**Publ. Astron. Obs. Belgrade No. 88 (2010), 101 - 109** *Contributed paper*

# SYMBOLIC COMPUTATION METHODS IN COSMOLOGY AND GENERAL RELATIVITY

D. N. VULCANOV

*West University of Timişoara*
*Theoretical and Applied Physics Dept. - "Mircea Zăgănescu"*
*Bd-ul V. Pârvan nr. 4, Timişoara, Romania*

*E–mail vulcan@physics.uvt.ro*

**Abstract.** This paper is a review based on recent published articles by the author (Vulcanov, 2001; 2003; 2004; 2008), and is dedicated to the use of algebraic computing in general relativity (GR) and cosmology.

## 1. INTRODUCTION

The main goal of this series of lectures given at SSSCP2009 was to introduce the audience in the field of the computer algebra applied in general relativity and cosmology. The article is organised as follows : the next Section no. 2 describes the use of computer algebra and computer algebra systems in GR and the free package GrTensorII for Maple platform designed to manage symbolic computation in riemannian geometry as it is used in GR. The 3-d section of the article is dedicated to a short description of a package-library (named Cosmo) specially composed for cosmology, developped from GrTensorII (Vulcanov, 2004). The last section describes some simple applications of the Cosmo library in cosmology with scalar field and the reverse engineering method (Vulcanov, 2008).

## 2. USING MAPLE AND GRTENSORII IN GENERAL RELATIVITY

### 2. 1. SOME WORDS BEFORE ...

In the last years computational physics has developped as a new branch of physics and not only ! In this aspect today we have computational relativity (CR) which can be described simply as doing general relativity (GR) on the computer (Hehl, 1996; Grabmeier et al., 2001). We can identify two different branches of CR namely computer algebra (see below) for GR and numerical simulations in GR. It can be easily seen a serious gap between the two brancehs of CR, by the fact that mainly the results of simbolic computations are tranposed in codes for numerical simulations mainly by hand, i.e. we need here a "human interface".

But the spectacular developpment of computer technology makes possible to fill this gap in the very next future. Already are important researches with promising results in implementing special routines and programmes for generating C or Forran codes inside a computer algebra platform (Grabmeier et al., 2001). The things are moving very fast in both hardware and software technology. The software evolved a lot from the early days of programming, so today we have : object oriented programming, visual techniques, integrated platforms (as Maple, Mathematica...), graphical and visualisation software, games on the computer etc. But the programming languages did not moved so fast : we are still using Fortran, C or even C++... Anyway the progress is visible towards a meta-language of programming, all the old and new languages (as **perl** or **java** for example) including facilities from other languages.

## 2. 2. COMPUTER ALGEBRA (GRABMEIER, 2001)

Computer algebra - CA, is a branch of symbolic computation (or symbolic mathematics - SM) Symbolic mathematics relates to the use of computers to manipulate mathematical equations and expressions in symbolic form, as opposed to manipulating the approximations of specific numerical quantities represented by those symbols. Such a system might be used for symbolic integration or differentiation, substitution of one expression into another, simplification of an expression, etc.(Wikipedia Free encyclopedia - http://www.wikipedia.org)

Related fields and domains to CA can be identified as : LISP and LISP programming, computational logic, automated theorem prover or computer-aided proof, artificial intellingence...

What a Computer Algebra System is ? A computer algebra system ( CAS) is a software program that facilitates symbolic mathematics. The core functionality of a CAS is manipulation of mathematical expressions in symbolic form. The expressions manipulated by CAS typically include polynomials in multiple variables; standard functions of expressions (sine, exponential, etc.); various special functions (gamma, zeta, erf, Bessel, etc.); arbitrary functions of expressions; derivatives, integrals, sums, and products of expressions; truncated series with expressions as coefficients, matrices of expressions, and so on. (This is a recursive definition.)

