New system for indicial computation and its applications in gravitational physics

Sergei A. Klioner

Lohrmann Observatory, Dresden Technical University,
Mommsenstraße, 13, 01062 Dresden, Germany
e-mail: klioner@rcs.urz.tu-dresden.de
WWW: http://rcswww.urz.tu-dresden.de/~klioner/

Abstract

EinS is a package for *Mathematica* intended for calculations with indexed objects (e.g., tensors). The package automatically handles dummy indices and Einstein's summation notation, enables one to define new indexed objects and to assign symmetries to that objects. EinS has an efficient simplification algorithm based on pattern matching technique which takes full account for the symmetries of the objects and the possibility to rename dummy indices. Other important features of EinS are the ability to perform automatically "3+1" split of implicit summations, printing expressions in a natural 2-dimensional form and exporting them into plain TeX or LaTeX with user-controllable alignment commands.

As a typical application of EinS, the problem of constructing of a local reference system for a massive extended body in the framework of the Parametrized Post-Newtonian formalism is briefly described. The use of EinS to solve the problem is also sketched.

1 Introduction

Various calculations in the field of general relativity represent one of the traditional application areas of computer algebra. Complexity of calculations typical for virtually any serious problem in this field makes manual computations extremely tedious and unreliable. Starting from the 1960s many computer algebra systems and packages were developed for the use in this particular research field [17,7,5]. Here, it is necessary to distinguish between the systems intended for indicial tensor computation and those for computation of components of tensors. These two classes of packages are quite different. The latter usually allows one to compute a set of quantities for given components

Computer Physics Communications, Thematic Issue on "Computer Algebra in Physics Research", edited by R.G. McLenaghan and E.S. Cheb-Terrab, Elsevier, 1998, to appear

of the metric tensor: components of Christoffel symbols, curvature tensor and so on. For example, SHEEP and ORTHOCARTAN as stand-alone systems, EXCALC and REDTEN for REDUCE, CTENSR for Macsyma, and recently developed GRTensor/II for MAPLE and TTC and CARTAN for *Mathematica* belong to that class of the systems (see, e.g., [17,7,5] for a more comprehensive list). On the contrary, we are interested mostly in indicial calculation. Here, indices (of tensors or other objects) remain symbols and software operates with symbolic relations between various indexed objects. Those relations typically involve Einstein (implicit) summations. The principal aim of a package for indicial tensor computation is to handle implicit summation rules and to be able to simplify expressions taking into account the user-defined symmetries of various objects and the possibility to rename dummy indices. Among packages for indicial computations one can mention ITENSR for Macsyma and STENSOR for SHEEP (both appeared in the late 1970s) and the two newer packages MathTensor and RICCI for *Mathematica* (again, see, e.g., [17,7,5] for a more comprehensive list). However, the algorithms for indicial tensor computation are tricky, and every new attempt to implement them is of interest.

On the other hand, each particular field of application has its own specific language (a set of notations, conventions, etc.) and put forward specific requirements to the software. The author of this paper works in the field of astronomical applications of metric gravity theories. Typical calculations in this field represent operations with power series (for example, in powers of the light velocity as within the post-Newtonian approximation scheme) involving indexed objects (functions, tensors, etc.) with various symmetry properties as well as partial derivatives of the objects with respect to coordinates or other parameters. Implicit (Einstein) summation notation is widely used here. Another important property is that in many cases "time" and "spatial" values of indices are treated separately ("3+1" split of space-time), and covariant and contravariant "spatial" indices are often not distinguished at all. None of the existing packages for indicial tensor calculation allows us to handle automatically all the mentioned properties of our research field.

This forced us to write our own package for operations with indexed objects. The package is called EinS which stands for "Einstein Summation handler". EinS is a package for *Mathematica* which is one of the best modern computer algebra systems. This allows one to use the full power of *Mathematica* to treat the problem under study. For example, such facilities of *Mathematica* as arbitrary precision arithmetic, simplification of rational expressions, symbolic differentiation and integration, highly controllable substitution mechanism, flexible formatted output system, etc. are very important for many kinds of applications. EinS is a relatively small package. It consists of about 3000 lines of *Mathematica* top level code including on-line help messages. It is a flexible package which is relatively easy to alter to treat any reasonable problem involving indexed objects. General design and functionality of the package

resulted from scientific problems in the field of astronomical applications of metric gravity theories. EinS automatically handles implicit summations and dummy indices, allows one to assign symmetry properties to new objects and has an efficient built-in simplification algorithm based on pattern matching technique. Further features include printing expressions in a 2-dimensional form, exporting into plain TEX or LATEX with user-controllable alignment commands, converting implicit summations into explicit ones, debugging capabilities and online help messages. EinS supports also two kinds of symbolic indices: "time-space" ones (running from 0 to 3 in the standard notations of general relativity), and purely "spatial" ones (running from 1 to 3). EinS is not specifically designed for tensors. The objects within EinS may be tensors or not. It was our intention to support no features to distinguish between covariant and contravariant indices. However, flexible printing capabilities of the package allows one in principle to mimic rigorous tensor notations if desired (see, Section 2). However, if tensor properties of some objects are essential for a particular problem, EinS is hardly the right software to treat it.

