[39] := (1)Vα
```

3.3.3. Construction of the perturbations of the metric. The perturbed fields introduced in subsection 3.2 are constructed with the *xPand* function `DefMetricFields`. The evaluation of the command

```
In[40] := DefMetricFields[ g, dg, h ];
```

calls the function `DefProjectedTensor` in order to define the projected components of the metric perturbations and allocate them all the SVT properties. The set of rules (37)–(39) are then automatically defined using the *xPand* function `SplitMetric`:

```
In[41] := GaugeRules = SplitMetric[ g, dg, h, ‘‘AnyGauge’’ ]
In[42] := dg[LI[1], -α, -β] /. GaugeRules
Out[42] := 2 (1)Eαβ + (1)Bβ  $\bar{n}_\alpha$  - (1)Bα  $\bar{n}_\beta$  - 2  $\bar{n}_\alpha$   $\bar{n}_\beta$  (1)φ - 2  $\bar{h}_{\alpha\beta}$  (1)ψ -  $\bar{n}_\beta$   $\bar{D}_\alpha$  (1)B +  $\bar{D}_\alpha$  (1)Eβ
-  $\bar{n}_\alpha$   $\bar{D}_\beta$  (1)B +  $\bar{D}_\beta$  (1)Eα + 2  $\bar{D}_\beta$   $\bar{D}_\alpha$  (1)E
```

The particular gauges we have implemented are ‘‘ComovingGauge’’, ‘‘FlatGauge’’, ‘‘IsoDensityGauge’’, ‘‘NewtonGauge’’ and ‘‘SynchronousGauge’’. Should the user wish to consider other gauges, own rules can be created as follows: `MyGauge = {dg[LI[ord_], μ_, ν_] :> ...}`.

3.3.4. Splitting of the background covariant derivative. We now have nearly all the necessary tools to obtain the final expression of any perturbed field. Let us quickly review the previous steps and then introduce the last function we need.

To derive the perturbation of, e.g., the four-dimensional Ricci scalar, we first perform a conformal transformation from `g` to `gah2` (and express it in terms of the metric `g`), then we perturb the resulting expression, and finally we substitute the metric perturbations by their SVT components using the set of rules `GaugeRules`:

```
In[43] := Conformal[g, gah2][ RicciScalarCD[] ]
In[44] := MyR = ExpandPerturbation@Perturbed[ %, 1 ]
In[45] := (MyR /. GaugeRules) // ProjectorToMetric //
GradNormalToExtrinsicK // ContractMetric // ToCanonical
```

The final result has been expanded using the *xTensor* functions `ProjectorToMetric`, which replaces $\bar{h}_{\mu\nu}$ by $\bar{g}_{\mu\nu} - \bar{n}_\mu \bar{n}_\nu$, and `GradNormalToExtrinsicK`, which replaces $\bar{\nabla}_\mu \bar{n}_\nu$ by $\bar{K}_{\mu\nu}$, and it has been simplified with `ContractMetric` and `ToCanonical`.

Since the relation (32) is not automatically evaluated so far for quantities other than the three-curvature tensors, the result of `In[45]` still involves the covariant derivative $\bar{\nabla}$. In order to split the latter in terms of the induced derivative and Lie derivative along \bar{n} , one finally needs to use the *xPand* function `SplitPerturbations`, which applies as well the Gauss–Codazzi decompositions. In the above example, we obtain

```
In[46] := SplitPerturbations[% ah[]^2 , h]
Out[46] := 6 H^2 + 6 H' + 6 K + ε(-12 H^2(1)φ - 12 H'(1)φ - 6 H(1)φ' + 12(1)ψ K
          - 18 H(1)ψ' - 6(1)ψ'' + 6 H D_α D^α(1)B - 2 D_α D^α(1)B' + 6 H D_α D^α(1)E'
          + 2 D_α D^α(1)E'' - 2 D_α D^α(1)φ + 4 D_α D^α(1)ψ)
```

This way of proceeding is, however, rather inefficient for general gauges at higher order. Instead, the set of rules `GaugeRules` can be used in an optimized manner by the function `SplitPerturbations` itself:

```
In[47] := SplitPerturbations[ah[]^2 MyR, GaugeRules, h]
```

In such a way, the rule following relation (32) is evaluated on the projected components of the metric *before* specifying their SVT decomposition, which takes much less time at higher orders.

4. Features of the algorithm

4.1. Summary

Let us review the main steps that need to be followed in order to derive the perturbation of any fields.

- (i) We define the background manifold M (with `DefManifold`) and the ambient metric g (with `DefMetric`). The splitting of M is realized with `SetSlicing`, according to the type of cosmology chosen by the user.
- (ii) We apply to the quantity to be perturbed a conformal transformation from \bar{g} to \tilde{g} and express the result with respect to the quantities defined on M . This is done with the function `Conformal`.
- (iii) We use the *xPert* tools `Perturbed` and `ExpandPerturbation` to perturb the previous expression at any order, in terms of the metric perturbations dg and other tensors.
- (iv) We use the functions `DefMetricFields` and `SplitMetric` to define and construct the SVT parameterization of the metric perturbations (thanks to the function `DefTensorProjected`), according to the gauge chosen by the user. The perturbations of the fluid quantities are defined and decomposed with the functions `DefMatterFields` and `SplitMatter` (see further below).
- (v) We finally use the function `SplitPerturbations` to decompose the covariant derivative $\bar{\nabla}$ in terms of the induced derivative and the Lie derivative along \bar{n} . The Gauss–Codazzi relations are also applied, and the constructed rules are used to transform the resulting expression.

These four last steps have been coded all at once in a single function called `ToxPand`. We present it further below, after introducing a minimal example.

4.2. A minimal example

We now propose a brief and self-contained example to illustrate our package. The parameterization of the metric perturbations is here constructed by hand, and we only consider the Bardeen potentials:

```
<<xAct/xPand.m;
DefManifold[ M, 4, {α, β, μ, ν} ];
DefMetric[ -1, g[-α,-β], CD, {‘;’, ‘∇’} ];
DefMetricPerturbation[ g, dg, ε ];
```

```

SetSlicing[ g, n, h, cd, {‘|’, ‘D’}, ‘FLCurved’ ];
order = 1;
DefProjectedTensor[  $\phi$ [], h ];
DefProjectedTensor[  $\psi$ [], h ];
MyRicciScalar = ExpandPerturbation@Perturbed[ Conformal[g, gah2]
  [RicciScalarCD[]], order ];
MyGauge = dg[LI[ord_],  $\alpha$ _,  $\beta$ _] :> - 2 n[ $\alpha$ ]n[ $\beta$ ]  $\phi$ [LI[ord]] - 2 h[ $\alpha$ ,  $\beta$ ]
   $\psi$ [LI[ord]];
SplitPerturbations[ ah[]^2 MyRicciScalar, MyGauge, h ]

```

The output generated by the last line is (compare with *Out[46]*)

$$\begin{aligned}
\text{Out}[47] := & 6\mathcal{H}^2 + 6\mathcal{H}' + 6\mathcal{K} + \varepsilon(-12\mathcal{H}^{2(1)}\phi - 12\mathcal{H}'^{(1)}\phi - 6\mathcal{H}^{(1)}\phi' + 12^{(1)}\psi\mathcal{K} \\
& - 18\mathcal{H}^{(1)}\psi' - 6^{(1)}\psi'' - 2D_\alpha D^{\alpha(1)}\phi + 4D_\alpha D^{\alpha(1)}\psi)
\end{aligned}$$

4.3. Secondary functions

Even though the user is free to parameterize tensor perturbations by creating its own rules with projected tensors defined from `DefProjectedTensor`, we have seen in paragraph 3.3.3 that the functions `DefMetricFields` and `SplitMetric` can take care of this procedure for the metric perturbations. Similarly, we have implemented in *xPand* the functions `DefMatterFields` and `SplitMatter` for the definition and parameterization of the fluid perturbations, and more precisely for those of the energy density, pressure and fluid 4-velocity.

We also wish to mention that we have extended in *xPand* the *xPert* function `GaugeChange` which performs gauge transformations at any order for a given expression. Our extension `SplitGaugeChange` executes a 3 + 1 splitting of these transformation rules. For a glance at the gauge transformations for metric perturbations and the construction of gauge-invariant variables, see [15, 18, 20, 21]. For examples of use of the secondary functions `SplitMatter` and `SplitGaugeChange`, we invite the reader to go through the example notebooks which are distributed along with the package *xPand*.

Furthermore, for the cases where one wants to consider the predefined gauges, it is enough, simple and straightforward to use the *xPand* function `ToxPand` to obtain the perturbation of any expression. For instance, the following five lines suffice to derive that of the four-Ricci scalar in any gauge and up to order 2:

```

<<xAct/xPand.m;
DefManifold[ M, 4, { $\alpha$ ,  $\beta$ ,  $\mu$ ,  $\nu$ } ];
DefMetric[ -1, g[- $\alpha$ , - $\beta$ ], CD, {“,”, “ $\bar{\nabla}$ ”} ];
SetSlicing[ g, n, h, cd, {‘|’, ‘D’}, ‘FLCurved’ ];
ToxPand[ RicciScalarCD[], dg, u, du, h, ‘AnyGauge’, 2 ]

```

where `u` and `du` are the fluid 4-velocity and its perturbation, respectively. The function `ToxPand` combines several functions so that the user can easily obtain the desired perturbations in a given gauge without having to deal with any detail of the algorithm.

Finally, the *xPand* function `ExtractComponents` allows us to extract the projected components of any tensor. For a given background slicing, the user only needs to specify the type of projection (see appendix A for an example). Note also that for a rank-1 (resp. rank-2) tensor, the *xPand* function `VisualizeTensor` allows us to display all projected components in a vector (resp. matrix) form. Again, we refer to the example notebooks which are distributed along with *xPand* [5], for more details about the syntax of these functions.

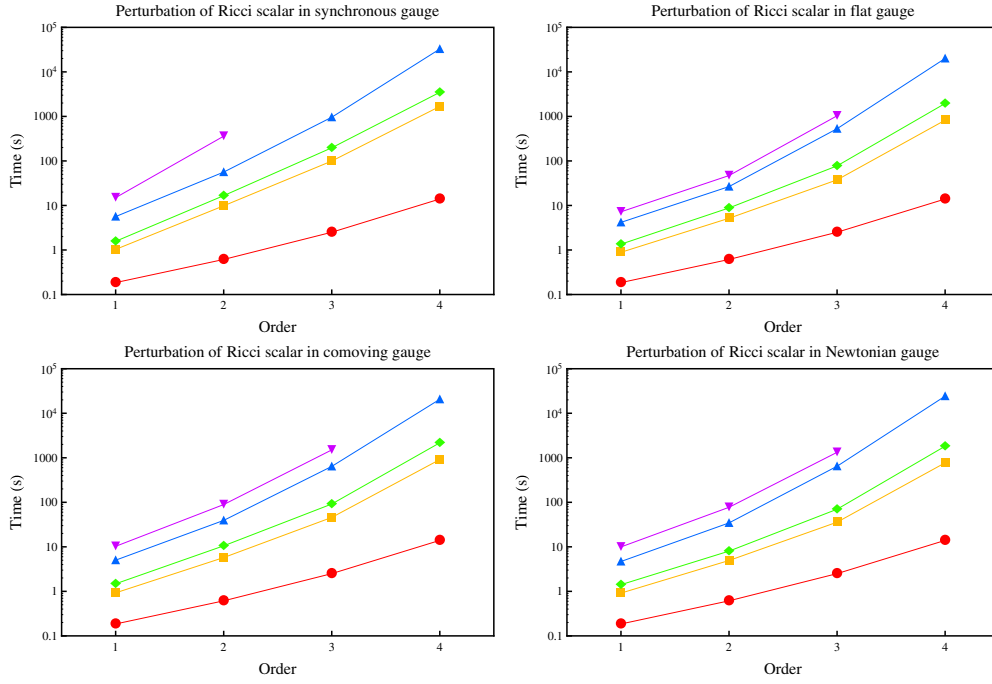


Figure 1. Timings for the perturbation of the four-dimensional Ricci scalar in different gauges. From top to bottom and left to right: synchronous gauge, spatially flat gauge, comoving gauge and Newtonian gauge. On each plot, the curves from bottom to top refer to: (i) formal perturbations with *xPert* using a conformal transformation (red line); (ii) perturbations for a Minkowski background (yellow); (iii) perturbations for a curved FLRW background (green); (iv) perturbations for a Bianchi I background (‘BianchiI’) (blue); and (v) perturbations for a general Bianchi background (‘BianchiB’) (purple). All timings were performed on a single 4 GHz core, with a 8 GB RAM.

4.4. Recovering standard results

We have checked that with our implementation we recover the standard results of cosmological perturbation theory. More precisely, our algorithm is in accordance with

- all first-order results of the Einstein equation and stress–energy tensor conservation equation, for flat and curved FLRW space–times in any gauge (see for instance [19]);
- all first-order results for Bianchi type-I background, in the gauge chosen in [22, 23];
- all second-order results of the Einstein equation and stress–energy tensor conservation equation, for flat and curved FLRW space–times in the Newtonian gauge (see for instance [15] for the complete set of equations).

Our package now enables to extend these results to higher order and in any gauge. It also allows for the study of any type of perturbed Bianchi cosmologies.

It is worth mentioning that in order to obtain useful standard differential equations with respect to the conformal time η , it is necessary to perform a mode expansion on the hypersurfaces, that is one needs to find the eigenmodes of the spatial Laplacian $\bar{D}_\mu \bar{D}^\mu$. This is simple for flat FLRW cosmologies, where one just has to employ a Fourier transformation, and it is also well known for curved FLRW models, where hyper-spherical Bessel functions need to be used [24]. However, the procedure is still unknown for general Bianchi cosmologies.

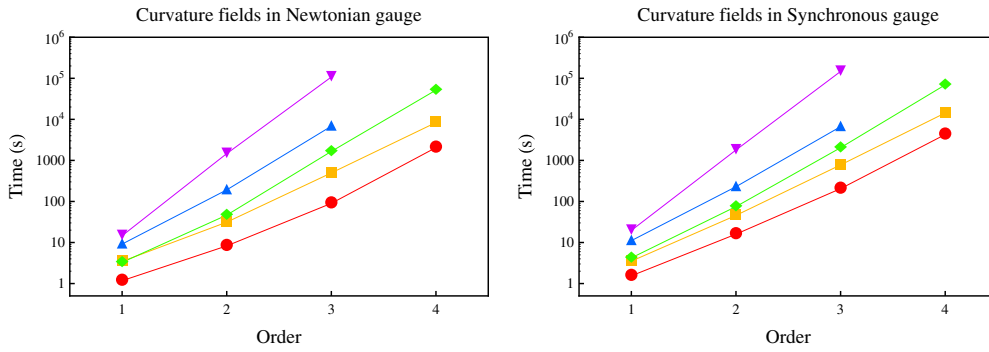


Figure 2. Timings for the perturbations of the four-curvature tensors, for a curved FLRW space–time, in different gauges. Left: Newtonian gauge; right: synchronous gauge. On each plot, the curves from bottom to top refer to: (i) Ricci scalar (red line); (ii) Ricci tensor (yellow); (iii) Einstein tensor (green); (iv) Riemann tensor (blue); and (v) Weyl tensor (purple). All timings were performed on a single 3.40 GHz core, with a 8 GB RAM.