The symbolic manipulations supported typically include :

- automatic simplification, including simplification with assumptions
- substitution of symbolic or numeric values for expressions
- change of form of expressions: expanding products and powers, rewriting as partial fractions, rewriting trigonometric functions as exponentials, etc.
- differentiation with respect to one or all variables

The symbolic manipulations supported typically include also : symbolic constrained and unconstrained global optimization, partial and full factorization, solution of linear and non-linear equations over various domains, solution of differential and difference equations, taking some limits, indefinite and definite integration, including multidimensional integrals, integral transforms, expansion as truncated Taylor, Laurent and Puiseux series, some infinite series expansion, series summation, matrix operations including products, inverses, etc., display of mathematical expressions in two-dimensional mathematical form, often using typesetting systems.

In addition, most CAS include numeric operations as : evaluating for particular numeric values, evaluating to high precision (bignum arithmetic), allowing for instance the evaluation of 21/3 to 10,000 digits, numeric linear algebra, plotting graphs and parametric plots of functions in two and three dimensions

Many also include a high level programming language, allowing users to implement their own algorithms. The study of algorithms useful for computer algebra systems is known as computer algebra.

The run-time of numerical programs implemented in computer algebra systems is normally longer than that of equivalent programs implemented in systems such as MATLAB, GNU Octave, or directly in C, since they are programmed for full symbolic generality and thus cannot use machine numerical operations directly for most of their functions.

Computer algebra systems began to appear in the early 1970's, and evolved out of research into artificial intelligence, though the fields are now regarded as largely separate. The first popular systems were Reduce, Derive, and Macsyma which are still commercially available; a copyleft version of Macsyma called Maxima is actively being maintained.

The current market leaders are Maple and Mathematica; both are commonly used by research mathematicians, scientists, and engineers. MuPAD is a commercial system too. Some computer algebra systems focus on a specific area of application; these are typically developed in academia and free.

We can give below a list of most used CAS organised in three categories, namely :

**Commercial sofware** : Derive, DoCon, Maple, MathCad, Mathematica, MuMATH, MuPAD, Reduce, WIRIS

**Free / open source software** : Axiom, dcas, Eigenmath, GiNaC, Mathomatic, Maxima, Yacas, SHEEP,

**Algebraic geometry, polynomial computations** : CoCoA, Macaulay, SINGULAR

## 2. 3. COMPUTER ALGEBRA (SYSTEMS) AND GENERAL RELATIVITY

General Relativity and Gravitation (GRG) is a theory of the dynamics of space-time, based on the differential geometry which implies long and complicated analytic computation, tensor manipulations, covariant derivatives and manny other geometrical ... ingredients !

Thus the use of CA in this field was very interesting and challenging from the very early beginning of all CAS (Hehl, 1996; Grabmeier, 2001).

CA is providing GRG new fast computational tools on the computer and on the other side, the developping of CA pushed also the research in GRG field. Several CAS were used during the precedent decades, following the developpment of computer technology, of course. Examples : REDUCE, Maple, Maxima and even Mathematica (?!)

Some CAS were specially designed for their use in GRG : SHEEP as an example. Two CAS proved to be the most viable, having the largest spectrum of applicability and are now the most used : REDUCE and MAPLE.

As in the next of this article we will concentrate on applications for GRG written and addapted to the Maple and GrTensor II we will describe now the Maple platform.

Maple (Maplesoft, 2005) is a general-purpose commercial computer algebra system. It was first developed in 1981 by the Symbolic Computation Group at the University of Waterloo in Waterloo, Ontario, Canada. Since 1988, it has been developed and sold commercially by Waterloo Maple Inc. (also known as Maplesoft), a Canadian company also based in Waterloo, Ontario. The current version is Maple 12. Maple is an interpreted, dynamically typed programming language. As is usual with computer algebra systems, symbolic expressions are stored in memory as directed acyclic graphs. Since Maple 6 the language has permitted variables of lexical scope.