EinS (as well as *Mathematica* itself) is not intended to operate with very large expressions. The most cumbersome computation which has been performed with the use of EinS was computing the Landau-Lifshits pseudotensor for a generic metric containing terms of order of $\mathcal{O}(c^{-1})$ in the time-space components g_{0i} of the metric tensor. This problem is interesting for defining angular velocity of an extended massive celestial body in the framework of general relativity [11]. The calculations involved about 2000 terms in the result and up to 5000 terms in intermediate expressions. Each term was a product (contraction) of up to 10 indexed objects with one or two indices. Some of the objects with two indices were defined as symmetric. This calculation was performed several years ago with *Mathematica* 2.1 under MS-DOS on IBM PC 486, 33 MHz with 16 Mbytes RAM. It took 7 hours of CPU time, and was about the most complicated calculation which could be fitted into that computer (the most serious limitation being RAM). Modern computers allow EinS to operate with 10000 or more terms depending on their structure. However, if one has to operate with much longer expressions (say, with $\sim$ 100000 terms) STENSOR or FORM [5] are probably a better choice.

2 General structure of EinS

EinS works under *Mathematica* starting from version 2.1. This allows one to employ the power of one of the most comprehensive computer algebra systems, but, as a consequence, it also requires some knowledge of the *Mathematica*'s user interface and its top level language. Although the semantics of the examples below are obvious, we suppose that the reader has certain experience with *Mathematica*. We describe below EinS by dividing its commands into several

functional groups and giving examples for each group. In the examples below the lines starting with symbol "`>`" are input from users and those starting with "`<`" are responses from EinS. The commands of EinS can be divided into the following groups.

- Commands which represent several built-in objects. E.g., `LeviCivita[i,j,k]` and `Delta[i,j]` denote the fully antisymmetric Levi-Civita and Kronecker symbols, respectively.
- Commands to define new indexed objects and their symmetries. E.g., the command

```
> DefObject[ A, 4, {3,4}, {1,2}, {3,4} ]

< DefObject:: Object defined as follows:
<              input name: A, valence 4
<              symmetric in indices: {3,4}
<              antisymmetric in indices: {1,2}
<              indices number(s) {1,2} are printed
<               as superscripts
<              indices number(s) {3,4} are printed
<               as subscripts
```

  defines object `A` with 4 indices. `A` is antisymmetric with respect to its first two indices and symmetric with respect to the last two ones. The first two indices will be printed as superscripts while the last two ones as subscripts:

```
> A[i,j,k,l]

<  ij
< A
<    kl
```

  When defining an object the user can specify the number of indexes of the object, its print name to be used instead of the input name (it is useful, e.g., for defining several different objects with different number of indices, but with the same name on output), symmetry properties of the objects and, finally if each particular index of the object is to be printed as a subscript or a superscript. We see that although EinS is not specifically intended for tensors and does not distinguish automatically between covariant and contravariant indices, its flexible printing capabilities allow one to mimic rigorous tensor notations if it is necessary.
- Commands to declare that certain objects do not depend on some parameters (e.g., coordinates or time). For example, the command

```
> DeclareConstant[gamma,0]
```

```
< DeclareConstant:: gamma, valence 0 is declared constant
```

declares an object `gamma` without indices to be constant (any derivatives of the object are zero). Under *Mathematica* 3.0 notebook interface EinS makes use of its new printing capabilities (e.g., by printing "space-time" dummy indices as greek letters, the Kronecker's `delta` as δ, etc.). The user can also specify any special characters as names of objects (γ for `gamma`, etc.).

- Commands to declare particular symbols to represent coordinates of a reference system. The command

```
> DefRS[x,t]
```

```
<                                    -1  0  i
< DefRS:: reference system (t = c   x ,x ) defined.
```

defines `x[i]` and `t` to represent the spatial coordinates and the coordinate time $t = x^0/c$ of a reference system. This information is used primarily to output various expression in traditional notations. E.g., $\dfrac{\partial \left(\gamma\, A^{ij}{}_{kl}\right)}{\partial x^s}$ can be defined as

```
> PD[gamma A[i,j,k,l],x[s]]
```

```
<         ij
< gamma A
<           kl,s
```

It is automatically simplified (taking into account that `gamma` is declared to be constant above) and printed in traditional notations.

- Commands to define relations between objects including implicit summation rules. E.g.,

```
> S = DefES[ A[i,j,k,l] Delta[i,j], {i,j} ];
```

defines `S` to represent $A^{ij}{}_{kl}\,\delta_{ij}$. EinS allows one to declare each dummy index appearing in an implicit summation to be "space-time" (running from 0 to N, where in the default case of 4-dimensional space-time $N = 3$) or purely "spatial" (running from 1 to N). The latter case is default. EinS automatically distinguishes between these two kinds of indices. This allows one to use the package effectively when working with the "3+1" split of space-time.

- Commands to simplify expressions. The principal command here is `ComputeES`. For example,

```
> ComputeES[S]
```

`< 0`

The simplification algorithm of EinS consists of several parts:

(1) Each term is transformed into a quasi-"canonical" form. At this step dummy indices in each term are renamed. The new names are taken from a standard sequence of names and the indices are renamed in the order they appear in the expression, the expression being sorted lexicographically by *Mathematica*. After this operation, which is very fast, equal terms may still have different forms and it is by no means true canonical form. Nevertheless, in many cases this quasi-"canonical" form decreases the number of different terms substantially.

(2) The expression is then further simplified by subsequent *pattern matching* among the rest of the terms with full account for the symmetries of the objects and the possibility to rename dummy indices. It is clearly the most time-consuming part of the simplification algorithm. Its practical implementation is tricky. Not going into too subtle technical details, the pattern matching algorithm can be described as follows.

