xPert: Computer algebra for metric perturbation theory

David Brizuela[1], José M. Martín-García[1,2], and Guillermo A. Mena Marugán[1]

[1]*Instituto de Estructura de la Materia, CSIC, Serrano 121-123, Madrid 28006, Spain and*
[2]*Matematiska institutionen, Linköpings universitet, Linköping, Sweden S-581 83*

We present the tensor computer algebra package *xPert* for fast construction and manipulation of the equations of metric perturbation theory, around arbitrary backgrounds. It is based on the combination of explicit combinatorial formulas for the n-th order perturbation of curvature tensors and their gauge changes, and the use of highly efficient techniques of index canonicalization, provided by the underlying tensor system *xAct*, for *Mathematica*. We give examples of use and show the efficiency of the system with timings plots: it is possible to handle orders $n = 4$ or $n = 5$ within seconds, or reach $n = 10$ with timings below 1 hour.

I. INTRODUCTION

Linear perturbation theory is one of the most successful techniques to extract information from the equations of general relativity. It allows us to study the evolution of small deviations with respect to a known background solution of the equations, showing the stability properties of that background or how it interacts with weak surrounding fields. Linearization is, however, the main limitation of the approach, because we loose the intrinsic nonlinear character of general relativity, and hence it is not possible to model the self-interaction or mutual interaction among the linear modes, and other interesting phenomena. Obviously, this could be cured by adding a second perturbative order to the analysis, but the large size of the resulting perturbative equations has generally prevented practical computations until very recently.

Already in 1980, Cunningham, Price and Moncrief studied and simulated second-order perturbations of collapse of a ball of dust (Oppenheimer-Snyder collapse), including the matching to perturbations of an exterior vacuum [1]. Since then, other backgrounds have been perturbed at second order. The oscillation modes of a rotating fluid star have been modelled at second order, introducing rotation at linear level [2, 3]. Teukolsky's formalism [4] for linear perturbations of a Kerr black hole was generalized to second order in [5]. The gravitational radiation emitted in a close limit head-on collision of two black holes has also been studied at second order, describing the final hole as a first-order quadrupolar perturbation of the Schwarzschild spacetime [6, 7]. Recently, also perturbing Schwarzschild, second-order quasinormal modes have been defined [8, 9]. Second-order perturbations have been considered in several cosmological scenarios, often taking Friedmann-Robertson-Walker as background spacetime; see for example [10, 11, 12] and references therein. We can also find high-order perturbations in various areas of field theory, like the problem of renormalization of a Lagrangian [13], though in this context the formalism used is slightly different. We shall later see how the general scheme can be adapted to include this as a particular case.

In references [14, 15] we have constructed the arbitrary-order generalization of the Gerlach and Sengupta formalism for gauge-invariant and covariant first-order perturbations of any spherical spacetime [16], for any matter model. This is the most convenient framework to study nearly-spherical astrophysical scenarios, like collapse of slowly-rotating stars, and some types of supernova explosions. The large size of the equations involved forced us to develop specialized tools of tensor computer algebra, which were summarized in an appendix of [14]. We have now improved these tools, extending their domain of applicability, increasing their efficiency, simplifying their use and fully documenting them.

We present version 1.0 of the package *xPert* for high-order metric perturbation theory. It is based on the combination of adapted mathematical algorithms and powerful techniques of tensor computer algebra. On the one hand, we use explicit precomputed formulas for the general terms of the perturbative expansions of the most important curvature tensors in differential geometry, avoiding slow recursive computations. On the other hand, *xPert* uses very efficient canonicalization algorithms for tensor computer algebra in *Mathematica*. This allows us to perform computations up to perturbative order 10 in several minutes with small computer resources, as we will see in some examples.

The article is organized as follows. Section II reviews the basic concepts of high-order metric perturbation theory and presents closed formulas to obtain the n-th perturbation of the most relevant geometric tensors. Section III explains how these formulas have been implemented in *Mathematica* and lists the main commands of the package *xPert*. Finally, we show the high efficiency of the system through some timing examples in section IV. Section V contains our conclusions.

II. HIGH-ORDER METRIC PERTURBATION THEORY

A. Basic concepts

Let us assume a family of N-dimensional manifolds $\overline{\mathcal{M}}(\varepsilon)$ equipped with a metric field $\overline{g}(\varepsilon)$ depending on a dimensionless parameter ε. This dependence will be taken to be smooth, so that all derivatives with respect to ε are well defined. Members with parameter $\varepsilon = 0$ will be referred to as *background* members and will be denoted without bar, that is $\mathcal{M} \equiv \overline{\mathcal{M}}(0), g \equiv \overline{g}(0)$. The Levi-Civita connection of g will be called ∇, and will be denoted with a semicolon in indexed expressions.

The goal in perturbation theory is comparing a tensor field $\overline{T}(\varepsilon)$ with its background counterpart T. The former is defined on the manifold $\overline{\mathcal{M}}(\varepsilon)$ and the latter on the background manifold $\mathcal{M}$ and therefore we have to introduce a family of point-to-point identification mappings (or *gauge*) ϕ_ε between them to compare. The pull-back $\phi_\varepsilon^* \overline{T}(\varepsilon)$ to the background can be now expanded as

$$\phi_\varepsilon^* \overline{T}(\varepsilon) \equiv T + \sum_{n=1}^{\infty} \frac{\varepsilon^n}{n!} \Delta_\phi^n[T], \tag{1}$$

with all terms in the equation being defined on the background manifold. We have introduced the perturbative operator Δ_ϕ in the gauge ϕ_ε, which is actually a derivative:

$$\Delta_\phi^n[T] \equiv \left. \frac{d^n \, \phi_\varepsilon^* \overline{T}(\varepsilon)}{d\varepsilon^n} \right|_{\varepsilon=0}. \tag{2}$$

From now on we shall omit the explicit dependence of perturbations on the chosen gauge ϕ_ε, to alleviate the notation. ε-dependent (barred) objects will be also assumed to be pulled-back to the background manifold. The theory and formulas for perturbations of changes of gauge have been fully developed for tensors by Bruni and collaborators [17], and extended in [18] to perturbations of distribution functions in phase space.