## 2. 4. MAPLE+GRTENSORII

GrTensor II is a computer algebra package for performing calculations in the general area of differential geometry. Its purpose is the calculation of tensor components on curved spacetimes specified in terms of a metric or set of basis vectors. Though originally designed for use in the field of general relativity, GRTensorII is useful in many other fields. GRTensor II is not a stand alone package, but requires an algebraic engine. The program was originally developed for MapleV. GRTensorII runs with all versions of Maple, Maple V Release 3 to Maple 12. A limited version (GRTensorM) has been ported to Mathematica. GRTensor II and related software and documentation are distributed **free of charge** as an aide for both research and teaching; see at http://grtensor.org . the authors are : Peter Musgrave, Denis Pollney and Kayll Lake.

The geometrical environment for which GrTensorII is designed is a Riemannian manifold with connection compatible with the riemannian metric. Thus there are special commands and routines for introducing and calculating geometrical objects as the metric, Christoffel symbols, curvature (Ricci tensor and scalar) and the Einstein tensor - as for a couple of examples. Manipulating with indices and extracting tensor components are easy to do from some special commands and conventions. GrTensorII has a powerful facility for defining new tensors, using their natural definitions. GrTensorII provides also an entire collection of predefined metrics, but the user can also define his owns using a simple command **grmap**. The package provides access to all symbolic computing facilities of Maple (as simplifying options, collectiong terms etc.) through a single **gralter** command. Actually the user is free to choose his own simplification strategy inside these commands.

Let us iustrate how GrTensorII works with a simple example : calculting the Bianchi identities for a certain riemannian metric :

$$G^i_{j;i} = 0 \quad \text{where} \quad G^i_j = R_{ij} - \frac{1}{2} g_{ij} R$$

is the Einstein tensor, $R_{ij}$ and $R$ are the Ricci tensor and scalar, $g_{ij}$ being the metric and ; means the covariant derivative. This can be transposed in a sequence of Maple+GrTensorII commands as

```
> grtw();
> qload(rob_sons);
> grdef(‘bia{ ^i }:=G{ ^i ^j ;j }‘);
> grcalc(bia(up));
> gralter(bia(up),simplify);
> grdisplay(bia(up));
```

The first two commands above initialize GrTensorII and loads a predifined metric (Robertson-Walker metric - see below). The next commands just defines, calculates and simplifies the new defined Bianchi tensor. If the metric is compatible with the riemannian connection the components of the Bianchi tensor must vanish. For the Robertson-Walker metric above (called "rob_sons") it gives:

```
For the rob_sons  spacetime:
bia(up)
bia(up) = All components are zero
```

For recent results in applying GrTensor II in different areas of GRG one can check on the website of the package at http://grtensor.org inside sections "Papers" and "Demonstrations" (examples : exact solutions of Einstein eqs, Newman-Penrose null tetrad formalism...) Application to canonical ADM formalism can be found at the same website (Vulcanov demonstrations)

For application in the study of Dirac equation on curved spacetimes with/without torsion see (Vulcanov, 2003) and the references cited there. For application of GrTensorII in cosmology see (Vulcanov, 2004) describing the Cosmo library, wich will be also the content of the next section below.

## 3. MAPLE+GRTENSORII PROGRAMS FOR COSMOLOGY (VULCANOV, 2004)

### 3. 1. WHY WE NEED COSMO LIBRARY ?

The aswer to the question in title comes from the new developpment of theoretical cosmology in the last years. New facts from astrophysical measurements show that the Universe is in an accelerated expansion - "cosmic acceleration" (Perlmutter et. al., 1999). The standard model of the Universe must be reformulated ! One of the solutions is to introduce one or more scalar fields minimally coupled with the gravity to trigger this expansion. New models are proposed daily, demanding new specific computational tools, computer algebra packages included. This is the purpose of our Cosmo library : to provide CA tools for theoretical cosmology, in an environment having all the cosmological parameters and functions defined ! Is entirely done using Maple + GrTensorII ! We will describe Cosmo library in the next subsections after a short introduction to cosmology.