At the first step EinS takes a term from the expression to be simplified and picks up all other terms which have (a) the same number of dummy indices (this number is stored internally by EinS together with each term to speed up certain operations) and (b) the same set of factors. For example, term $A^{ij}\,B^{jk}\,A^{ki}$ has two factors A^{ij} and one factor B^{ij}. All terms having also two factors A^{ij} and one factor B^{ij} irrespective of the indices of those factors are then selected. Thus, we split the whole expression into several classes of terms. The terms can be further simplified (matched) only within one class. Further pattern matching is then performed among the terms from one class, the classes being considered one by one.

For the moment two implementations of the pattern matching algorithm are used in various versions of EinS: an implementation which uses *Mathematica*'s built-in function `MatchQ` and an implementation which operates essentially in the same way as the first one, but our own function with almost the same functionality as `MatchQ` is used instead. `MatchQ[x,y]` decides if its first argument `x` matches pattern `y`. Pattern `y` should be constructed according to the corresponding rules of *Mathematica*. It can involve functions defined in *Mathematica* with attribute `Orderless`. Arguments of such functions are automatically sorted by *Mathematica* in lexicographical order. In this case `MatchQ` takes into account the fact that the arguments of such functions can be reordered to match certain pattern, and uses usual backtracking algorithm to check possible permutation of the arguments until the relevant permutation is found or matching fails. The corresponding subroutine of EinS generates the pattern for a term in the class under consideration in such a way that all defined symmetries of the objects appearing in the term are taken into account. That pattern

enables us to find all the terms in the current class which can be matched with the term under consideration. All these terms are then simplified into one single term (possibly with a complicated factor which does not contain dummy indices), and the algorithm is applied to the rest of the current class.

A further detail of the algorithm which greatly improves its performance is that the indices of the objects having no symmetries (that is, when there exists only one possibility to match the indices) are matched first. An example can be found in Section 4.

Our own implementation of `MatchQ` was caused by the idea to choose certain controllable order of checking the permutations of the indices. However, it turns out that it is possible to have the desired control by choosing different forms of the pattern for `MatchQ`. The latest version of EinS does not use our own implementation of `MatchQ`.

One clear reason why the pattern matching algorithm is superior to casting each term into a true canonical form (say, into its lexicographically minimal form) is that we do have to check all possible permutations of the indices only in the worst case. Normally only a part (less then 50%) of the permutations are to be checked. Anyway the algorithm may be slow when symmetric objects with large number of indices are involved (the number of permutations of n indices is anyway $n!$).

(3) The built-in objects are then simplified by using their pre-defined properties. E.g., $\delta^{ij}\,\delta^{jk} = \delta^{ik}$. Each simplification algorithm of this kind is programmed as a separate set of *Mathematica*'s rules to be applied by request.

(4) Then we check if each term can be further simplified in virtue of some additional circumstances. For the moment only one additional simplification modul of this kind is implemented: $A^{ij}\,B^{ij}$ will be simplified to zero if A is symmetric and B is antisymmetric as in the example with `S` above. This simplification rule is implemented quite generally, so that it works whenever possible. E.g., for an antisymmetric object $K^{ij} = -K^{ji}$ we have $K^{ij}\,K^{jk}\,K^{ki} = 0$:

```
> DefObject[K,2,{},{1,2}]

< DefObject:: Object defined as follows:
<             input name: K, valence 2
<             antisymmetric in indices: {1, 2}
<             indices number(s) {1, 2} are printed
<              as superscripts

> L=DefES[K[i,j] K[j,k] K[k,i],{i,j,k}];

> ComputeES[L]
```

```
< 0
```

The steps of the simplification algorithm can be executed manually one by one or automatically in a reasonable sequence with the command `ComputeES`.

- Commands to print expressions in a natural 2-dimensional form with "pretty" automatically generated (and controllable by the user) names for dummy indices

```
> S = ComputeES[ DefES[ A[i,j,k,s] Delta[j,k], {j,k} ] ];
> PrintES[S]

<  ia
< A
<    sa
```

- Commands to convert implicit summations into partially (containing only "spatial" dummy indices) or fully explicit (containing no dummy indices) forms. E.g., the fully explicit form of `S` reads

```
> ToComponents[S]

<  i1      i2      i3
< A     + A     + A
<    s1      s2      s3
```

- Commands to export expressions into plain TEX or LATEX form with flexible automatically generated line breaking and alignment commands. This property is very important if one works with long expressions and is unique among all other indicial tensor packages known to the author. Here is an example of output with the LATEX alignment commands:

```
> T = S + ( S /. { i->s, s->i } );
> PrintES[T]

<  sa      ia
< A     + A
<    ia      sa

> $TeXDialect=$LaTeX;
> $TermsPerLine=1;
> EinS2TeX[T]

< \begin{eqnarray}
< &&
< A^{sa\ \ }_{\ \ ia}
< \nonumber\\
```

```
< &&
< +A^{ia\ \ }_{\ \ sa}
< \nonumber
< \end{eqnarray}
```

This output can be saved into a file and then edited with a plain text editor to get desired LaTeX or TeX layout. Usually only a few changes are necessary (say, inserting `\label`, and deleting unnecessary `\nonumber`).

- A set of commands to save definitions of objects and expressions in a form which can be read back into EinS.
- A set of low level commands which can be used by experienced users to implement any kind of additional operations with indexed objects.
- Several built-in procedures simplifying debugging of the user programs.
- Online help messages for all commands.

All functions of EinS can work automatically with power series (say, in powers of c^{-1}) which allows one to use EinS effectively when working within typical approximation schemes.

3 A typical application of EinS: local reference systems in the PPN formalism

Typical applications for which EinS has been actually conceived are various calculations in [the post-Newtonian approximation of] metric gravity theories. Several important problem in this area has been solved with the help of EinS. As an example, we describe below the problem of definition of a local reference system of a massive extended body in the Parametrized Post-Newtonian (PPN) formalism.

