Proceedings of the 8th Marcel Grossmann Meeting, World Scientific, Singapore, 1998, in press

EinS: A *Mathematica* PACKAGE FOR COMPUTATIONS WITH INDEXED OBJECTS

Sergei A. Klioner
Lohrmann Observatory, Dresden Technical University, Mommsenstraße, 13, 01062 Dresden, Germany

A new *Mathematica* package EinS is intended for calculations involving sums of indexed objects (e.g., tensors). EinS automatically handles implicit summations and dummy indices, allows user to assign symmetry properties to new objects and has an efficient built-in simplification algorithm based on pattern matching. Further features include printing expressions in a 2-dimensional form, exporting into plain TEX or LATEX with user-controlled alignment commands, converting implicit summations into explicit ones, debugging capabilities and online help messages.

1 What is EinS?

EinS ("Einstein Summation handler") is a *Mathematica* package for operations with indexed objects (tensors or functions). EinS is a relatively small (about 3000 lines of *Mathematica* top level code), flexible package which is easy to alter for solving 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 tackled by the author during last several years. Typical calculations in this field represent operations with power series (e.g., in powers of c^{-1}) involving many 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 being widely used herewith. Another important property is that usually "time" and "spatial" values of indices are treated separately, and covariant and contravariant "spatial" indices are not distinguished.

It is well known that there exist several software systems for doing indicial tensorial computations on a computer[4,1]. However, all of the existing systems has their specific areas of application and none of them allowed us to handle automatically all the properties of our research field mentioned above. On the other hand, the algorithms of indicial tensor operations are tricky, and every new attempt to implement them is of interest. That is why even a small summary published about EinS attracted certain attention and a reference on EinS appeared in a recent review on applications of computer algebra in general relativity[1].

EinS (as well as *Mathematica* itself) is not intended to perform calculations with very large number of terms. The most cumbersome computation which we have done with the use of EinS was computing Landau-Lifshits pseudotensor for a generic metric containing terms of order of $\mathcal{O}(c^{-1})$ in g_{0i}. The calculations involved about 2000 terms in the result and up to 5000 in intermediate expressions. Each term was a product of up to 10 indexed objects, some of which were defined to be symmetric. Recent progress in hardware allows EinS to operate with 10000 or more terms depending on their structure. However, if one has to operate with much longer expressions (with ~ 100000 terms) STENSOR[1] is probably a better choice.

2 General structure of EinS

EinS works under *Mathematica* starting from version 2.1. This allows user to employ the whole power of one of the most comprehensive computer algebra systems, but also requires from the user some basic knowledge of the *Mathematica*'s user interface and its top level language. We describe below EinS by dividing its functions into several groups.

- Commands which represent several built-in objects. E.g., `LeviCivita[i,j,k]` and `Delta[i,j]` stand for the Levi-Civita and Kronecker symbols, respectively.
- Commands to define new indexed objects, their symmetries and their independence of certain parameters (say, coordinates or time). E.g., the command `DefObject[gcov,2,"g",{1,2},ESIndicesLow -> True]` defines an object `gcov` with 2 indices, which will be printed as `g` instead of `gcov`, which is symmetric, and both indices of which will be printed as subscripts. `DeclareIndependent[gcov,2,{t}]` defines it to be independent of some parameter `t`. EinS is not specifically intended for tensors and does not distinguish automatically covariant and contravariant indices. However, its flexible printing capabilities allow user to mimic rigorous tensor notations if it is necessary.
- Commands to declare particular symbols to represent coordinates of a reference system. E.g., `DefES[x,t]` defines `x[i]` and `t` to be coordinates and coordinate time $t = x^0/c$ of a reference system). This information is used primarily for outputting (say, $\frac{\partial U}{\partial x^i}$ will be printed as $U_{,i}$).
- Commands to define relations between objects including implicit summation rules. E.g., `A[i_,j_]:=DefES[Delta[i,k] Delta[k,j], {k}]` defines A^{ij} to represent $\delta_{ik}\,\delta^{kj}$. EinS allows user to declare each dummy index appearing in an implicit summation to be "space-time" (running from 0 to N, where in default case of 4-dimensional space-time $N = 3$) or purely "spatial" (running from 1 to N). EinS automatically distinguish these two kinds of indices which allows to use the package effectively when working with "3+1" split of space-time.
- Commands to simplify expressions. E.g., `B[i,j]=ComputeES[A[i,j]]` results in $B_{ij} = \delta_{ij}$. EinS simplifies the expressions into several steps: (1) transforming each term into a simple "canonical" form (equal terms may still have different forms after this operation, but it decreases substantially the number of terms); (2) subsequent *pattern matching* among the rest of terms with the full account for symmetries of the objects and the possibility to rename dummy indices; (3) simplification of the built-in objects by using their pre-defined properties (say, $\delta_{ij}\,\delta_{jk} = \delta_{ik}$); (4) checking if each term of the result can be further simplified in virtue of some additional circumstances (e.g., $A^{ij}\,B^{ij}$ will be simplified to zero if A is symmetric and B is antisymmetric). The steps can be executed manually one by one or automatically in a reasonable sequence.
- Commands to print expressions in natural 2-dimensional form. User can define if a particular index of an object is to be printed as sub- or superscript. EinS makes full use of the new *Mathematica* 3.0 interface (e.g., by printing "space-time" dummy indices as greek letters).

- Commands to convert implicit summations into partially (only "spatial" dummy indices) or fully explicit (no dummy indices) form.
- 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.

Besides that, EinS has several built-in procedures simplifying debugging and online help messages for all user 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 some typical approximation scheme.

3 Future prospects

It was our intention to keep the package sufficiently small and easy to tune up for a particular problem. That is why, potential users should not expect that EinS is too general. Further developments of the package strongly depend on the scientific problems in which the author will be involved. Most probable future improvement of EinS is refining the simplification algorithm in three major directions: (1) splitting of dummy indices into subgroups which cannot intersect apriori when performing pattern matching in the algorithm of simplification; (2) handling more complicated symmetry properties including linear and possibly non-linear identities[5,2,3]; (3) automatical consistency check of symmetry properties (e.g., A^{ijk} should be recognized to be zero if it is defined as $A^{ijk} = A^{jik}$ and $A^{ijk} = -A^{ikj}$ [1]). Writing a special package on the basis of EinS for calculation of Christoffel symbols, curvature tensor, covariant derivatives, etc., for a given metric is another possible development of EinS. Implementing such a package is rather trivial task, and many potential pieces of such a package are distributed over various applications of EinS.

Further details on EinS as well as EinS itself are available from the author and from its home page `http://rcswww.urz.tu-dresden.de/~klioner/eins.html` .

References

1. Hartley, D. (1996) In: Hehl, F.W., Puntigam, R.A., Ruder, H. (eds.) *Relativity and Scientific Computing*, 173, Springer, Berlin
2. Ilyin, V.A., Kryukov, A.P. (1991) In: Watt, S.M. (ed.) *ISSAC'91, Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, ACM Press, Singapore, 224
3. Ilyin, V.A., Kryukov, A.P. (1991) *Programmirovanie* (Computer Science), January 1994, Nauka, Moscow, 83 (in Russian)
4. MacCallum, M.A.H. (1987) In: Davenport, J.H. (ed.) *EUROCAL'87, European Conference on Computer Algebra*, Springer, Berlin, 34
5. Rodionov, A.Ya., Taranov, A.Yu. (1987) In: Davenport, J.H. (ed.) *EUROCAL'87, Proceedings of the European Conference on Computer Algebra*, Springer, Berlin, 192