### 3. 2. COSMOLOGY (ISLAM, 1992)

In modern cosmology we are using the Friedmann-Robertson-Walker metric (FRW), having the line element in spherical coordinates

$$ds^2 = -c^2dt^2 + R(t)^2\left[\frac{dr^2}{1-kr^2} + r^2(d\theta^2 + \sin^2\theta\, d\phi^2)\right] \tag{1}$$

as a generic metric for describing the dynamics of the universe. Above $k = -1, 0, 1$ and $R(t)$ si th scale factor of the universe. The dynamic equations (called Friedmann equations) are obtained introducing (1) in the non-vacuum Einstein equations

$$G_{ij} = R_{ij} - \frac{1}{2}g_{ij}R + \lambda G_{ij} = \frac{8\pi G}{c^4}T_{ij}$$

having the matter content of the universe given by the stress-energy tensor $T^{ij} = T_m^{ij} + T_\phi^{ij}$ which contains terms comming from the massless scalar field minimally coupled with gravity ($\phi$) and the matter content of the universe as a perfect fluid. Thus we have :

$$T_\phi^{ij} = (p_\phi + \rho_\phi) u^i u^j + p_\phi g^{ij}$$

where as for a perfect fluid we have for the pressure and the density :

$$p_\phi = -\frac{1}{2}\partial^j\partial_j\phi - \frac{1}{2}V(\phi) \quad \text{and} \quad \rho_\phi = -\frac{1}{2}\partial^j\partial_j\phi + \frac{1}{2}V(\phi)$$

where $V(\phi)$ is the potential of the scalar field. Similar expressions are valid for the matter content of the universe (oher than he scalar field) described again as a perfect fluid. To complete this introduction, the main cosmological parameters and functions, namely the Hubble "constant" and the deceleration function are defined as :

$$H(t) = \frac{\dot{R}(t)}{R(t)} \;\; ; \;\; Q(t) = -\frac{\ddot{R}(t)}{2H(t)^2 R(t)}$$

## 3. 3. THE COSMO LIBRARY DESCRIBED

With the above definitions and equations a sequence of GrTensorII and Maple commands for all this follows :

```
> restart;grtw();qload(rob_sons);
> grdef(`Scal := Phi(t)`);
> grdef(`T1{ i j } := Scal{ ,i } * Scal{ ,j } -
     g{ i j } * (g{ ^a ^b }*Scal{ ,a }*Scal{ ,b }+ V(t) ) / 2 `);
> pphi(t):=diff(Phi(t),t)^2/2/c^2-V(t)/2;
> epsilonphi(t):=diff(Phi(t),t)^2/2/c^2+V(t)/2;
> grdef(`T2{ i j } := (epsilon(t) + p(t))*u{ i }*u{ j } +
                                    p(t) * g { i j } `);
> grdef(`T{ i j } :=T1{ i j } + T2{ i j }`);
> grdef(`cons{ i }:= T{ i ^j ;j }`); grcalc(cons(dn));
> EcuKG:=grcomponent(Box[Scal],[]) -DV(t)/2;
> grdef(`Ein{ i j } := G{ i j } - 8*Pi*G*T{ i j }/ c^4`);
> grcalc(Ein(dn,dn)); gralter(Ein(dn,dn),expand);
```

Where **pphi(t)** , **epsilonphi(t)** , **p(t)** and **epsilon(t)** are the pressure and density of the scalar field and matter field, respectively. With **cons()** we denoted the conservation law components for the total stress-energy tensor. The Klein-Gordon equation is **EcuKG** and **DV(t)** is the derivative of the potential in terms of the scalar field !!!