3.1 Local reference systems of a massive extended body

The problem can be described in the following way. From the physical point of view any reference system covering a region of space-time under study can be used to describe physical phenomena within that region. However, a reference system offering a simpler mathematical description of physical laws is obviously more convenient for practical calculations. For a test observer one can construct a "proper" local reference system where gravitational field appears only as tidal (i.e., second-order with respect to spatial coordinates) potentials. In that proper reference system the coordinates in the infinitesimal vicinity of

the observer (that is, of the origin of the reference system) are directly related to the observable quantities. For many applications related with physically adequate description of astronomical observations it is important to construct similar "proper" local reference system for a massive extended body (for example, for the Earth as a whole). In general relativity this problem has been treated in the Brumberg-Kopeikin [1,15,14] and Damour-Soffel-Xu (DSX) [2] formalisms. In both approaches a local reference system (T, X^i) of a massive extended body has the following two properties:

(A) the gravitational field of external bodies is represented in the form of tidal potentials being $\mathcal{O}(\mathbf{X}^2)$;

(B) the internal gravitational field of the body coincides with the gravitational field of a corresponding isolated source provided that the tidal influence of the external matter is neglected.

Generalization of the formalisms onto the framework of the PPN formalism is a very important problem. Modern astronomical observations represent one of the most important sources of observational data for testing general relativity and for estimating the PPN parameters (e.g., β and γ). On the other hand, for many kinds of astronomical observations the local reference system of the Earth should be used in their relativistic models in order to get physically meaningful information from that observations. A good example is the relativistic model for Very Long Baseline Interferometry. This technique is used in particular to establish an Earth-bound reference frame, i.e. to derive numerical values for the coordinates of observing sites on the Earth with the highest possible accuracy being of order of 5 mm in the late 1990. These coordinates are than used to investigate the dynamics of the Earth's crust, irregularities of the rotational motion of the Earth, etc. Needless to say, that the coordinates should be derived with respect to a rigorously defined local geocentric reference system to efface any spurious coordinate-dependent effects (e.g., Lorentz contraction) as much as possible (see, the two properties A and B above). Currently, the VLBI is the source of the best current estimate of the PPN parameter γ [3]. Therefore, in order to get physically meaningful estimates of the PPN parameters one also have to use a physically adequate local reference system for the Earth within the framework of the PPN formalism. Our several recent publications [12,13] are devoted to that subject. Below we describe briefly the technique we used to tackle the problem and the way we used EinS.

3.2 Metric tensors and coordinate transformations

We begin with the global PPN metric tensor of an isolated N body system in the form (see, e.g., [20])

$$
\begin{aligned}
g_{00} &= -1 + \frac{2}{c^2}\, w(t,\mathbf{x}) - \frac{2}{c^4}\, \beta\, w^2(t,\mathbf{x}) + \mathcal{O}(c^{-5}),\\
g_{0i} &= -\frac{2(1+\gamma)}{c^3}\, w^i(t,\mathbf{x}) + \mathcal{O}(c^{-5}),\\
g_{ij} &= \delta_{ij}\left(1 + \frac{2}{c^2}\, \gamma\, w(t,\mathbf{x})\right) + \mathcal{O}(c^{-4}),
\end{aligned}
\tag{1}
$$

where the gravitational potentials w and w^i are defined as volume integrals

$$
\begin{aligned}
w &= G \int \sigma(t,\mathbf{x}')\, \frac{1}{|\mathbf{x}-\mathbf{x}'|}\, d^3x'\\
&\quad + \frac{1}{2c^2}\, G\, \frac{\partial^2}{\partial t^2} \int \sigma(t,\mathbf{x}')\, |\mathbf{x}-\mathbf{x}'|\, d^3x' + \mathcal{O}(c^{-4}),
\end{aligned}
\tag{2}
$$

$$
w^i = G \int \sigma^i(t,\mathbf{x}')\, \frac{1}{|\mathbf{x}-\mathbf{x}'|}\, d^3x' + \mathcal{O}(c^{-2}),
\tag{3}
$$

$$
\sigma = \frac{1}{c^2}\left(T^{00} + \gamma T^{kk} + \frac{1}{c^2} T^{00}\,(3\gamma - 2\beta - 1)\, w\right) + \mathcal{O}(c^{-4}),
\tag{4}
$$

$$
\sigma^i = \frac{1}{c}\, T^{0i} + \mathcal{O}(c^{-2}),
\tag{5}
$$

and $T^{\alpha\beta}$ is the energy-momentum tensor in the global PPN reference system (t, x^i). We retain here only two PPN parameters: γ and β. This simplest version of the PPN formalism covers, nevertheless, most viable theories of gravitation. The σ and σ^i play the role of effective densities of the corresponding post-Newtonian gravitational potentials w and w^i.

We have written the metric tensor of the global PPN reference system without specifying an explicit form of the energy-momentum tensor. This is no more than a formal way to specify the global PPN metric tensor. The metric (1)–(5) coincides with the version of the PPN formalism described in [20] as well as with the version discussed in [18], provided that only two parameters γ and β are retained in both versions and the corresponding models for $T^{\alpha\beta}$ are substituted into (1)–(5). Effectively, we consider only those theories of gravitation which produce the metric (1)–(5) in the first post-Newtonian approximation for any energy-momentum tensor.