An important consequence of the derivative character of Δ is that it obeys the Leibnitz rule, whose n-th order generalization on the product of m tensors is

$$\Delta^n[T_1 \ldots T_m] = \sum_{\{k_i\}} \frac{n!}{k_1! \ldots k_m!} \Delta^{k_1}[T_1] \ldots \Delta^{k_m}[T_m], \tag{3}$$

where the sum extends to all sorted partitions of n in m nonnegative integers (that is, including zero) obeying $k_1 + \ldots + k_m = n$. It is possible to generalize this formula to the case $\Delta^n[F(S_1, ..., S_m)]$ for an arbitrary function F of m scalar arguments $S_1, ..., S_m$ using the Faà di Bruno formula [19] (the n-th order chain rule), generalized to multiple arguments.

We want to do metric perturbation theory, meaning that our fundamental object to be perturbed is the metric field g_{ab} (abstract indices in any of the tangent spaces of the $\overline{\mathcal{M}}$ manifolds will be lowercase Latin indices). It is convenient to name the n-th order perturbation of the metric as ${}^{\{n\}}h_{ab} \equiv \Delta^n[g_{ab}]$, with a left-superindex denoting the perturbative order. In this notation, the perturbed metric is given by

$$\overline{g}_{ab}(\varepsilon) = g_{ab} + \sum_{n=1}^{\infty} \frac{\varepsilon^n}{n!} \, {}^{\{n\}}h_{ab}. \tag{4}$$

With nonvanishing perturbations of the background metric, it is essential to carefully keep track of index positions. For example we have $\Delta[v^a] \neq g^{ab}\Delta[v_b]$ for a tensor field v^a. The notation Δv^a could be ambiguous and therefore we use square brackets in our notation for perturbations to avoid problems. Should we find the perturbation of a tensor with inconvenient index positions, we shall introduce metric factors to correct those positions and then use the Leibnitz rule (3).

An important variation of this general perturbative formalism is frequently used in quantum field theory in curved backgrounds. It is usually called 'background field method' and decomposes the full metric $\overline{g}$ as $\overline{g}_{ab} = g_{ab} + h_{ab}$ [13]. Other objects depending nonlinearly on $\overline{g}$ are later truncated at a given order in powers of h. This can be considered as a particular case of the general formalism used here, in which all but the first perturbations of the covariant metric are defined to be zero: ${}^{\{n\}}h_{ab} = 0$ for $n \geq 2$. All formulas in this article can be translated to background field method using such a simple restriction.

B. Perturbations of curvature tensors

The first step in any perturbative computation in general relativity is expressing the perturbation of one or several curvature tensors in terms of perturbations of the metric. This is a straightforward but very long computation for high orders, and hence requires the use of computer algebra tools like *xPert*. There are essentially two ways to perform such a computation: either recursively, computing the n-th order perturbation from the previous order, or using explicit formulas giving the result directly. *xPert* implements such explicit formulas for the most important tensors in general relativity, as shown in this and the next subsections. Some of these formulas have been already presented in [14].

The general term of the expansion of the inverse metric $\overline{g}^{ab}$ as a power series in ε can be obtained as follows. Let us define $H_{ab} \equiv \overline{g}_{ab} - g_{ab}$ and temporarily drop the indices. Then, the inverse of the metric is given by

$$\overline{g}^{-1} = (g + H)^{-1}, \tag{5}$$

which can be expanded around $H = 0$ as

$$\overline{g}^{-1} = g^{-1} \sum_{m=0}^{\infty} (-1)^m (g^{-1}H)^m. \tag{6}$$

Reintroducing indices and expanding H itself using (4) we obtain the general term

$$\Delta^n[g^{ab}] = \sum_{(k_i)} (-1)^m \frac{n!}{k_1! \ldots k_m!} \, {}^{\{k_m\}}h^{ae_m} \, {}^{\{k_{m-1}\}}h_{e_m}{}^{e_{m-1}} \ldots \, {}^{\{k_2\}}h_{e_3}{}^{e_2} \, {}^{\{k_1\}}h_{e_2}{}^{b}, \tag{7}$$

where the sum $\sum_{(k_i)}$ runs over the 2^{n-1} sorted partitions of n in $m \leq n$ positive integers (not including zero this time) such that $k_1 + \ldots + k_m = n$.

With the general term of the expansion of the metric and its inverse at hand, we can obtain the general term for the perturbation of the Christoffel symbols starting from its definition in terms of the metric:

$$\Delta^n[\Gamma^a{}_{bc}] = \sum_{(k_i)} (-1)^{m+1} \frac{n!}{k_1! \ldots k_m!} \, {}^{\{k_m\}}h^{ae_m} \, {}^{\{k_{m-1}\}}h_{e_m}{}^{e_{m-1}} \ldots \, {}^{\{k_2\}}h_{e_3}{}^{e_2} \, {}^{\{k_1\}}h_{e_2 bc}, \tag{8}$$

where the three-indices tensor perturbation is given by

$$ {}^{\{n\}}h_{abc} \equiv \frac{1}{2} \left({}^{\{n\}}h_{ab;c} + {}^{\{n\}}h_{ac;b} - {}^{\{n\}}h_{bc;a} \right), \tag{9}$$

which is symmetric in its last two indices. For $n = 1$ formula (8) must be read as $\Delta[\Gamma^a{}_{bc}] = {}^{\{1\}}h^a{}_{bc}$. Note that the perturbation of the Christoffel symbols is constructed from the metric perturbations ${}^{\{n\}}h_{ab}$ and their first covariant derivative at all orders.