Then follows a sequence of boring **grcomponent** and **gralter** commands to extract the main cosmological equations. Finally we substitute the cosmological functions in the resulting MAPLE expressions :

```
> Ecunr1:=expand(simplify(subs(k=K(t)*RR(t)^2,Ecunr1)));
> Ecunr2:=expand(simplify(subs(k=K(t)*RR(t)^2,Ecunr2)));
> Ecunr1:=subs(diff(RR(t),t)=H(t)*RR(t),Ecunr1);
> Ecunr22:=subs(diff(RR(t),t,t)=-2*H(t)^2*RR(t)*Q(t),Ecunr2);
> Ecunr22:=subs(diff(RR(t),t)=H(t)*RR(t),Ecunr22);
> Ecunr2:=subs(diff(RR(t),t)=H(t)*RR(t),Ecunr2);
> Ecunr2:=expand(Ecunr2);
> Ecunr2:=subs(diff(RR(t),t)=H(t)*RR(t),Ecunr2);
> Ecunr3:=subs(diff(RR(t),t)=H(t)*RR(t),Ecunr3);
> EcuKG:=subs(diff(RR(t),t)=H(t)*RR(t),EcuKG);
```

Actually all these commands are transferred in Maple expressions/objects the main comological functions, defined and calculated in GrTensorII. As a result the user will not use anymore GrTensorII !!!

Thus the program is providing the classical Friedmann equations (3), (4) abd (5) together with Klein-Gordon equation (2) and the conservation law (6) below :

$$\frac{1}{c^2}\left[\ddot{\phi}(t) + 3H(t)\dot{\phi}(t)\right] + \frac{1}{2}DV(t) = 0 \tag{2}$$

$$3H(t)^2 + 3c^2K(t) - \frac{4\pi G}{c^4}\left[\dot{\phi}(t)^2 + c^2V(t) + 2c^2\epsilon(t)\right] = 0 \tag{3}$$

$$\begin{aligned}2\dot{H}(t) + 3H(t)^2 + c^2K(t) + \frac{4\pi G}{c^4}\Big[\dot{\phi}(t)^2 \\ - c^2V(t) + 2c^2p(t)\Big] = 0\end{aligned} \tag{4}$$

$$\begin{aligned}H(t)^2(1-4Q(t)) + c^2K(t) + \frac{4\pi G}{c^4}\Big[\dot{\phi}(t)^2 \\ - c^2V(t) + 2c^2p(t)\Big] = 0\end{aligned} \tag{5}$$

$$\begin{aligned}\frac{1}{c^2}\left[\ddot{\phi}(t)\dot{\phi}(t) + 3H(t)\dot{\phi}(t)^2\right] + \frac{1}{2}\dot{V}(t) + \dot{\epsilon}(t) + \\ 3H(t)(p(t)+\epsilon(t)) = 0\end{aligned} \tag{6}$$

All this stuff is then saved in a library called **cosmo.m** and can be loaded for fast processing. As a result all the facilities of MAPLE + GrTensorII are available together with the cosmology environment described above. The library is kept in the main directory of GrTensorII (Grtii(6) for Windows versions or grii in Unix implementations. Then it can be loaded in a Maple worksheet using **read** or **load** commands.

Note : intermediate components of the **Ein(i,j)** tensor are kept also (apart from **Ecunr1**...**Ecun3** and **EcuKG** objects) for later use and processing. This makes easy the development of the library for more applications

A simple example of how to use and developp the Cosmo library follows !

## 4. REVERSE ENGINEERING METHOD IN COSMOLOGY WITH SCALAR FIELD

In the standard treatment of cosmological models with scalar field, it is prescribed a certain potential function for the scalar field (taking into account some physical reasons specific to the model processed) and then the dynamical Friedmann equations are solved (if it is possible) to obtain the time behavior of the scale factor of the universe. As recently some authors pointed out (Ellis and Madsen, 1991; Ellis et al., 2004), a somehow "reverse" method is also interesting, where the time behavior of the scale factor is "a priori" prescribed (as a function of time which will model the supposed time behavior of the universe in inflation or in cosmic accelerated expansion). Then solving the Friedmann equations we can extract the shape of the corresponding potential for the theory. This is the so called "reverse engineering" method (REM) and we shall use it here to illustrate the usage of our **cosmo.m** library in a simple case, more complicated cases, including achionic potentials can be found in (Vulcanov, 2008).