Apart from the special case of Bianchi type I, where the modes can also be found from a Fourier transformation (and thus lead to a simple set of equations [25]), there is no *general* technique to obtain the eigenmodes of the Laplacian for all other types. Only in special cases (see for instance [26]) this has been done explicitly.

4.5. Timings

In practice, the timing for the computation grows like (slightly faster than) a power law of the perturbation order, whatever the gauge (see figure 1). It takes *xPand* less than 2 min (see figure 2) to decompose completely the perturbation of a rank-2 curvature tensor, such as the Ricci or Einstein tensor, up to second order in any gauge. The decomposition for the perturbations of the Riemann or Weyl tensor, up to second order and in any gauge, takes a little more than 2 and 13 min, respectively.

Conclusion

xPand is the first comprehensive package that allows one to perform algebraic calculations in cosmological perturbation theory for homogeneous background space–times, up to any order and in any gauge. It is worth stressing again the several features and advantages of this package.

- The expression of any perturbed field can be obtained for all Bianchi cosmologies, up to any order of interest and in any gauge, in a very simple and straightforward way.
- The package avoids the complexities of a component-by-component computation of the perturbed fields, thanks to the use of the $3 + 1$ formalism.
- The package is relatively fast: at first order, all the perturbed equations can be derived in approximately two seconds, and as the order of the perturbation increases, the timing grows roughly as a power law.
- The computations can be applied to space–times of arbitrary dimension, and for any extension of general relativity (the current exception being any gravitational theory with torsion).

- Taking advantage of the package *xTensor*, *xPand* handles tensor indices just the same way a user would do when calculating by hand. In particular, it does not break the summation over repeated indices (as e.g. in *Out[46]*).
- It totally eliminates the laborious summation over repeated indices, an element that has inhibited the use of other packages developed to solve similar models.
- The user who has little knowledge of the *Mathematica* or *xTensor* syntax can obtain from the package almost the same utility as an expert would do.

Finally, we plan to extend the scope of *xPand* beyond the derivation of perturbed equations in general relativity, so as to provide the entire Einstein–Boltzmann system for radiation transfer and the Einstein–Jacobi map system needed for understanding the effect of weak gravitational lensing.

Acknowledgments

It is a pleasure to thank José M Martín-García and Guillaume Faye for long and fruitful discussions about abstract tensor calculus in *xAct*. We also thank Chris Clarkson for comments on the SVT decomposition, and José M Martín-García for comments on a previous version of this paper. XR acknowledges support from the Claude Leon Foundation. He is grateful for the hospitality received in the cosmology group at ‘*Institut d’Astrophysique de Paris*’, and thanks Jean-Philippe Uzan for his invitation. OU is funded by the South African Square Kilometer Array (SKA) Project and Faculty of Science, University of Cape Town. This work was supported by French state funds managed by the ANR within the Investissements d’Avenir programme under reference ANR-11-IDEX-0004-02.

Appendix A. What *xPand* can do: a non-trivial example

We here described by way of an example how *xPand* can be used to derive relativistic perturbed equations for flat FLRW backgrounds, up to first order and in the Newtonian gauge. An extension to any other background, order or gauge simply requires a change of the related arguments in the following example.

As usual, we define the geometry of the model by evaluating the following commands:

```
In[1] := <<xAct/xPand.m;
        DefManifold[ M, 4, {α, β, μ, ν} ];

In[3] := DefMetric[ -1, g[-μ, -ν], CD, {‘;’, ‘CD’} ];
        DefMetricPerturbation[ g, dg, ε ];

In[5] := SetSlicing[ g, n, h, cd, {‘|’, ‘D’}, ‘FLFlat’ ];
        order = 1;
```

For simplicity, we customize the use of the command *ToxPand* as

```
In[7] := MyToxPand[ expr_, gauge_, order_ ] := ToxPand[ expr, dg, u, du,
        h, gauge, order ];
```

We define the constants $\kappa = 8\pi G$ and Λ with

```
In[8] := DefConstantSymbol[ κ ];
        DefConstantSymbol[ Λ ];
```

and then the energy–momentum tensor of the fluid:

```
In[10] := DefTensor[ T[-μ, -ν], M ]
```

We introduce a boolean variable to switch on or off the pressure of the fluid filling the space-time. This allows us to consider either dust- or radiation-dominated era in a simple way:

$$\begin{aligned} \text{In}[11] &:= \$\text{Dust} = \text{True}; \\ \text{In}[12] &:= \text{IndexSet}[\text{T}[\alpha_,\beta_], (\rho\text{u}[] + \text{If}[\$\text{Dust}, 0, \text{Pu}[]]) \text{u}[\alpha]\text{u}[\beta] \\ &\quad + \text{If}[\$\text{Dust}, 0, \text{Pu}[]] \text{g}[\alpha,\beta]]; \end{aligned}$$

The fields ρu (energy density), Pu (pressure) and u (fluid 4-velocity) will be automatically created later, when calling the function `(My)ToxPand`. The user is encouraged to set the boolean variable to `False` in order to include the effects of pressure.

We now derive the perturbations of Einstein equations for the model at stake. We define the Einstein equations as

$$\text{In}[13] := \text{MyGR}[\mu_,\nu_] := \text{EinsteinCD}[\mu, \nu] + \text{g}[\mu,\nu] \Lambda/\kappa - \kappa \text{T}[\mu,\nu];$$

and we evaluate their perturbations simply using

$$\text{In}[14] := \text{MyGRresult} = \text{MyToxPand}[\text{MyGR}[\mu, -\nu], \text{‘NewtonGauge’}, \text{order}];$$

Since the computation of this expression is the time-consuming part, we have stored its value in the variable `MyGRresult`.

The background components of the resulting expression are extracted using the `xPand` function `ExtractOrder[expr, 0]` (where 0 indicates the background value), and the projected components with the `xPand` function `ExtractComponents`:

$$\text{In}[15] := \text{ExtractComponents}[\text{ExtractOrder}[\text{ah}[]^2 \text{MyGRresult}, 0], \text{h}, \{ \text{‘Time’}, \text{‘Time’} \}]$$

$$\text{Out}[15] := \frac{\Lambda a^2}{\kappa} - 3\mathcal{H}^2 + \kappa a^2 \rho$$

$$\text{In}[16] := \text{ExtractComponents}[\text{ExtractOrder}[\text{ah}[]^2 \text{MyGRresult}, 0], \text{h}, \{ \text{‘Space’}, \text{‘Space’} \}]$$

$$\text{Out}[16] := \frac{\Lambda a^2 h^\mu{}_\nu}{\kappa} - h^\mu{}_\nu \mathcal{H}^2 - 2 h^\mu{}_\nu \mathcal{H}'$$

Similarly, the first-order equations are obtained by evaluating

$$\text{In}[17] := \text{ExtractComponents}[\text{ExtractOrder}[\text{ah}[]^2 \text{MyGRresult}, 1], \text{h}, \{ \text{‘Time’}, \text{‘Time’} \}]$$

$$\text{Out}[17] := \kappa a^{2(1)}\rho + 6\mathcal{H}^{2(1)}\phi + 6\mathcal{H}^{(1)}\psi' - 2D_\alpha D^{\alpha(1)}\phi$$

$$\text{In}[18] := \text{ExtractComponents}[\text{ExtractOrder}[\text{ah}[]^2 \text{MyGRresult}, 1], \text{h}, \{ \text{‘Time’}, \text{‘Space’} \}]$$

$$\text{Out}[18] := -\kappa a^{2(1)}B_\nu \rho - \kappa a^{2(1)}V_\nu \rho + \frac{1}{2}D_\alpha D^{\alpha(1)}B_\nu - \kappa a^2 \rho D_\nu^{(1)}V - 2\mathcal{H}D_\nu^{(1)}\phi - 2D_\nu^{(1)}\psi'$$

$$\text{In}[19] := \text{ExtractComponents}[\text{ExtractOrder}[\text{ah}[]^2 \text{MyGRresult}, 1], \text{h}, \{ \text{‘Space’}, \text{‘Time’} \}]$$

$$\text{Out}[19] := 2^{(1)}B^\mu \mathcal{H}^2 - 2^{(1)}B^\mu \mathcal{H}' + \kappa a^{2(1)}\mathcal{V}^\mu \rho + D_\alpha^{(1)}E'^{\mu\alpha} - \frac{1}{2} D_\alpha D^\alpha{}^{(1)}B^\mu + \kappa a^2 \rho D^\mu{}^{(1)}\mathcal{V} \\ + 2 \mathcal{H} D^\mu{}^{(1)}\phi + 2 D^\mu{}^{(1)}\psi'$$

$$\text{In}[20] := \text{ExtractComponents}[\text{ExtractOrder}[\text{ah}[]^2 \text{MyGRresult}, 1], \text{h}, \\ \{ \text{'Space'}, \text{'Space'} \}]$$

$$\text{Out}[20] := {}^{(1)}E''^\mu{}_\nu + {}^{(1)}E'^\mu{}_\nu \mathcal{H} + 2 h^\mu{}_\nu \mathcal{H}^2{}^{(1)}\phi + 4 h^\mu{}_\nu \mathcal{H}'{}^{(1)}\phi' + 2 h^\mu{}_\nu \mathcal{H}{}^{(1)}\phi' \\ + 4 h^\mu{}_\nu \mathcal{H}{}^{(1)}\psi' + 2 h^\mu{}_\nu {}^{(1)}\psi'' \\ - D_\alpha D^\alpha{}^{(1)}E^\mu{}_\nu + h^\mu{}_\nu D_\alpha D^\alpha{}^{(1)}\phi - h^\mu{}_\nu D_\alpha D^\alpha{}^{(1)}\psi - \mathcal{H} D^\mu{}^{(1)}B_\nu - \frac{1}{2} D^\mu{}^{(1)}B'_\nu \\ - \mathcal{H} D_\nu{}^{(1)}B^\mu - \frac{1}{2} D_\nu{}^{(1)}B'^\mu + D_\nu D^\mu{}^{(1)}\phi + D_\nu D^\mu{}^{(1)}\psi$$