The perturbation of the Riemann tensor of an arbitrary connection (not necessarily a metric connection) is a simple application of the Leibnitz rule (3):

$$\Delta^n[R_{abc}{}^d] = \nabla_b \left(\Delta^n[\Gamma^d{}_{ac}] \right) - \sum_{k=1}^{n-1} \binom{n}{k} \Delta^k[\Gamma^e{}_{bc}] \Delta^{n-k}[\Gamma^d{}_{ea}] - (a \leftrightarrow b). \tag{10}$$

For a metric-compatible connection we can use (8), getting

$$\Delta^n[R_{abc}{}^d] = \sum_{(k_i)} (-1)^m \frac{n!}{k_1! \ldots k_m!} \Bigg[{}^{\{k_m\}}h^{de_m} \ldots {}^{\{k_2\}}h_{e_3}{}^{e_2} \, {}^{\{k_1\}}h_{e_2 cb;a} \tag{11}$$

$$+ \sum_{s=2}^{m} {}^{\{k_m\}}h^{de_m} \ldots {}^{\{k_{s+1}\}}h_{e_{s+2}}{}^{e_{s+1}} \, {}^{\{k_s\}}h_{e_s e_{s+1} a} \, {}^{\{k_{s-1}\}}h^{e_s e_{s-1}} \ldots {}^{\{k_2\}}h_{e_3}{}^{e_2} \, {}^{\{k_1\}}h_{e_2 bc} \Bigg] - (a \leftrightarrow b).$$

Other relevant curvature tensors are given in terms of Riemann, and therefore their perturbations can be expressed in terms of the latter formula. For instance, perturbations of Ricci are given by

$$\Delta^n[R_{ab}] = \Delta^n[R_{acb}{}^c]. \tag{12}$$

The Ricci scalar $R \equiv g^{ab} R_{ab}$ can be perturbed using the generalized Leibnitz rule (3):

$$\Delta^n[R] = \sum_{k=0}^{n} \binom{n}{k} \Delta^k[g^{ab}] \Delta^{n-k}[R_{ab}], \tag{13}$$

and similarly for the Einstein tensor:

$$\Delta^n[G_{ab}] = \Delta^n[R_{ab}] - \frac{1}{2} \sum_{k=0}^{n} \sum_{j=0}^{k} \frac{n!}{k!\,j!\,(n-j-k)!} \, {}^{\{j\}}h_{ab} \, \Delta^k[g^{cd}] \, \Delta^{n-j-k}[R_{cd}]. \tag{14}$$

We have not tried to further simplify these two last formulas, as they are already quite efficient and give directly the n-th term of the respective perturbative expansions.

C. Perturbation of the metric determinant

In Hamiltonian field theory we frequently find the perturbation of the determinant of the metric. This is a basis-dependent concept, in the sense that what we are computing is the determinant of the components of the metric in a given basis and the result depends on the basis we have chosen. Under a change of basis the determinant changes with a squared Jacobian and it is, hence, a *density* of *weight* $+2$.

The determinant of the metric g_{ab} can be defined as

$$\det(g_{ab}) \equiv \frac{1}{N!} \widetilde{\eta}^{a_1 \ldots a_N} \widetilde{\eta}^{b_1 \ldots b_N} g_{a_1 b_1} \cdots g_{a_N b_N}, \tag{15}$$

using the upper antisymmetric density $\widetilde{\eta}^{a_1 \ldots a_N}$ (the overtilde denotes weight $+1$), whose components in the chosen basis are $+1$, -1 or 0. This object, as well as its lower counterpart $\underset{\sim}{\eta}_{a_1 \ldots a_N}$ of weight -1, stays invariant under the Δ perturbation. Therefore, the Leibnitz rule (3) implies

$$\Delta^n[\det(g_{ab})] = \frac{1}{N!} \widetilde{\eta}^{a_1 \ldots a_N} \widetilde{\eta}^{b_1 \ldots b_N} \sum_{\{k_i\}} \frac{n!}{k_1! \ldots k_N!} \, {}^{\{k_1\}}h_{a_1 b_1} \ldots \, {}^{\{k_N\}}h_{a_N b_N}. \tag{16}$$

Finally, this can be simplified using the well known relation

$$\widetilde{\eta}^{a_1 \ldots a_N} \widetilde{\eta}^{b_1 \ldots b_N} = \det(g_{ab}) \begin{vmatrix} g^{a_1 b_1} & \cdots & g^{a_1 b_N} \\ \vdots & & \vdots \\ g^{a_N b_1} & \cdots & g^{a_N b_N} \end{vmatrix}. \tag{17}$$

We conclude that the n-th order perturbation of the determinant of the metric is always the product of the determinant itself times a scalar formed by contraction of metric perturbations. It is interesting to note that such a scalar factor contains the product of at most N metric perturbations, and not n, as we might have anticipated by inspection of the formulas in the previous section. This is actually the only place in this article in which the dimension of the manifold being perturbed plays a role.

With this formula at hand it is straightforward to give the n-th order perturbation of the volume form $\epsilon_{a_1 \ldots a_N} \equiv |\det(g)|^{1/2} \underset{\sim}{\eta}_{a_1 \ldots a_N}$.

D. Perturbations of derivatives

There are several types of derivatives which may appear in a perturbative computation. We shall study here three of those types: partial, covariant and Lie derivatives.

Partial derivatives are associated to coordinate systems and hence do not change under perturbations. Therefore, by construction, they commute with the Δ operator: $\Delta^n[T_{,a}] = \Delta^n[T]_{,a}$ for any tensor field T of any rank.

General covariant derivatives can be perturbed. For instance, the Levi-Civita connection of a metric will change when its metric is perturbed. The question arises then about what is the perturbation of the covariant derivative

of a tensor. Transforming to partial derivatives and Christoffel symbols, perturbing and coming back to covariant derivatives, we get, for an arbitrary tensor density T of weight ω,

$$
\begin{aligned}
\Delta^n[T^{a_1...a_m}{}_{b_1...b_k;c}] = {} & \Delta^n[T^{a_1...a_m}{}_{b_1...b_k}]_{;c} \\
& + \sum_{i=1}^{m} \left\{ \Delta^n[\Gamma^{a_i}{}_{cd}T^{a_1...d...a_m}{}_{b_1...b_k}] - \Gamma^{a_i}{}_{cd}\Delta^n[T^{a_1...d...a_m}{}_{b_1...b_k}] \right\} \\
& - \sum_{j=1}^{k} \left\{ \Delta^n[\Gamma^{d}{}_{cb_j}T^{a_1...a_m}{}_{b_1...d...b_k}] - \Gamma^{d}{}_{cb_j}\Delta^n[T^{a_1...a_m}{}_{b_1...d...b_k}] \right\} \\
& - \omega \left\{ \Delta^n[\Gamma^{d}{}_{cd}T^{a_1...a_m}{}_{b_1...b_k}] - \Gamma^{d}{}_{cd}\Delta^n[T^{a_1...a_m}{}_{b_1...b_k}] \right\},
\end{aligned}
\tag{18}
$$