A first step in the REM : simply solve two of the above Friedmann eqs. to get :

$$V(t) = \frac{1}{4\pi}\left[\dot{H}(t) + 3H(t)^2 + \frac{2k}{R(t)^2}\right]$$

$$\dot{\phi}^2 = \frac{1}{4\pi}\left[-\dot{H}(t) + \frac{k}{R(t)^2}\right]$$

Here we have the scalar field as the only matter content of the Universe and geometrical units : c=G=1 For a DeSitter exponential expansion, namely

$$R(t) = R_0 e^{\omega t} \quad ; \quad H(t) = \omega$$

we have, after a sequence of subs and simplify commands :

$$V(t) = \frac{3\omega^2}{4\pi} + \frac{k}{2\pi e^{2\omega t}} \quad ; \quad \dot{\phi}(t)^2 = \frac{k}{4\pi e^{2\omega t}}$$

Then, from the last expression we have :

$$\phi(t) = -\frac{1}{2}\frac{\sqrt{k}e^{-\omega t}}{\sqrt{\pi}\omega} + \phi_0$$

The result : equations Ecunr1...Ecunr3 are automatically satisfied and the Klein-Gordone equation becomes

$$\mathbf{EcuKG} = \frac{\sqrt{k}\omega}{\sqrt{\pi}e^{\omega t}} + \frac{1}{2}DV(t) = 0$$

This one is used to check the calculations, solving it for the object DV(t). Therefore DV(t) must fit with the one obtained directly from the scalar field expression above, after eliminating the time.

Final step : expressing the potential and DV(t) in terms of the scalar field, again after a boring sequence of **subs** and **simplify** commands :

$$V(\phi(t)) = \frac{3\omega}{4\pi} + 2\omega^2(\phi(t) - \phi_0)^2 \qquad DV(\phi(t)) = 4\omega^2(\phi(t) - \phi_0)$$

We need the last expression for DV for some numerical investigations we done, as initial data for numerical eolving the Einsten equations for the scalar field. But this is for the moment out of the goals of this lecture.

This was a simple example, just to ilustrate how the things are working with Cosmo library and the reverse engineering method.

Other exmples, more complicated, were processed, even with two scalar fields included to simulate different components of the dark-matter and energy - see in (Vulcanov, 2008).

## Acknowledgements:

Manny thanks to prof.dr. Goran Djordjevic and to the Southeastern European Network in Mathematical and Theoretical Physics - SEENET-MTP for the kind invitation and support.

## References

Grabmeier, J., Kaltofen, E., Weispfennig, U. (eds.) : 2001, *Handbook of Computer Algebra*, Springer-Verlag, Berlin.

Ellis, G. F. R., Madsen, M. S. : 1991, *Classical and Quantum Gravity*, **8**, 667.

Ellis, G. F. R. et. al. : 2004, *Classical and Quantum Gravity*, **21**, 233.

Hehl, F. W., Puntigam, R. A., Ruder, H. (eds.) : 1996, *Relativity and Scientific Computing*, Springer-Verlag, Berlin.

Islam, J. N. : 1992 *An Introduction to Mathematical Cosmology*, Cambridge University Press, Cambridge.

*Maple User Manual*, Maplesoft, Waterloo, Maple Inc., 2005

Perlmutter, S. et. al. : 1999, *Science*, **284**, 1481.

Vulcanov, D. N. : 2001, *On the use of algebraic programming in the general relativity*, **gr-qc/0010085**.

Vulcanov, D. N. : 2003, *Computer Physics Communications*, **154**, 205.

Vulcanov, D. N., Vulcanov, V. N. : 2004, *Analele Univ. din Timisoara*, **XLII**, fasc, special, Seria Matematica-Informatica, 181.

Vulcanov, D. N. : 2008, *Central Europeean Journal of Physics*, **6(1)**, 84.