Computer methods in general relativity: algebraic computing

Marcelo E Araujo, Departamento de Matemática, Universidade de Brasília, 70919 Brasília DF, BRAZIL, E-mail: meda@v1.astro.cf.ac.uk.

Tevian Dray, Department of Mathematics, Oregon State University, Corvallis, OR 97331, USA, E-mail: tevian@math.orst.edu.

James E F Skea, School of Mathematical Sciences, Queen Mary and Westfield College, London, E1 4NS, UK, E-mail: jimsk@cbpfsu1.cat.cbpf.br.

Andreas Koutras, School of Mathematical Sciences, Queen Mary and Westfield College, LONDON, E1 4NS, UK, E-mail: andreas@maths.qmw.ac.uk.

Andrzej Krasinski, Polish Academy of Sciences, Bartycka 18, 00 716 Warszawa, Poland, E-mail: akr@camk.edu.pl.

David Hobill, Department of Physics and Astronomy, University of Calgary, Calgary, Alberta T2N 1N4, Canada, E-mail: hobill@acs.ucalgary.edu.

R G McLenaghan, University of Waterloo, Ontario N2L 3G1, Canada, E-mail: rgmclena@daisy.waterloo.edu.

Steven M Christensen, MathSolutions, Inc., P.O. Box 16175, Chapel Hill, NC 27516 USA, Email: steve@physics.unc.edu.

• The first presentation in this session was **Finding isometry groups in theory and practice** by Araujo, Dray, and Skea. They summarized their work as follows:

Karlhede & MacCallum [1] gave a procedure for determining the Lie algebra of the isometry group of an arbitrary pseudo-Riemannian manifold, which they intended to implement using the symbolic manipulation package SHEEP but never did. We have recently finished making this procedure explicit by giving an algorithm suitable for implementation on a computer [2]. Specifically, we have written an algorithm for determining the isometry group of a spacetime (in four dimensions), and partially implemented this algorithm using the symbolic manipulation package CLASSI, which is an extension of SHEEP.

Our procedure is the following: Apply the classification algorithm built into CLASSI to determine the isotropy group and the set $\{J^\mu\}$ consisting of the functionally independent quantities in the frame components of the curvature tensor of M and its covariant derivatives. By construction, the J^μ are constant on each orbit of the isometry group. Let $\hat{\omega}^i$ be the frame in standard form produced by CLASSI as being appropriate to the isotropy, and let ω^i be the corresponding frame on the isotropy bundle $I(M)$ (the frame bundle restricted to the isotropy group). Calculate the connection 1-forms $\omega^i{}_j$, and note that $\{\omega^i, \omega^j{}_k\}$ is a basis for the cotangent space of $I(M)$. From this point onwards, there are at least three possible approaches which will produce a basis for the orbits of the isometry group in the frame bundle, namely

1) Find the dual vectors, take the subbasis orthogonal to the dJ^μ (which are therefore a basis in the orbit), and take their commutators (see MacCallum & Skea [3]). Though conceptually the most intuitive method, it has the disadvantage, for a computer

Summary compiled by Steven M. Christensen, Session Chair

algebra system, of involving the calculation of a determinant to obtain the vectors dual to the basis one-forms, and this is not necessary as we shall see.

2) The "functional independence" of $\{J^\mu\}$ means that the 1-forms dJ^μ are independent on M (and hence also on $I(M)$); extend them to a basis on M by adding as many ω^i as possible and then extend this to a basis on $I(M)$ by adding the independent $\omega^i{}_j$. Call this basis $\{\sigma^\alpha\}$, and take its exterior derivative. We can now further restrict ourselves to a particular orbit of the isometry group in M by solving the equations $dJ^\mu = 0$, then substituting appropriately on both sides of the remaining equations obtained by exterior differentiation. We thus obtain a basis of the cotangent space of the restriction of $I(M)$ to an isometry group orbit in M. This method has the disadvantage that the calculation of the structure constants is done from a basis which spans a space larger than that of the orbits of the isometry group, and so requires more computation than necessary.

3) use the relations $dJ^\mu = 0$ to rewrite the remaining 1-forms in terms of a reduced set of variables and calculate their exterior derivatives. This method has the disadvantage that dependencies on the reduced set of variables need to be calculated.

The method adopted in the examples presented in [2] is method (3), though which of the methods will eventually be adopted in CLASSI remains to be seen. In any case, since the $\{\sigma^\alpha\}$ form a basis, their exterior algebra closes. But from the Cartan structure equations, the structure constants involve only ± 1, the Riemann tensor of (M, g), and its covariant derivatives, whereas by the construction of the isotropy group given by Karlhede & MacCallum [1], the Riemann tensor and all of its covariant derivatives are constant on each orbit. Thus, the structure constants are constant on each orbit. As pointed out by Karlhede & MacCallum, they are therefore precisely the structure constants of the Lie algebra of the isometry group.