which can be rewritten applying the Leibnitz rule (3) as

$$
\begin{aligned}
\Delta^n[T^{a_1...a_m}{}_{b_1...b_k;c}] = {} & \Delta^n[T^{a_1...a_m}{}_{b_1...b_k}]_{;c} \\
& + \sum_{l=1}^{n} \binom{n}{l} \left\{ \sum_{i=1}^{m} \Delta^l[\Gamma^{a_i}{}_{cd}]\Delta^{n-l}[T^{a_1...d...a_m}{}_{b_1...b_k}] - \sum_{j=1}^{k} \Delta^l[\Gamma^{d}{}_{cb_j}]\Delta^{n-l}[T^{a_1...a_m}{}_{b_1...d...b_k}] \right. \\
& \left. \qquad -\omega\, \Delta^l[\Gamma^{d}{}_{cd}]\Delta^{n-l}[T^{a_1...a_m}{}_{b_1...b_k}] \right\}.
\end{aligned}
\tag{19}
$$

Finally, the perturbation formula for the Lie derivative along the vector field v of a tensor T of any rank can be computed by intermediate transformation to partial or covariant derivatives:

$$
\Delta^n[\mathcal{L}_v T] = \sum_{k=0}^{n} \binom{n}{k} \mathcal{L}_{\Delta^k[v]}\Delta^{n-k}[T].
\tag{20}
$$

This expression bears an obvious similarity with the Leibnitz rule, reflecting the fact that both v and T are being perturbed but not the Lie structure itself, which is directly given by the differential structure of the manifold and thus remains unperturbed.

III.   THE *xPert* PACKAGE

The general terms for the perturbation expansions in the previous section are combinatorial in nature and hence contain a number of terms which grows exponentially with the perturbative order n. Only through the use of a specialized tensor computer algebra system it can be possible to handle them for n beyond 1 or 2. Our work in second-order perturbation theory around spherical spacetimes [14, 15] motivated us to construct such a system, a very preliminary version of which was presented in an appendix of [14].

Here we introduce version 1.0 of the free-software package *xPert*, which implements all those expansion formulas and a number of other efficient tools to work with metric perturbation theory in the standard scenarios of general relativity, cosmology, quantum gravity or string theory. This section describes the main commands and features of the package, simultaneously constructing a very simple example session. (The *In[*]:=* and *Out[*]:=* prompts represent respectively input and output lines in *Mathematica*. Code lines without prompt indicate internal definitions in the package.) Next section will provide timings with harder examples.

xPert is a module of the framework *xAct* [20] for efficient tensor computer algebra in *Mathematica*. *xAct* is based on the use of fast algorithms of computational group theory, which allow very fast canonicalization of indexed expressions with arbitrary symmetries [21, 22]. Its core tensor packages are *xTensor* for abstract computations and *xCoba* for component computations, but *xAct* has also modules for manipulation of the Riemann tensor (*Invar* [23]) or tensor spherical harmonics (*Harmonics* [20]), and others are under development. These packages share a consistent notation, so that they can be run together, and are all free software.

Following Section II, which is valid for any background metric, *xPert* has been developed at the abstract computation level, to avoid dealing with choices of coordinate systems or frames. Therefore, it runs on *xTensor*, using its notations and conventions. A tensor field $T_a{}^b$ is denoted by `T[-a, b]` and the partial derivative operator ∂_a is represented as `PD[-a]`, such that $\partial_a v^b$ is `PD[-a][v[b]]` in the system. Let us start by loading *xTensor*:

> *In[1]* := `<<xAct`xTensor``
>
> (Version and copyright messages)

We first define our background structure: a four-dimensional manifold `M` whose tangent vector space will have (abstract) indices {a,b,c,d,e},

    In[2] := `DefManifold[ M, 4, {a,b,c,d,e} ]`

Then we define a metric tensor field `g` with negative determinant and associated Levi-Civita covariant derivative `CD`,

    In[3] := `DefMetric[ -1, g[-a,-b], CD, {";","∇"} ]`

        (Info messages on construction of associated tensors)

We have provided the symbols `{";","∇"}` to format the derivative in postfix or prefix output notation, respectively. `DefMetric` automatically defines all tensors normally associated to the metric or its connection, such as `ChristoffelCD[a,-b,-c]`, `RiemannCD[-a,-b,-c,-d]`, `EinsteinCD[-a,-b]`, `Detg[]`, and so on, with obvious meanings. We can define other tensors with the syntax

    In[4] := `DefTensor[ MaxwellF[a,b], M, Antisymmetric[{a,b}], PrintAs->"F" ]`

The arrow `->` is the *Mathematica* representation for an optional named argument.

    Now we load *xPert* (this would also load automatically *xTensor* if it wasn't already in memory):

    In[5] := `<<xAct'xPert'`

        -

        Package xAct'xPert' version 1.0.0, {2008, 6, 30}

        Copyright (C) 2005–2008 David Brizuela, Jose M. Martin-Garcia

         and Guillermo A. Mena Marugan, under GPL

        -

This adds several new commands and reserved words to the system, of which we shall here describe the four most important, namely `DefMetricPerturbation`, `Perturbation`, `ExpandPerturbation` and `GaugeChange`.

    A perturbative structure having metric `g` as background and the tensor `h` as its perturbation is defined using

    In[6] := `DefMetricPerturbation[ g, h, ε ]`

which also identifies ε as the perturbative parameter of the expansions. From now on, the *n*-th perturbation of the metric `g[-a,-b]` will be denoted as `h[LI[n],-a,-b]`, where `LI` is the *xTensor* head to denote so-called 'label indices', that is, indices with no vector space association. Labels can be considered as general non-geometric purpose indices.