The material system is supposed to consist of N bodies which represent simply spatially bounded blobs of matter. The energy-momentum tensor $T^{\alpha\beta}$ is supposed to vanish outside these N areas occupied by the bodies. Potentials w and w^i are defined by (2)–(3) as volume integrals over the whole 3-dimensional space as defined in the global reference system (t, x^i). Splitting the area of integration into volume V of a selected body, for which we want to construct

the local reference system, and the remaining part of space, we split w and w^i into the internal potentials (the potentials of the body under consideration) and the external ones (the potentials due to the other bodies):

$$
\begin{aligned}
w(t,\mathbf{x}) &= w_{\mathrm{E}}(t,\mathbf{x}) + \overline{w}(t,\mathbf{x}),\\
w^i(t,\mathbf{x}) &= w^i_{\mathrm{E}}(t,\mathbf{x}) + \overline{w}^i(t,\mathbf{x}).
\end{aligned}
\tag{6}
$$

We assume the metric tensor of the local PPN reference system (T, X^a) of a selected body to be of the same functional form as that of the global reference system

$$
\begin{aligned}
G_{00} &= -1 + \frac{2}{c^2}\,W(T,\mathbf{X}) - \frac{2}{c^4}\,\beta\,W^2(T,\mathbf{X}) + \mathcal{O}(c^{-5}),\\
G_{0a} &= -\frac{2(1+\gamma)}{c^3}\,W^a(T,\mathbf{X}) + \mathcal{O}(c^{-5}),\\
G_{ab} &= \delta_{ab}\left(1 + \frac{2}{c^2}\,\gamma\,W(T,\mathbf{X})\right) + \mathcal{O}(c^{-4}).
\end{aligned}
\tag{7}
$$

The local gravitational potentials W and W^a are supposed to be split as

$$
W(T,\mathbf{X}) = W_{\mathrm{E}}(T,\mathbf{X}) + Q_a(T)\,X^a + W_{\mathrm{T}}(T,\mathbf{X}) + \frac{1}{c^2}\,\Psi(T,\mathbf{X}), \tag{8}
$$

$$
W^a(T,\mathbf{X}) = W^a_{\mathrm{E}}(T,\mathbf{X}) + \frac{1}{2}\,\varepsilon_{abc}\,C_b(T)\,X^c + W^a_{\mathrm{T}}(T,\mathbf{X}). \tag{9}
$$

The local internal gravitational potentials W_{E} and W^a_{E} have the same functional form (2)–(5) as their global counterparts w_{E} and w^i_{E}, but all the quantities should be taken in the local reference system

$$
\begin{aligned}
W_{\mathrm{E}} &= G\int\limits_V \Sigma(T,\mathbf{X}')\,\frac{1}{|\mathbf{X}-\mathbf{X}'|}\,d^3X'\\
&\quad + \frac{1}{2c^2}\,G\,\frac{\partial^2}{\partial T^2}\int\limits_V \Sigma(T,\mathbf{X}')\,|\mathbf{X}-\mathbf{X}'|\,d^3X' + \mathcal{O}(c^{-4}),
\end{aligned}
\tag{10}
$$

$$
W^a_{\mathrm{E}} = G\int\limits_V \Sigma^a(T,\mathbf{X}')\,\frac{1}{|\mathbf{X}-\mathbf{X}'|}\,d^3X' + \mathcal{O}(c^{-2}), \tag{11}
$$

$$
\Sigma = \frac{1}{c^2}\left(\mathcal{T}^{00} + \gamma\mathcal{T}^{aa} + \frac{1}{c^2}\mathcal{T}^{00}\,(3\gamma - 2\beta - 1)\,W\right) + \mathcal{O}(c^{-4}), \tag{12}
$$

$$\Sigma^a = \frac{1}{c}\mathcal{T}^{0a} + \mathcal{O}(c^{-2}), \tag{13}$$

$\mathcal{T}^{\alpha\beta}$ being the components of the energy-momentum tensor in the local reference system. The external potentials $W_{\rm T}$ and $W^a_{\rm T}$ represent tidal gravitational field of the other bodies of the system and are assumed to be $\sim \mathcal{O}(\mathbf{X}^2)$. The function $Q_a(T)$ represents the acceleration of the instantaneous locally inertial reference system (whose origin coincides with that of the local reference system (T, X^i) at a given moment of time T) expressed in the local reference system. The function $C_a(T)$ describes the spatial orientation of the local reference system with respect to the global one [1,15,2].

Finally, the function $\Psi(T, \mathbf{X})$ is some unknown function containing internal potentials of the central body which appear in the local PPN reference system in addition to $W_{\rm E}$ and $W^a_{\rm E}$. Clearly the appearance of Ψ is related with a violation of the Equivalence Principle which make it impossible to satisfy simultaneously properties A and B of the local reference system formulated above. By assuming that $W_{\rm T}$ and $W^a_{\rm T}$ are $\sim \mathcal{O}(\mathbf{X}^2)$ we assume that property A is satisfied. Therefore, property B may be violated which results apriori in the appearance of Ψ.