Appendix B. Bianchi cosmologies

B.1. Constructing a four-dimensional basis

We here review the general properties of Bianchi space–times. The reader can find a detailed presentation of their classification in [17], and summaries in, e.g., [27, 28].

Bianchi space–times possess by definition a set of (three-dimensional) homogeneous hypersurfaces. One can therefore introduce three linearly independent spatial Killing vector fields (KVF) ξ_i , with $i \in \{1, 2, 3\}$, satisfying

$$\mathcal{L}_{\xi_i} \bar{g}_{\mu\nu} = 0 \Leftrightarrow \bar{\nabla}_{(\mu} \xi_{i\nu)} = 0, \quad \bar{n}_\mu \xi_i^\mu = 0. \quad (\text{B.1})$$

From these properties together with equation (23), we obtain

$$\mathcal{L}_{\bar{n}} \xi_i = [\bar{n}, \xi_i] = 0. \quad (\text{B.2})$$

The nature of the Bianchi model is determined by the spatial structure coefficients $C^k{}_{ij}$, defined from the commutators of the KVF:

$$[\xi_i, \xi_j] \equiv C^k{}_{ij} \xi_k, \quad \text{with } C^k{}_{ij} = -C^k{}_{ji}. \quad (\text{B.3})$$

The Jacobi identity

$$[\xi_i, [\xi_j, \xi_k]] + [\xi_j, [\xi_k, \xi_i]] + [\xi_k, [\xi_i, \xi_j]] = 0, \quad (\text{B.4})$$

constrains these constants to verify

$$C^m{}_{[ij} C^l{}_{k]m} = 0 \Rightarrow C^m{}_{ij} C^l{}_{lm} = 0, \quad (\text{B.5})$$

where we have used the fact that the $C^k{}_{ij}$ are constant on spatial slices.

We now construct, on a given hypersurface, a vector basis $\{\mathbf{e}_i\}$ and its dual $\{\mathbf{e}^i\}$ invariant under the action of the KVF, namely satisfying

$$\mathcal{L}_{\xi_i} \mathbf{e}_j = [\xi_i, \mathbf{e}_j] = 0, \quad \mathcal{L}_{\xi_i} \mathbf{e}^j = 0. \quad (\text{B.6})$$

From these properties along with relation (B.3), we infer that the \mathbf{e}_i can be chosen such that

$$[\mathbf{e}_i, \mathbf{e}_j] = C^k{}_{ij} \mathbf{e}_k, \quad 2e_i^\mu e_j^\nu \nabla_{[\mu} e^k{}_{\nu]} = -C^k{}_{ij}. \quad (\text{B.7})$$

The constants of structure can be further developed in terms of a symmetric ‘tensor’ N^{ij} and a ‘vector’ A^i as

$$C^k{}_{ij} = \epsilon_{ijm} N^{mk} + 2A_{[i} \delta^k{}_{j]} \quad (\text{B.8})$$

where ϵ_{ijm} denotes the totally anti-symmetric Levi-Civita symbol. Note that the Jacobi identity (B.5) translates in that case to the simple relation

$$N_{ij} A^j = 0. \quad (\text{B.9})$$

We then extend the bases $\{\mathbf{e}_i\}$ and $\{\mathbf{e}^i\}$ to the whole space–time by Lie dragging them with $\bar{\mathbf{n}}$, which implies the properties

$$\mathcal{L}_{\bar{\mathbf{n}}}\mathbf{e}_i = [\bar{\mathbf{n}}, \mathbf{e}_i] = 0, \quad \mathcal{L}_{\bar{\mathbf{n}}}\mathbf{e}^i = 0. \tag{B.10}$$

With the above procedure, we are able to construct a four-dimensional basis $\{\mathbf{e}_a\} \equiv \{\bar{\mathbf{n}}, \mathbf{e}_i\}$ along with its dual $\{\mathbf{e}^a\} \equiv \{\bar{\mathbf{n}}, \mathbf{e}^i\}$ (where $\bar{\mathbf{n}}$ is the dual form of $\bar{\mathbf{n}}$ and $a \in \{0, 1, 2, 3\}$), that are invariant under the action of the KVF. The commutation relations of these new bases simply follow from expressions (B.7) and (B.10): the structure coefficients C^c_{ab} vanish when any of the indices is zero and take the values C^k_{ij} otherwise. This method of building a four-dimensional basis out of a three-dimensional one defined on a given spatial hypersurface is the simplest one¹⁸.

B.2. Expression of the geometrical tensors

The components of the induced metric are written with respect to the Bianchi basis $\{\mathbf{e}_i\}$ and its dual as

$$\bar{h}_{\mu\nu} = \bar{h}_{ij}e^i{}_{\mu}e^j{}_{\nu}. \tag{B.11}$$

From relations (B.1) and (B.2), we deduce

$$\mathcal{L}_{\bar{\xi}}\bar{h}_{\mu\nu} = 0, \tag{B.12}$$

and using equation (B.11) together with (B.6), we obtain: $\mathbf{e}_k(\bar{h}_{ij}) = 0$. The components \bar{h}_{ij} are thus only time-dependent, namely $\bar{h}_{ij} = \bar{h}_{ij}(\eta)$. Any tensor field T lying on the background space–time possesses the same symmetries, and so we can expand it with respect to the Bianchi bases as

$$\bar{T}_{\mu_1\dots\mu_p} = \bar{T}_{i_1\dots i_p}(\eta)e^{i_1}{}_{\mu_1}\dots e^{i_p}{}_{\mu_p}. \tag{B.13}$$