    The Δ operator of section II is represented by the head `Perturbation`. It has two arguments: the background expression being perturbed and the perturbative order (with default value 1):

    In[7] := `Perturbation[ MaxwellF[a,b], 3 ]`

    Out[7] := $\Delta^3[F^{ab}]$

Note that the tensor is represented with its symbol F and that the perturbation order is an exponent of Δ, as in section II. Following normal *Mathematica*, the output is formatted for the sake of clarity, but the internal notation is still the same. `Perturbation` acts mainly as a wrapper for tensor expressions, but has been instructed to evaluate them under certain conditions. First, it automatically combines perturbative orders of composed heads (symbols with an underscore are named patterns in *Mathematica*):

        `Perturbation[ expr_, 0 ] := expr`

        `Perturbation[ Perturbation[ expr_, n_ ], m_ ] := Perturbation[ expr, n+m ]`

Being a derivative, `Perturbation` is linear and gives zero on the `delta` identity tensor and constants:

        `Perturbation[ x_ + y_, n_ ] := Perturbation[ x, n ] + Perturbation[ y, n ]`

```
Perturbation[ delta[a_, b_], n_ ] := 0

Perturbation[ expr_?ConstantQ, n_ ] := 0
```

The Leibnitz rule is also automatic, and has been implemented following equation (3) for any number of factors and any perturbative order, using fast algorithms to compute partitions implemented in *xPert*. Perturbation commutes with partial derivatives of general expressions and with any covariant derivative of a scalar expression:

```
Perturbation[ PD[-a_][ expr_ ], n_ ] := PD[-a][ Perturbation[ expr, n ] ]

Perturbation[ CD_?CovDQ[-a_][ expr_?ScalarQ ], n_ ] :=

    CD[-a][ Perturbation[ expr, n ] ]
```

The index of the derivatives is required to be always covariant, to avoid a metric mismatch, and that is implemented through a pattern index -a_ . Finally, Perturbation does not change the density weight of the perturbed expression:

```
WeightOf[ Perturbation[ expr_, n_ ] ] := WeightOf[ expr ]
```

The DefMetricPerturbation in *In[6]* defines special rules for the metric g and its perturbations h with covariant indices:

```
Perturbation[ g[-a_,-b_], n_ ] := h[LI[n],-a,-b]

Perturbation[ h[LI[n_],-a_,-b_], m_ ] := h[LI[n+m],-a,-b]
```

With the setup and internal definitions so far we can now perform computations like this second order perturbation

$In[8] :=$ Perturbation[g[-a,-b] RicciCD[c,d] + RiemannCD[-a,-b,c,d], 2]

$Out[8] := \ 2\,h^1{}_{ab}\,\Delta[R_{cd}] + g_{ab}\,\Delta^2[R_{cd}] + \Delta^2[R_{ab}{}^{cd}] + h^2{}_{ab}\,R_{cd}$

Actually, we could now proceed to perform any computation in metric perturbation theory by decomposing the curvature tensors in partial derivatives of the metric and using the code already given recursively. Only the definition $\Delta[g^{ab}] = -\,{}^{\{1\}}h^{ab}$ would be missing. However, that would be highly inefficient already for moderate perturbative order n. Instead, we shall use the expansion formulas of section II, which allow the nonrecursive construction of perturbations at any order n.

Formulas (18, 20) for derivative expansions and formulas (7, 8, 11, 12, 13, 14, 16) for the relevant curvature tensors have all been encoded in a single command called ExpandPerturbation, the most powerful part of *xPert*. ExpandPerturbation takes any expression and replaces the arbitrary-order perturbations of known background objects by their expansions in terms of metric perturbations, but only if those objects have their indices in the appropriate positions. For example the perturbation of the Einstein tensor has only been stored for covariant indices. In all other cases there is an internal call to the *xTensor* function SeparateMetric, which introduces metric factors to bring all indices to their *natural* positions, which are those given at definition time. To show how this works, we perform an explicit metric separation by hand (symbol % represents the previous output):

$In[9] :=$ Perturbation[EinsteinCD[a,b]]

$Out[9] := \ \Delta[G^{ab}]$

$In[10] :=$ SeparateMetric[ ][%]

$Out[10] := \ G_{cd}\,g^{bd}\,\Delta[g^{ac}] + g^{ac}\left(g^{bd}\,\Delta[G_{cd}] + G_{cd}\,\Delta[g^{bd}] \right)$

Now ExpandPerturbation can expand the perturbation of the Einstein tensor with covariant indices, and the perturbation of the inverse metric.

$In[11] :=$ ContractMetric[ExpandPerturbation[%]]

$Out[11] := \ -G_c{}^b h^{1ac} - G^a{}_c h^{1bc} + \frac{1}{2}g^{ab}h^{1cd}R_{cd} - \frac{1}{2}h^{1ab}R - \frac{1}{2}h^{1c}{}_c{}^{;b;a}$

$-\frac{1}{2}h^{1cb}{}_{;c}{}^{;a} + \frac{1}{2}h^{1b}{}_c{}^{;c;a} + \frac{1}{2}h^{1cb;a}{}_{;c} + \frac{1}{2}h^{1ca;b}{}_{;c} - \frac{1}{2}h^{1ba;c}{}_{;c}$

```
AbsoluteTiming[
 Perturbation[EinsteinCD[-a, -b], 2] // ExpandPerturbation // ContractMetric //
  ToCanonical]
```

$$\Big\{ 0.783872\ \text{Second},\ h^1{}_{ab}\,h^{1\,cd}\,R_{cd} + \tfrac{1}{2}\,g_{ab}\,h^{2\,cd}\,R_{cd} - g_{ab}\,h^1{}_c{}^e\,h^{1\,cd}\,R_{de} - \tfrac{1}{2}\,h^2{}_{ab}\,R + h^{1\,cd}\,h^1{}_{cd;b;a} +$$