Most of our algorithm has been implemented using CLASSI although the restriction of the 1-forms to a basis in the orbit has not yet been fully automated and thus needs to be done by hand. Automating this part of the algorithm is a reasonably straightforward programming task which when completed will mean that the structure constants for the isometry Lie algebra can be automatically determined from any metric for which CLASSI can compute the isotropy group.

• The second paper was **An algorithm for determining whether a space-time admits a homothety** given by Andreas Koutras and James E. F. Skea who said:

As part of the equivalence problem, and as a problem in its own right, it is useful to have an algorithm which decides whether or not a spacetime admits a homothety. For such a spacetime there exists a vector field x, such that any geometrical quantity q obeys the relation

$$\underset{x}{\pounds} q = cq$$

for some constant c (the weight of q).

To do so, we use a result of Defrise-Carter (later refined by Hall) that a spacetime which admits a homothety group H_n is conformal to a spacetime which admits an isometry group G_n, together with the scaling properties of geometrical quantities in a homothetic spacetime.

¿From CLASSI we can determine a canonical set, $\{R^\mu\}$, of frame components of (the spin decomposition of) the Riemann tensor and its covariant derivatives to an order k which completely determines the local character of the spacetime. As a byproduct, CLASSI gives us the dimension of the isometry group.

We construct a set $\{S^\mu\}$ of the independent ratios of members of $\{R^\mu\}$ such that each member of $\{S^\mu\}$ has zero weight. For a homothetic spacetime there exists a (pseudocanonical) basis related to the canonical basis for $\{R^\mu\}$ by a pure boost in which the number of functionally independent functions of the coordinates in $\{S^\mu\}$ is one less than that in $\{R^\mu\}$.

The problem is now reduced to the determination of the pseudocanonical basis of a homothetic spacetime. This problem parallels the determination of the canonical basis, requiring an analysis of all the possible isotropy groups which can arise at each stage of derivation of the Riemann tensor, and has been solved for all Weyl spinors and physical Segre types.

• Andrzej Krasinski presented **The program Orthocartan**:

The program Ortocartan is written in Lisp and designed for automatic calculation of curvature tensors (Riemann, Ricci, Einstein and Weyl) from a given metric tensor. The input data are the components of an orthonormal tetrad of exterior forms representing the metric, the output are the tetrad components (and, on request, also the coordinate components) of all the quantities calculated at intermediate stages. An "abacus" program Calculate is provided together with Ortocartan for performing simple algebraic operations on expressions defined by the user. The program is presently available on the Atari Mega STE computers, and is being implemented on an iPSC-SYM1 parallel computer by M. Perkowski and his student. Unlike most other programs, Ortocartan was never extended into an elaborate system allowing many kinds of specialized procedures to be carried out automatically (like e.g. determining the Petrov type of a metric or the Segre type of its energy-momentum tensor, or calculation in other tetrads). However, much effort was invested into making its algorithms efficient (in the sense of speed and core-economy), general and fail-safe. By this we mean that the user will not be forced to use the facility of substitutions to correct obvious but nasty little failures. This is the list of advantages that the creators of the program think the users may find attractive:

1. Rational powers of rational numbers are handled automatically, producing the result in the standardized form: either an integer number or an integer base with the exponent having the numerator equal to one.

2. Substitutions can automatically replace sub-sums and sub-products of sums and products, respectively.

3. The chain rule for differentiating composite functions is automatically applied to the maximal depth necessary in a given case.

4. The algebraic simplification is automatically called on the result of each single user-defined substitution.

5. Substitutions by pattern-matching are directly available to the user.

6. The printing procedure will correctly handle nested exponentials to an arbitrary height.

The program, together with the examples and the user manual, is available on a single diskette. A general user-oriented description by A. Krasinski is now in press in the Gen. Rel. Grav. journal, and it contains references to more literature about Ortocartan.

• David Hobill presented **Symbolic computing for numerical relativity**:

In its attempts to model realistic physical systems whose behavior is governed by the Einstein equations, numerical relativity must employ very general expressions for the spacetime metric. This makes the derivation of the Einstein equations and their

subsequent translation into high level computer languages difficult due mainly to the large number of terms that must be manipulated. Computer algebra software packages have long been used to provide relief from the tedium associated with keeping track of the hundreds to thousands of terms associated with analytic calculations in general relativity theory and now it is being used to pre- and post-process numerical codes that model black hole dynamics, stellar collapse and gravitational wave generation.