Denoting by ${}^3\bar{\Gamma}_{ijk}$ the coefficients of the connection \bar{D} in the Bianchi bases, we then deduce that

$$\bar{D}_k\bar{T}_{i_1\dots i_p} \equiv e_k{}^{\alpha}e_{i_1}{}^{\mu_1}\dots e_{i_p}{}^{\mu_p}\bar{D}_{\alpha}\bar{T}_{\mu_1\dots\mu_p} = -\sum_{j=1}^p {}^3\bar{\Gamma}^q{}_{kj}\bar{T}_{i_1\dots i_{j-1}qi_{j+1}\dots i_p}. \tag{B.14}$$

This relation is used in *xPand* to compute the induced covariant derivative of any background tensor, such as the extrinsic curvature. The connection coefficients associated with the background connection $\bar{\nabla}$ are given in the bases $\{\mathbf{e}_a\}$ and $\{\mathbf{e}^a\}$ by

$$\bar{\Gamma}_{abc} = \frac{1}{2}(-\mathbf{e}_a(\bar{g}_{bc}) + \mathbf{e}_b(\bar{g}_{ca}) + \mathbf{e}_c(\bar{g}_{ab}) + C_{abc} - C_{bca} + C_{cab}). \tag{B.15}$$

Given that the components \bar{h}_{ij} only depend on η , we deduce that the spatial-connection coefficients ${}^3\bar{\Gamma}_{ijk}$ are expressed only in terms of the constants of structure. We have indeed

$${}^3\bar{\Gamma}_{ijk} = \bar{\Gamma}_{ijk} = \frac{1}{2}(C_{ijk} - C_{jki} + C_{kij}), \quad C_{ijk} = \bar{\Gamma}_{ijk} - \bar{\Gamma}_{ikj}. \tag{B.16}$$

Note that the tensor indices in the bases $\{\mathbf{e}_i\}$ and $\{\mathbf{e}^i\}$ are lowered with \bar{h}_{ij} and raised with its inverse \bar{h}^{ij} , so that for instance $C_{kij} \equiv \bar{h}_{km}C^m{}_{ij}$. Note also that the constants of structure of the

¹⁸ Note that in this framework, only $\bar{\mathbf{n}}$ is a unit vector. An alternative approach consists in building a basis of vectors that all are unitary, by renormalizing the \mathbf{e}_i . However, by doing so, the spatial vectors hence constructed do not commute with $\bar{\mathbf{n}}$ anymore, and their associated structure coefficients become time dependent. We shall not consider such possibility in this paper, but details can be found in, e.g., [17, 27, 28].

basis $\{\mathbf{e}_i\}$, that we here denote $C[\mathbf{e}]_{ij}^k$, for the sake of clarity, are the components, in this specific basis, of a tensor. The associated components in a general basis can be recovered from¹⁹

$$C[\mathbf{e}]_{\mu\nu}^\alpha \equiv C[\mathbf{e}]_{ij}^k e_k^\alpha e^\mu e^\nu. \quad (\text{B.17})$$

Similarly, the components of N_{ij} and A_i in a general basis are found from

$$N[\mathbf{e}]_{\mu\nu} \equiv N[\mathbf{e}]_{ij} e^\mu e^\nu, \quad A[\mathbf{e}]_\mu \equiv A[\mathbf{e}]_i e^\mu. \quad (\text{B.18})$$

Relation (B.14) is also used to compute the induced covariant derivative of these three tensors, since they also live on the background space–time. Note, finally, that from equation (B.10) and the fact that the C_{jk}^i are constant, we obtain the useful relation

$$\mathcal{L}_{\bar{n}}(C_{\mu\nu}^\alpha) = 0. \quad (\text{B.19})$$

The three-Riemann tensor of the hypersurfaces can be expressed only in terms of the constants of structure. In the bases $\{\mathbf{e}_i\}$ and $\{\mathbf{e}^i\}$, its components are given by

$$\begin{aligned} {}^3\bar{R}_{ij}{}^{kl} = & -\frac{1}{2}C_{ij}^p C_p{}^{kl} + \frac{1}{2}C_{pi}{}^l C^{pk}{}_j + C_{pj}{}^l C_i{}^{kp} + C_{pj}{}^l C_i{}^{kp} + C_{ijp} C^{pkl} + \frac{1}{2}C_{ip}{}^l C_j{}^{kp} + \frac{1}{2}C_{ip}{}^l C_j{}^{kp} \\ & + C_{jp}{}^k C_i{}^{lp}, \end{aligned} \quad (\text{B.20})$$

where a double anti-symmetrization $[ij]$ and $[kl]$ is implied on the indices on the right-hand side. The three-Ricci tensor and three-Ricci scalar can then be deduced, and we obtain

$${}^3\bar{R}_{ij} = -\frac{1}{2}C_{kil} C_j{}^{kl} - \frac{1}{2}C_{kil} C_j{}^{kl} + \frac{1}{4}C_i{}^{kl} C_{jkl} + C_{(ij)}{}^p C_{pk}, \quad (\text{B.21})$$

$${}^3\bar{R} = -\frac{1}{4}C_{ijk} C^{ijk} - \frac{1}{2}C_{ijk} C^{jik} + C^{kj}{}_k C^p{}_{pj}. \quad (\text{B.22})$$

Note that owing to the Jacobi identity (B.5), the three-curvature tensors can take several equivalent forms. Finally, due to the homogeneity of the hypersurfaces, any induced derivative acting on the three-curvature tensors can be computed using equation (B.14). All the rules of this appendix, namely equations (B.14), (B.16), (B.20), (B.21) and (B.22), are automatically created in *xPand* when calling the function `SetSlicing`, in case the specified space type is a Bianchi space–time. The boolean variable `$OpenConstantsOfStructure` controls whether or not the constants of structure should be opened in the final expressions using the parameterization (B.8).