The transformations between the global and local reference systems are supposed to have the form

$$\begin{aligned} T = t &- \frac{1}{c^2}\left(A + v^i_{\rm E} r^i_{\rm E}\right) \\ &+ \frac{1}{c^4}\left(B + B^i r^i_{\rm E} + B^{ij} r^i_{\rm E} r^j_{\rm E} + C(t, \mathbf{x})\right) + \mathcal{O}(c^{-5}), \end{aligned} \tag{14}$$

$$X^a = R^a_j\left(r^j_{\rm E} + \frac{1}{c^2}\left(\frac{1}{2} v^j_{\rm E}\, v^k_{\rm E} r^k_{\rm E} + D^{jk}\, r^k_{\rm E} + D^{jkl}\, r^k_{\rm E}\, r^l_{\rm E}\right)\right) + \mathcal{O}(c^{-4}), \tag{15}$$

where $r^i_{\rm E} = x^i - x^i_{\rm E}(t)$, $x^i_{\rm E}(t)$ is the coordinates of the origin of the local reference system relative to the global one, and $v^i_{\rm E} = dx^i_{\rm E}/dt$. The functions $A(t)$, $B(t)$, $B^i(t)$, $B^{ij}(t) = B^{ji}(t)$, $D^{ij}(t) = D^{ji}(t)$, $D^{ijk}(t) = D^{ikj}(t)$, $R^a{}_j(t)$ (being orthogonal matrix) and $C(t, \mathbf{x}) \sim \mathcal{O}(r^3_{\rm E})$ are some unknown functions to be derived later.

3.3 The derivation of the unknown functions and the use of EinS

Now one should check the consistency of all the assumptions on the structure of the local metric tensor and the coordinate transformations. This check is

made by applying the usual transformation law of the covariant metric tensor

$$g_{\alpha\lambda}(t,\mathbf{x}) = \frac{\partial X^\mu}{\partial x^\alpha}\,\frac{\partial X^\nu}{\partial x^\lambda}\,G_{\mu\nu}(T,\mathbf{X}). \tag{16}$$

This equation enables one to derive or constrain the unknown functions Q_a, C_a, W_T, W_T^i, Ψ from the local PPN metric tensor (7)–(12) and the unknown functions $x_E^i(t)$, A, B, B^i, B^{ij}, $C(t,\mathbf{x})$, D^{ij}, D^{ijk}, $R^a{}_i$ from the transformations (14)–(15).

For this work we made an extensive use of EinS and it was one pleasant week of programming to solve the problem with β and γ [12,13] instead of several months of tedious manual computation for the simpler case $\beta = \gamma = 1$ [14]. Moreover, since the problem in the PPN formalism is much more subtle than in general relativity it was necessary to perform matching many times with various forms of the local metric tensor and the transformations. Here, the help of EinS was crucial. Let us sketch the program for EinS we which actually used. Here, some of the output from EinS is suppressed which is indicated by "`< ...`".

- First of all, we define the coordinates of the global reference system with the command `DefRS[x,t]` exactly as we showed in Section 2.
- Then we define the metric tensors $g_{\alpha\beta}$ and $G_{\alpha\beta}$ as well as the local coordinates X^α as abstract objects

```
> DefObject[g,2,{1,2},{},{1,2}]
> DefObject[G,2,{1,2},{},{1,2}]
> DefObject[X,1]
> X[0]= T/c1;

< ...
```

  Here and below `c1` stands for c^{-1}.

- Then matching (16) is described by the following procedure

```
> match[i_,j_]:= g[i,j] -
>                ComputeES[
>                   SplitTime[
>                      DefES[ PD[X[k],x[i]] *
>                             PD[X[l],x[j]] *
>                             G[k,l],
>                             {k,l},
>                             ESRange -> $ESDimension]]]
```

Here, parameter `ESRange -> $ESDimension` tells EinS that the dummy indices in the summation should run from 0 to 3 and command `SplitTime` performs "3+1" split of those indices. For example,

```
> PrintES[match[0,0]]
```

```
<    -2      a    b                -1      a                  2
< -(c   G   X    X   ) + g   - 2 c   G   X    T   - G   T
<        ab   ,t   ,t     00          0a   ,t  ,t    00  ,t
```

- The various objects entering the metric tensors and the transformations as well as some their properties should then be defined. It is done in the same way as we showed in Section 2. For example, the whole definition of r^i_E, v^i_E and $a^i_E = dv^i_E/dt$ reads

```
> DefObject[re,1]
> DefObject[ve,1]
> DefObject[ae,1]
> Format[re,OutputForm] = SequenceForm[r,Subscript[E]]
> Format[ve,OutputForm] = SequenceForm[v,Subscript[E]]
> Format[ae,OutputForm] = SequenceForm[a,Subscript[E]]
> PD[ve[i_],t] := ae[i]
> PD[re[i_],t] := -ve[i]
> PD[re[i_],x[j_]] := Delta[i,j]
> DeclareIndependent[ve,1,{x}]
> DeclareIndependent[ae,1,{x}]

< ...
```

  The last two commands declare `ve[i]` and `ae[i]` to be independent of spatial coordinates `x[i]` of the global reference system.

- With similar definitions for the other objects the coordinate transformations can be written as

```
> T := t - c1^2 ( A + DefES[ ve[i] re[i], {i} ] )+
>          c1^4 ( B + DefES[ B[i] re[i], {i} ] +
>                 DefES[ B[i,j] re[i] re[j], {i,j} ] + C ) +
>          O[c1]^5
```

  Now it is easy to compute, for example, partial derivatives of T with respect to t and x^i:

```
> PrintES[ComputeES[PD[T,t]]]
```

```
<          a   a      a  a         -2
< 1 + (-(a   r  ) + v  v   - A  ) c   +
<         E   E      E  E     ,t
<
<    a     a    ab     a   b    a   a      ab   b   a
<  (B    r   + B     r   r   - B  v   - 2 B   r   v   + B
<     ,t  E       ,t  E   E        E           E   E     ,t
<
<         -4      -1 5
< + C  ) c   + O[c  ]
<    ,t
```

```
> PrintES[ComputeES[PD[T,x[i]]]]
```

```
<     i  -2      i      ia   a         -4      -1 5
< -(v   c  ) + (B  + 2 B   r   + C  ) c   + O[c  ]
<    E                     E     ,i
```

- Then we define the structure of the metric tensors. E.g.,

```
> g[0 ,0 ] := -1 + 2 c1^2 w - 2 c1^4 beta w^2 + O[c1]^5
> g[0 ,i_] := -c1^3 2 (gamma+1) w[i] + O[c1]^5
> g[i_,j_] := Delta[i,j] (1 + 2 gamma c1^2 w) + O[c1]^4
```

  as well as the split of the gravitational potentials w, w^i, W and W^i (Eqs. (6) and (8)–(9)).