For the most part numerical relativity uses variables that are quite different from the variables used in most symbolic tensor packages for general relativity. Therefore a new relativity package has been written that produces the ADM or 3+1 form of the Einstein equations in order that a numerical evolution of Cauchy data may be performed. The package presently sits on top of MACSYMA and accepts as input the lapse function, shift vector, three-metric, and extrinsic curvature components to be used in a numerical code. In addition the dependency of these objects upon the spacetime coordinates needs to be specified, but only implicitly. The package then calculates the connection coefficients, Ricci tensor components and the scalar curvature associated with the t=const. spacelike slices. The evolution equations for the three-metric and extrinsic curvature components as well as the scalar and vector constraint equations are then calculated as are other expressions that are important for 3+1 numerical relativity (e.g. the trace of the extrinsic curvature, the trace-free part of the extrinsic curvature, etc.).

As numerical simulations progress to two and three spatial dimensions and as one experiments with different coordinate and gauge conditions, the need to quickly derive new forms of the 3+1 equations becomes evident. In addition these equations must be transcribed into a high level computer language used by the numerical code. In order to avoid typographic and transcription errors another package has been written that receives the output from the 3+1 equation generation package, allows the modeller to replace the partial derivatives appearing in the Einstein equations with Fortran expressions, and then breaks a complicated expression into simple expressions that can be summed over in a Fortran DO loop. The code produced is readable to both humans and machines in order to facilitate the eventual debugging process.

While these examples demonstrated the use of symbolic manipulation for the pre-processing of numerical codes, examples of post-processing were provided. In particular it was shown how a Newman-Penrose formalism packages written specifically for 3+1 variables was used to determine the spin coefficients and Weyl curvatures in terms of hypersurface information. This package also was written to run on top of MACSYMA and interfaces with the 3+1 equation generator.

Examples of the use of Mathematica for producing graphical images from numerical data and for fitting the hole quasi-normal mode parameters to the extracted wave forms from numerical simulations were also demonstrated.

• R.G. McLenaghan gave the fifth talk, **General relativity calculations in Maple**:

A collection of packages and procedures for performing calculations in general relativity and differential geometry that have been written in the Maple computer algebra system was described. Maple is an interactive system for symbolic mathematical computation currently supported by the following operating systems: UNIX and various UNIX-like systems, 386 DOS, Macintosh Finder, DEC VMS, IBM VM/CMS, NeXT and Amiga DOS.

Packages and/or procedures have been written to perform the following tasks: (i)

computation of the connection and curvature, in the natural basis, in a general moving frame and in a complex null frame; (ii) determination of the Petrov type of the Weyl tensor and the Segré type of the Ricci tensor; (iii) computer aided integration of the field equations both in the Newman-Penrose (NP) formalism and in the Ellis-MacCallum formalism. The NP package has been extended [4] to include the transformation of spinor equations into NP form. (iv) calculations with differential forms; (v) determination of the Bianchi-type of a three-dimensional Lie algebra for a given set of structure constants. Most of these packages and procedures are contained in the library of the currently distributed version of Maple (Maple V). Descriptions of these packages and procedures are given in [5] and [6]. A Maple version of the muTENSOR system of Harper and Dyer [7] is also available.

The power of the NPspinor package was illustrated by a description of its use to obtain the following new results in the theory of second order differential invariants of the metric tensor [8]. The spinor equivalent of the Géhéniau-Debever (GD) [9] invariants $D_{(5)}$ and $\dot{D}_{(5)}$ contains the spinor expression $m_6 := \Psi_A{}^B{}_{KL}\Psi_B{}^C{}_{MN}\Phi^{KL}{}_{\dot{A}\dot{B}}\Phi^{MN}{}_{\dot{B}}{}^{\dot{C}}\Phi_C{}^A{}_{\dot{C}}{}^{\dot{A}}$. By means of the NPspinor functions *contract* and *dyad* and the Maple commands *expand* and *factor* it may be shown that $m_6 = (1/6)Ir_2$, where $I := \Psi_{ABCD}\Psi^{ABCD}$, is the complex quadratic Weyl invariant and $r_2 := \Phi_{AB\dot{A}\dot{B}}\Phi^B{}_C{}^{\dot{B}}{}_{\dot{C}}\Phi^{CA\dot{C}\dot{A}}$, is the cubic Ricci invariant. It follows that the GD invariants $C^{(1)}_{(2)}$, $E_{(3)}$, $D_{(5)}$ and $\dot{D}_{(5)}$ are related by the algebraic equation $12(D_{(5)} - \dot{D}_{(5)}) + C^{(1)}_{(2)}E_{(3)} = 0$. It may thus be concluded that their set of fourteen invariants contains at most thirteen independent invariants. The package was subsequently used to obtain a new complex invariant of fifth degree which when adjoined to the GH set yields a set that contains complete minimal sets in the Einstein-Maxwell and perfect fluid cases. The package has also been used to obtain some new results on Huygens' principle [10].