Appendix C. Principal commands of *xPand*

Table C1. Information about these commands can be obtained within *Mathematica* by evaluating `?NameOfCommand`.

| | | |
|-------------------------------------|---------------------------------|--------------------------------|
| <code>Conformal</code> | <code>ConformalWeight</code> | <code>DefMatterFields</code> |
| <code>DefMetricFields</code> | <code>DefProjectedTensor</code> | <code>ExtractComponents</code> |
| <code>SetSlicing</code> | <code>SplitGaugeChange</code> | <code>SplitMatter</code> |
| <code>SplitMetric</code> | <code>SplitPerturbations</code> | <code>ToBianchiType</code> |
| <code>ToConstantsOfStructure</code> | <code>ToxPand</code> | <code>VisualizeTensor</code> |

¹⁹ Note that we are *not* considering the constants of structure of another basis with this construction. We rather build a tensor $C[\mathbf{e}] \equiv C[\mathbf{e}]_{ij}^k \mathbf{e}_k \otimes \mathbf{e}^i \otimes \mathbf{e}^j$. It is obvious that the components of this tensor in the basis $\{\mathbf{e}_i\}$ and $\{\mathbf{e}^i\}$ are the $C[\mathbf{e}]_{ij}^k$, but its components can also be taken in a general basis even though we only refer to the commutation structure of the \mathbf{e}_i . Since *xTensor* manipulates only abstract indices (that is, indices in a general basis), this covariant point of view is necessary to implement the structure of the Bianchi space–times in our package.

References

- [1] Hinshaw G *et al* 2012 arXiv:1212.5226 [astro-ph.CO]
- [2] Sievers J L *et al* 2013 arXiv:1301.0824 [astro-ph.CO]
- [3] Araujo D *et al* 2012 *Astrophys. J.* **760** 145 (arXiv:1207.5034 [astro-ph.CO])
- [4] Musgrave P, Pollney D and Lake K 1996 GRTensor <http://grtensor.phy.queensu.ca>
- [5] Pitrou C, Roy X and Umeh O 2012 *xPand*: perturbations Are Not Difficult (www.xact.es/xPand/)
- [6] Martín-García J M 2004 *xAct*: efficient tensor computer algebra for mathematica (www.xact.es)
- [7] Brizuela D, Martín-García J M and Mena Marugan G A 2009 *xPert*: computer algebra for metric perturbation theory (www.xact.es/xPert arXiv:0807.0824 [gr-qc])
- [8] Brizuela D, Martín-García J M and Mena Marugan G A 2006 *Phys. Rev. D* **74** 044039 (arXiv:gr-qc/0607025)
- [9] Brizuela D, Martín-García J M and Tiglio M 2009 *Phys. Rev. D* **80** 024021 (arXiv:0903.1134 [gr-qc])
- [10] Brizuela D, Martín-García J M, Sperhake U and Kokkotas K D 2010 *Phys. Rev. D* **82** 104039 (arXiv:1009.5605 [gr-qc])
- [11] Arnowitt R L, Deser S and Misner C W 2004 The dynamics of general relativity arXiv:gr-qc/0405109[gr-qc]
- [12]ourgoulhon E 2012 3 + 1 *Formalism in General Relativity: Bases of Numerical Relativity (Lecture Notes in Physics* vol 846) (Berlin: Springer) (arXiv:gr-qc/0703035[gr-qc])
- [13] Smarr L and York J W 1978 *Phys. Rev. D* **17** 2529
- [14] Bruni M, Matarrese S, Mollerach S and Sonego S 1997 *Class. Quantum Grav.* **14** 2585
- [15] Nakamura K 2007 *Prog. Theor. Phys.* **117** 17 (arXiv:gr-qc/0605108 [gr-qc])
- [16] Wald R M 1984 *General Relativity* (Chicago: University of Chicago Press)
- [17] Ellis G F R and MacCallum M A H 1969 *Commun. Math. Phys.* **12** 108
- [18] Bardeen J M 1980 *Phys. Rev. D* **22** 1882
- [19] Malik K A and Wands D 2009 *Phys. Rep.* **475** 1 (arXiv:0809.4944 [astro-ph])
- [20] Bruni M, Matarrese S, Mollerach S and Sonego S 1997 *Class. Quantum Grav.* **14** 2585 (arXiv:gr-qc/9609040 [gr-qc])
- [21] Malik K A 2001 *PhD Thesis* University of Portsmouth (arXiv:astro-ph/0101563 [astro-ph])
- [22] Pereira T S, Pitrou C and Uzan J-P 2007 *J. Cosmol. Astropart. Phys.* **JCAP09(2007)006** (arXiv:0707.0736 [astro-ph])
- [23] Dulaney T R and Gresham M I 2010 *Phys. Rev. D* **81** 103532 (arXiv:1001.2301 [astro-ph.CO])
- [24] Abbott L and Schaefer R K 1986 *Astrophys. J.* **308** 546
- [25] Pitrou C, Pereira T S and Uzan J-P 2008 *J. Cosmol. Astropart. Phys.* **JCAP04(2008)004** (arXiv:0801.3596 [astro-ph])
- [26] Pereira T S, Carneiro S and Marugan G A M 2012 *J. Cosmol. Astropart. Phys.* **JCAP05(2012)040** (arXiv:1203.2072 [astro-ph.CO])
- [27] Ellis G F and van Elst H 1999 *NATO Adv. Stud. Inst. C* **541** 1 (arXiv:gr-qc/9812046 [gr-qc])
- [28] Pontzen A and Challinor A 2007 *Mon. Not. R. Astron. Soc.* **380** 1387 (arXiv:0706.2075 [astro-ph])