$$\tfrac{1}{2}\,h^{1\,cd}{}_{;a}\,h^1{}_{cd;b} - \frac{h^2{}_c{}^c{}_{;a;b}}{2} + \tfrac{1}{2}\,h^1{}_b{}^c{}_{;a}\,h^1{}^d{}_{d;c} + \tfrac{1}{2}\,h^1{}_a{}^c{}_{;b}\,h^1{}^d{}_{d;c} + \frac{h^2{}_b{}^c{}_{;a;c}}{2} + \frac{h^2{}_a{}^c{}_{;b;c}}{2} - \frac{h^2{}_{ab}{}^{;c}{}_{;c}}{2} -$$

$$\tfrac{1}{2}\,h^1{}^d{}_{d;c}\,h^1{}_{ab}{}^{;c} - h^1{}_b{}^c{}_{;a}\,h^1{}_c{}^d{}_{;d} - h^1{}_a{}^c{}_{;b}\,h^1{}_c{}^d{}_{;d} + h^1{}_{ab}{}^{;c}\,h^1{}_c{}^d{}_{;d} - h^{1\,cd}\,h^1{}_{bc;a;d} - h^{1\,cd}\,h^1{}_{ac;b;d} +$$

$$h^{1\,cd}\,h^1{}_{ab;c;d} - h^1{}_{ab}\,h^{1\,cd}{}_{;c\,;d} - g_{ab}\,h^{1\,cd}\,h^1{}^e{}_{e;c;d} - \tfrac{1}{2}\,g_{ab}\,h^{2\,cd}{}_{;c\,;d} + h^1{}_{ab}\,h^1{}_c{}^c{}^{;d}{}_{;d} + \tfrac{1}{2}\,g_{ab}\,h^{2\,c}{}_c{}^{;d}{}_{;d} +$$

$$g_{ab}\,h^{1\,cd}\,h^1{}_c{}^e{}_{;e;d} - h^1{}_{bd;c}\,h^1{}_a{}^{c;d} + h^1{}_{bc;d}\,h^1{}_a{}^{c;d} + \tfrac{1}{4}\,g_{ab}\,h^1{}^e{}_{e;d}\,h^1{}_c{}^{c;d} + g_{ab}\,h^{1\,cd}{}_{;c}\,h^1{}_d{}^e{}_{;e} -$$

$$g_{ab}\,h^1{}_c{}^{;d}\,h^1{}_d{}^e{}_{;e} + g_{ab}\,h^{1\,cd}\,h^1{}_c{}^e{}_{;d;e} - g_{ab}\,h^{1\,cd}\,h^1{}_{cd}{}^{;e}{}_{;e} + \tfrac{1}{2}\,g_{ab}\,h^1{}_{ce;d}\,h^{1\,cd;e} - \tfrac{3}{4}\,g_{ab}\,h^1{}_{cd;e}\,h^{1\,cd;e} \Big\}$$

FIG. 1: The second-order perturbation of the Einstein tensor is constructed and canonicalized in less than one second. The blue labels of the `h` tensors denote the perturbative order.

$$+\tfrac{1}{4}g^{ab}h^{1d}{}_c{}^{;c}{}_{;c} + \tfrac{1}{4}g^{ab}h^{1dc}{}_{;d;c} - \tfrac{1}{4}g^{ab}h^{1c}{}_c{}^{;d}{}_{;c} - \tfrac{1}{2}g^{ab}h^{1dc}{}_{;c;d} + \tfrac{1}{4}g^{ab}h^{1c}{}_c{}^{;d}{}_{;d}$$

The *xTensor* command `ContractMetric` has been used to absorb all possible metric factors. Finally, `ToCanonical` moves indices around bringing equal terms together,

$In[12] :=$ `ToCanonical[ % ]`

$Out[12] := -G^{bc}h^{1a}{}_c - G^{ac}h^{1b}{}_c + \tfrac{1}{2}g^{ab}h^{1cd}R_{cd} - \tfrac{1}{2}h^{1ab}R - \tfrac{1}{2}h^{1c}{}_c{}^{;a;b}$

$+\tfrac{1}{2}h^{1bc;a}{}_{;c} + \tfrac{1}{2}h^{1ac;b}{}_{;c} - \tfrac{1}{2}h^{1ab;c}{}_{;c} - \tfrac{1}{2}g^{ab}h^{1cd}{}_{;c;d} + \tfrac{1}{2}g^{ab}h^{1c}{}_c{}^{;d}{}_{;d}$

Figure 1 shows the output of the second perturbation of the covariant Einstein tensor, computed with the same combination of commands.

Another useful command implemented in *xPert* is `GaugeChange`. A ε-dependent family of diffeomorphisms on a manifold can be parametrized by an infinite collection of vector fields $^{\{n\}}\xi^a$, one per perturbative order, and the action of these on any background tensor at any order has been given by Bruni and collaborators [17]. The general case has been implemented here and can be used as follows. First define the family of generator vector fields on the manifold `M` transforming from the current gauge to a new gauge:

$In[13] :=$ `DefTensor[ ξ[LI[n],a], M ]`

The third-order perturbation of F^{ab} can be changed to the new gauge using

$In[14] :=$ `GaugeChange[ Perturbation[ MaxwellF[a,b], 3 ], ξ ]`

$Out[14] := \Delta^3[F^{ab}] + 3\,\mathcal{L}_{\xi^1}\Delta^2[F^{ab}] + 3\,\mathcal{L}_{\xi^1}\mathcal{L}_{\xi^1}\Delta[F^{ab}] + \mathcal{L}_{\xi^1}\mathcal{L}_{\xi^1}\mathcal{L}_{\xi^1}F^{ab}$

$+3\,\mathcal{L}_{\xi^1}\mathcal{L}_{\xi^2}F^{ab} + 3\,\mathcal{L}_{\xi^2}\Delta[F^{ab}] + \mathcal{L}_{\xi^3}F^{ab}$

We finish this section by coming back to the problem of perturbation theory using the background field method, in which all but the first metric perturbations vanish, as stated in section II. This can be easily implemented setting