• The final talk was **A practical application of MathTensor** by Steven M. Christensen:

MathTensor [11] is a *Mathematica*-based [12] system with over 250 functions and objects added into *Mathematica* for the purpose of doing both abstract and concrete tensor analysis. One of the earliest reasons for producing *MathTensor* was to perform coincidence limit computations in quantum field theory in curved spacetimes. These calculations involve tensor equations with thousands and ultimately millions of terms or more. The answers are in terms of complicated products of Riemann tensors and their derivatives or contractions.

The basic object we study first is the bi-scalar of geodetic interval, $\sigma(x, x')$, which measures the square of the distance along a geodesic between to spacetime points. The defining equations for σ are:

$$\sigma - \frac{1}{2}\sigma_{;\rho}\sigma_{;}{}^{\rho} = 0, \lim_{x'\to x}\sigma_{;\alpha} = 0, \lim_{x'\to x}\sigma_{;\alpha\beta} = g_{\alpha\beta}.$$

In some applications, we need to determine the coincidence limit of as many as twelve derivatives of σ. Some test computations show that this might involve equations with more than a billion terms! Clearly, a computer program will be needed to generate the terms and then simplify and combine them into a more usable form.

With *MathTensor* it is very easy to set up the equations above and automate taking the derivatives and then substituting in lower covariant derivative coincidence

limits. The only problem that remains is that the products of Riemann tensors that appear may not be independent of each other. Using a program now under development called Schur, Lie group analysis tools like Young Tableau can be used to determine the number and structure of a linearly independent set of Riemann tensor invariants so that the coincidence limits can be put into a canonical form.

Once the computation of the σ coincidence limits is done, then these results can be put into the Schwinger-DeWitt recursion relation formula so that the coincidence limits of the famous Hamidew coefficients can be computed. These are then used in things like the computation of stress tensors, anomalies, point-splitting algorithms, divergences, and so forth. All this can be done with *MathTensor*.

A brief discussion of the functionality of *MathTensor* in general relativity, mathematics, and engineering was also given.

References

[1] Karlhede A and MacCallum M A H 1982 *Gen. Rel. Grav.* **14** 673

[2] Araujo M E, Dray T and Skea J E F 1992 *Gen. Rel. Grav.* **24** 477

[3] MacCallum M A H and Skea J E F 1992 *SHEEP: A Computer Algebra System for General Relativity*, in *Algebraic Computing in General Relativity, Lecture Notes from the 1st Brazilian School on Computer Algebra* Roque W L and Rebouças M J Eds. Oxford University Press to appear

[4] Czapor S R, McLenaghan R G and Carminati J 1992 *Gen. Rel. Grav.* in press

[5] McLenaghan R G 1992 *Maple applications to general relativity* in *Algebraic computing in general relativity, Lecture notes from the 1st Brazilian School on Computer Algebra* **2** Roque W L and Rebouças M J Eds. Oxford University Press to appear

[6] Char B W, Geddes K O, Gonnet G H, Leong B L, Monagan M B and Watt S W 1991 *Maple V Library Reference Manual* (Springer-Verlag, New York)

[7] Harper J F and Dyer C C 1985 *The muTENSOR User Manual*, Department of Astronomy and David Dunlap Observatory, University of Toronto, Toronto, Ontario, Canada

[8] Carminati J and McLenaghan R G 1991 *J. Math. Phys.* **32** 3135; Report on Workshop A1 in these proceedings

[9] Debever R 1964 *Cah. Phys.* **168-169** 303

[10] Carminati J, Czapor S R, McLenaghan R G and Williams G C 1991 *Ann. Inst. Henri Poincaré* - Phys. théor. **54** 9

[11] Parker L and Christensen S M 1991 *MathTensor: A System for Doing Tensor Analysis by Computer*, MathSolutions, Inc., P.O. Box 16175, Chapel Hill, NC 27516, USA

[12] Wolfram S 1991 *Mathematica: A System for Doing Mathematics by Computer*, Wolfram Research, Inc., Champaign, IL 61820, USA