- After all these definitions have been done, three calls of the procedure `match` (`match[0,0]`, `match[0,i]` and `match[i,j]`) give the desired equations which should vanish as implied by (16). These equations are to be further analyzed. Since the equations should vanish identically for any local body, any external potential and at any point of space-time, it is clear that (a) the terms containing internal potentials of the body (both local W_E and W_E^i, and global w_E and w_E^i) and those containing only external potentials should vanish separately; (b) the external terms can be expanded in powers of r_E^i and the coefficients at each power should be zero also separately. This split of the equations produced by `match` is performed by another special procedure. As a result we get a full set of equations for all the unknown functions.

The detailed results of the matching can found, e.g., in [13]. Here we note only with the help of EinS it has been proved that function Ψ in (8) read

$$\Psi(T,\mathbf{X}) = -\eta\,\left(w_{\rm E}(t,\mathbf{x})\left(\overline{w}(t,\mathbf{x}_{\rm E}(t)) + a^i_{\rm E} r^i_{\rm E}\right)\right.$$

$$-\chi^{\mathrm{E}}{}_{,i}(t,\mathbf{x})\,a^i_{\mathrm{E}}\Big) + \mathcal{O}(c^{-2}), \tag{17}$$

$$\chi^{\mathrm{E}} = \frac{1}{2}\,G \int\limits_V \sigma(t,\mathbf{x}')\,|\mathbf{x}-\mathbf{x}'|\,d^3x'. \tag{18}$$

Here $\eta = 4\beta - \gamma - 3$ is the Nordtvedt parameter, which is not zero only in the theories implying a violation of the Strong Equivalence Principle. This means that property B from Section 3.1 is violated unless $\eta \neq 0$. One can show that by changing the transformations (14)–(15) one can construct another version (many versions of matching!) of the local reference system where property B is valid, but property A is violated. One can also show that in a theory where the Strong Equivalence Principle is violated, it is impossible to construct a local reference system satisfying properties A and B from Section 3.1 simultaneously.

4 What EinS still cannot do

Although EinS can be used for a broad class of problems, potential users should not expect that EinS is too general. It was our intention to keep the package sufficiently compact and easy to tune up for a particular problem. Possible future developments of the package strongly depend on the scientific problems in which the author will be involved.

In order to give an idea of general direction of the possible improvements let us list some features already scheduled for the next releases of EinS:

- The simplification algorithm should be refined in three major directions:

(1) Splitting of dummy indices into subgroups which cannot intersect apriori when performing pattern matching in the algorithm of simplification. As a rough example let us consider the following expression:

$$T^{ijkl}\,S^{ij}\,V^{kl} + T^{ijkl}\,S^{jk}\,V^{il}. \tag{19}$$

Here, we suppose for simplicity that S^{ab} and V^{ab} have no symmetries. A naive simplification algorithm would try all 4!=24 permutations of the four dummy indices to check if the two terms can be simplified into a single one with a numerical factor. However, a more intellectual approach is to notice that there is only one factor S^{ab} in both terms. This means immediately that dummy index j in the second term must be unified with index i in the first term: $j_2 \to i_1$. In the same way for the second index of S^{ab} we have: $k_2 \to j_1$. In analogy for V^{ab}: $i_2 \to k_1$ and $l_2 \to l_1$. Then, the only thing to check is if T^{jkil} (the permutation of the indices resulting

from the unification (matching) of the indices of S^{ab} and V^{ab}) is equal to T^{ijkl} which depends on the symmetry properties of the object T^{abcd}. Mathematically this means to split the whole permutation group for the dummy indices into reasonable subgroups. The simplification algorithm implemented in the current version of EinS does the split (e.g., in the example above), but not always when it is possible.

(2) Handling more complicated symmetry properties including linear (and possibly non-linear) identities. An interesting algorithm for the simplification taking into account symmetries of the objects, the possibility to rename dummy indices and linear identities is described in [19,8,9]. The algorithm fits very well into the "philosophy" of EinS and we hope to implement a version of the algorithm in the future.

(3) Automatic consistency check of the symmetry properties. For example, A^{ijk} should be immediately recognized to be zero if it is defined as $A^{ijk} = A^{jik}$ and $A^{ijk} = -A^{ikj}$ [5]. More generally, the procedure of `DefObject` should calculate the number of independent components of each defined object (note that it is in general impossible if the object satisfies non-linear identities) and check if the specified symmetries imply some explicit numerical values for the components. In the example above the number of independent components is zero and the only possible values for all the components of A^{ijk} is zero.

- Another possible task is to implement a set of procedures dealing with symmetric trace-free (STF) objects (computing the STF part of an arbitrary expression, taking into account the property of an object to be STF during simplification, etc.). Symmetric trace-free objects (e.g., cartesian tensors) are getting more and more popular in gravitational physics [15,2] and, e.g., in celestial mechanics [6]. We are not aware of any software enabling one to operate with STF objects.