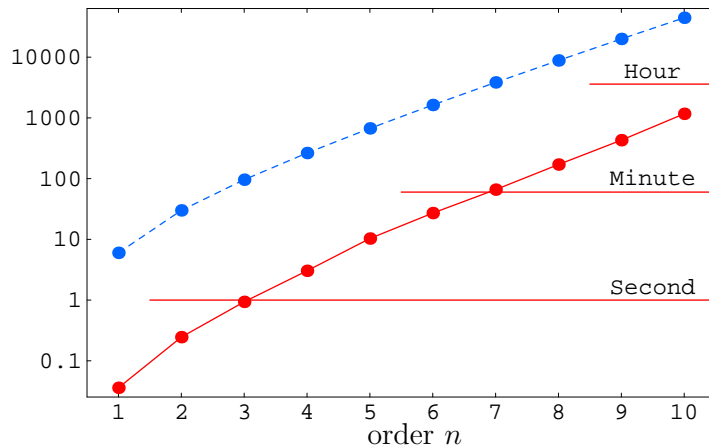$In[15] :=$ `h[LI[n_],a_,b_]:=0 /; n > 1`

FIG. 2: Canonicalization timings (in seconds) for the perturbation of the Riemann tensor at perturbative orders $n = 1...10$ (lower, red line). Also shown number of terms in the expression (upper, blue dashed line). Both lines are clear exponentials in n, with the timings growing slightly faster because terms with larger n are harder to canonicalize due to their larger average number of indices.

IV. EXAMPLES

This section presents several examples of computations, focusing on the dependence of their timings on the perturbative order and the number of objects being perturbed. The intrinsic combinatorial nature of the problem will always imply exponential dependence, but we will see that the timings in *xPert* are short enough to handle all useful cases. These examples have been performed using a Linux box with a 3 GHz Pentium IV processor and 2 Gb of RAM.

In perturbation theory the overall level of complexity is mainly determined by the perturbative order n. It affects the computation in two different ways: on the one hand the expressions to manipulate are sums with a number of terms which grows exponentially with n; on the other hand each term is a product of objects and the number of factors also grows (typically linearly) with n. Canonicalizing a sum of terms is obviously a linear process because each term can be dealt with independently (this is an ideal scenario for parallelization), but canonicalization of a product of objects is naturally factorial in the number of indices and this could prevent any practical computation. The canonicalization algorithms in *xTensor* are fast enough to render the problem effectively polynomial in the number of indices, allowing us to deal with expressions of a few dozen indices in hundredths of a second. Figure 2 shows the number of terms and the timing of canonicalization of the perturbation of the Riemann tensor at different perturbative orders. (The timing of construction of the expression is negligible in comparison.) The 10-th order perturbation is canonicalized in slightly less than 20 minutes and contains 44544 terms. We see clearly the exponential growth of both curves, but with manageable timings. Third-order perturbation expressions can be manipulated in 1 second.

Other possible sources of complexity in perturbative computations are the expansion of perturbations of a product of tensors and the expansion of perturbations of a function of a number of scalar arguments. Our implementations of the nth-order Leibnitz rule and Faà di Bruno formula, respectively, are fast enough to neglect their timings in comparison with those of canonicalization. Figures 3 and 4 display example timings for those problems. The Leibnitz rule is simpler than the Faá di Bruno formula and produces faster results, also taking less memory.

Overall, we see that we are limited in size by RAM memory, which allows us to work with up to 10^5 terms roughly with a few Gbytes, corresponding to $n = 10$ approximately. Time limitations come mainly from the canonicalization process (other expansions are faster): Within seconds we can manipulate all equations up to orders $n = 4$ or $n = 5$. The $n = 10$ equations require canonicalization times of the order of 1 hour. This gives an idea of the power and efficiency of *xPert*, and what can be achieved with it.

V. CONCLUSIONS

Linear perturbation theory is an important tool in general relativity, nowadays playing a complementary role to numerical relativity [24]. Its precision and applicability can be extended by adding higher-order perturbations in the computation, transforming the intrinsic non-linearity of the theory into sources for the linearized equations of motion. In most cases these sources are huge, and hence practical calculations require the use of specialized computer algebra
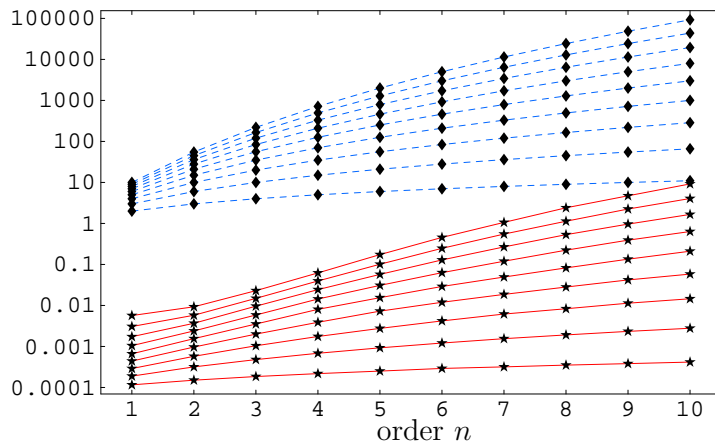
FIG. 3: Timings of expansion (in seconds; red lines, stars) and number of terms (blue dashed lines, diamonds) of the perturbation of the product of m factors, for different perturbative orders $n = 1...10$ [formula (3)]. Different lines correspond to increasing values of m, from $m = 2$ to $m = 10$ starting from below. All practical cases stay below one second.
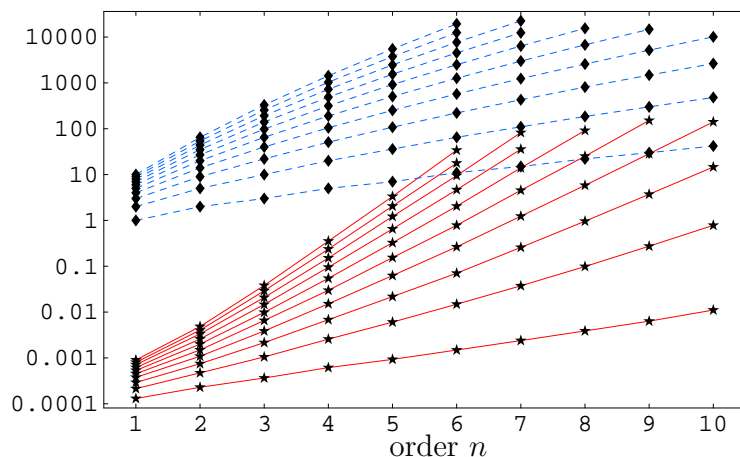

FIG. 4: Timings of expansion (in seconds; red lines, stars) and number of terms (blue dashed lines, diamonds) of the perturbation of a scalar function of m scalar arguments, for different perturbative orders $n = 1...10$. Different lines correspond to increasing values of m, from $m = 1$ to $m = 10$ starting from below. We include only those cases which can be handled with 2Gb of RAM memory, corresponding to a few tens thousand terms. For example, for $n = 10$ we can only handle up to $m = 4$.

to construct and manipulate them.

In this article we have introduced version 1.0 of the package *xPert* for high-order metric perturbation theory. This is a very hard problem, of intrinsic combinatorial nature, and hence exponential in the perturbative order n. We have developed and coded up highly efficient algorithms and tools, based on the combination of explicit non-recursive formulas to build the equations at any order n, and powerful algorithms of computational group theory to manipulate them. *xPert* can work both with a perturbative scheme in which the metric is expanded as an infinite series of perturbations (the usual approach in general relativity) and with the alternative scheme in which the metric is written as a background plus a single 'deviation' term (as used in quantum field theory). *xPert* allows working with orders $n = 4$ and $n = 5$ in seconds, and we have shown that it is possible to reach $n = 10$ with average timings below 1 hour with moderate computer resources. The main limitation (both in time and memory) is the huge number of terms produced, but this can be easily dealt with using parallelization, because different terms in the expression can be treated independently by different processors.

xPert is a module of the general framework *xAct* [20] for tensor computer algebra in *Mathematica*, sharing a consistent notation with the whole system. This means that once the perturbative equations have been constructed, there are several hundred additional commands available for tensor manipulations, including particularization to a

special background metric, and several thousand more for generic manipulation in *Mathematica*. In particular, if one wants to convert the produced equations into C or FORTRAN code, it can be easily achieved with the `CForm` and `FortranForm` commands in *Mathematica*.

xPert has been already used in several research projects: Our construction of a general formalism for high-order metric perturbations around spherical backgrounds [14] and its gauge-invariant form [15]. In a cosmological setting, high-order perturbations of non-linear radiation transfer have been studied using *xPert*, both in kinetic theory [18] and in a fluid approximation [25], as well as first-order perturbations of scalar field inflation for anisotropic spacetimes [26].

The package *xPert* is free software distributed under the GNU general public license, and can be download from the webpage `http://metric.iem.csic.es/Martin-Garcia/xAct/xPert/` .

Acknowledgments

[1] Cunningham, C. T., Price, R. H., Moncrief, V.: Astroph. J. **236**, 674 (1980)
[2] Chandrashekar, S., Ferrari, V.: Proc. R. Soc. London **A433**, 423 (1991)
[3] Kojima, Y.: Phys. Rev. D **46**, 4289 (1992)
[4] Teukolsky, S. A.: Phys. Rev. Lett. **29**, 1114 (1972)
[5] Campanelli, M., Lousto, C. O.: Phys. Rev. D **59**, 124022 (1999)
[6] Price, R. H., Pullin, J.: Phys. Rev. Lett. **72**, 3297 (1994)
[7] Gleiser, R. J., Nicasio, C. O., Price, R. H., Pullin, J.: Class. Quantum Grav. **13**, L117 (1996); Phys. Rev. Lett. **77**, 4483 (1996); Phys. Rev. D **57**, 3401 (1998); Phys. Rept. **325**, 41 (2000)
[8] Nakano, H., Ioka, K.: Phys. Rev. D **76**, 084007 (2007)
[9] Ioka, K., Nakano, N.: Phys. Rev. D **76**, 061503 (2007)
[10] Matarrese, S., Pantano, O., Saez, D.: Phys. Rev. Lett.**72**, 320 (1994); Mon. Not. Roy. Astron. Soc. **271**, 513 (1994)
[11] Salopek, D. S., Stewart, J. M., Croudace, K. M.: Mon. Not. R. Astron. Soc. **271**, 1005 (1994)
[12] Bruni, M., Mena, F., Tavakol, R.: Class. Quantum Grav. **19**, L23 (2002)
[13] Goroff, M. H., Sagnotti, A.: Nucl. Phys. B **266**, 709 (1986)
[14] Brizuela, D., Martín-García, J. M., Mena Marugán, G. A.: Phys. Rev. D **74**, 044039 (2006)
[15] Brizuela, D., Martín-García, J. M., Mena Marugán, G. A.: Phys. Rev. D **76**, 024004 (2007)
[16] Gerlach, U. H., Sengupta, U. K.: Phys. Rev. D **19**, 2268 (1979)
[17] Bruni, M., Matarrese, S., Mollerach, S., Sonego, S.: Class. Quantum Grav. **14**, 2585 (1997)
[18] Pitrou, C.: Class. Quantum Grav. **24**, 6127 (2007)
[19] Abramowitz, M., Stegun, I. A.: *Handbook of mathematical functions with formulas, graphs and mathematical tables.* Dover, New York (1964)
[20] Martín-García, J. M.: *xAct:* Efficient Tensor Computer Algebra, `http://metric.iem.csic.es/Martin-Garcia/xAct/`
[21] Manssur, L. R. U., Portugal, R., Svaiter, B. F.: Int. J. Modern Phys. C **13**, 859 (2002)
[22] Martín-García, J. M.: *xPerm:* Fast index canonicalization for tensor computer algebra, Comp. Phys. Commun. (2008), in press
[23] Martín-García, J. M., Portugal, R., Manssur, L. R. U.: Comp. Phys. Commun. **177**, 640 (2007)
[24] Seidel, E.: Class. Quantum Grav. **21**, S339 (2004)
[25] Pitrou, C., Uzan, J. P., Bernardeau, F.: arXiv:0807.0341v1 [astro-ph]
[26] Pitrou, C., Pereira, T. S., Uzan, J. P.: JCAP **04**, 004 (2008)