A further challenge is to write a package enabling one to deal with families of indexed objects: objects with similar symmetry properties, but different number of indices. Typical example here is the multipole expansion of the gravitational potential

$$U = \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} M_L \, \hat{w}^L, \tag{20}$$

where $L = i_1 \, i_2 \ldots i_l$ is a multiindex, $\hat{w}^L$ is the STF part of product $w^{i_1} \, w^{i_2} \ldots w^{i_l}$ and M_L is a family of the STF multipole moments of the gravitational field. The ability to operate with families of objects (like M_L) and multiindices (like L) requires a significant number of new procedures and changes of the exist-

ing ones. However, the design of EinS is flexible enough to accommodate these changes without re-writing the whole package.

Further details on EinS and its applications as well as the package itself with User Guide and sample *Mathematica* notebooks are available from the author and from the EinS's home page

`http://rcswww.urz.tu-dresden.de/~klioner/eins.html`

Acknowledgement

I am grateful to Dr. A.V. Voinov whose brilliant physical ideas pushed me to start writing EinS and Dr. R. Zalaletdinov whose questions forced me to release the first official version and to write the User Guide. I am also thankful to all the people who used EinS and made comments and suggestions.

References

[1] Brumberg, V. *Essential Relativistic Celestial Mechanics.* Adam Hilder, Bristol, 1991.

[2] Damour, T., Soffel, M., and Xu, C. General relativistic celestial mechanics. *Physical Review D 43*, 3273–3307, 1991; *Physical Review D 45*, 1017–1044, 1992; *Physical Review D 47*, 3124–3135, 1993.

[3] Eubanks, T.M., Matsakis, D.N., Martin, J.O., Archinal, B.A., McCarthy, D.D., Klioner, S.A., Shapiro, S. and Shapiro, I.I. Advances in Solar System Tests of Gravity. Bull. American Physical Society, **42**, 1107, 1997 (also available at `http://www.aps.org/BAPSAPR97/abs/S1280005.html`).

[4] Fock, V. *The Theory of Space, Time and Gravitation*, 2nd revised ed. Pergamon Press, Oxford, 1964.

[5] Hartley, D. Overview of computer algebra in relativity. In *Relativity and Scientific Computing*, F. Hehl, R. Puntigam, and H. Ruder, Eds. Springer, Berlin, 173–191, 1996.

[6] Hartmann, T., Soffel, M. and Kioustelidis, T. On the use of SFT-tensors in celestial mechanics. *Celestial Mechanics*, **60**, 139–159, 1994.

[7] Hehl, F.W. Workshop A.5: Computer Methods in General Relativity (ii) Algebraic Computing. In *Proceeding of the 14th International Conference on General Relativity and Gravitation* (Florence, 1995), M. Francaviglia,

G. Longhi, L. Lusanna, E. Sorace, Eds., World Scientific, Singapore, 469–475, 1997.

[8] Ilyin, V., and Kryukov, A. Symbolic simplification of tensor expressions using symmetries, dummy indices and identities. In *ISSAC'91, Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation* (Bonn, 1991), S. Watt, Ed., ACM Press, Singapore, 224–228, 1991.

[9] Ilyin, V., and Kryukov, A. A symbolic simplification algorithm for tensor expressions in computer algebra. *Programmirovanie (Computer Science)*, Nauka, Moscow, 83–91, January 1994, in Russian.

[10] Klioner, S. EinS: A Mathematica package for tensorial calculation in astronomical applications of relativistic gravity theories. In *Abstracts of GR14, the 14th International Conference on General Relativity and Gravitation* (Florence, 1995), M. Francaviglia, Ed., SIGRAV-GR14, Turin, A.182, 1995.

[11] Klioner, S. Angular velocity of extended bodies in general relativity. In *Dynamics, ephemerides and astrometry in the solar system*, S. Ferraz-Mello, B. Morando, and J. Arlot, Eds. Kluwer, Dordrecht, 309–320, 1996.

[12] Klioner, S., and Soffel, M. Local reference systems with PPN parameters. In *Proceedings of the 8th Marcell Grossmann Meeting* (Jerusalem, 1997), R. Ruffini, Ed., World Scientific, Singapore, 1998, in press.

[13] Klioner, S., and Soffel, M. The Nordtvedt effect in rotational motion. *Physical Review D*, 1998, accepted.

[14] Klioner, S., and Voinov, A. Relativistic theory of astronomical reference systems in closed form. *Physical Review D 48*, 1451–1461, 1993.

[15] Kopeikin, S. Relativistic reference systems in solar system. In *Itogi Nauki i Tekhniki*, M. Sazhin, Ed., **87**, Nauka, Moscow, 87–146, 1991.

[16] Landau, L., and Lifshits, E. *The Classical Theory of Fields.* Pergamon Press, Oxford, 1971.

[17] MacCallum, M. Symbolic computation in relativity theory. In *EUROCAL'87, European Conference on Computer Algebra* (Berlin, 1987), J. Davenport, Ed., Springer, Berlin, 34–43, 1987.

[18] Misner, C.W., Thorne, K.S., Wheeler J.A. *Gravitation.* Freeman, New York, 1973.

[19] Rodionov, A., and Taranov, A. Combinatorial aspects of simplification of algebraic expressions. In *EUROCAL'87, Proceedings of the European Conference on Computer Algebra* (Berlin, 1987), J. Davenport, Ed., Springer, Berlin, 192–201, 1987.

[20] Will, C. *Theory and experiment in gravitational physics*, revised ed. Cambridge University Press, Cambridge, 1993.