Algebraic Computing in Torsion
Theories of Gravitation

by

Joel B. Fonseca-Neto, M.J. Rebouças
and M.A.H. MacCallum

Rio de Janeiro
1993
NOTAS DE FÍSICA é uma pré-publicação de trabalho original em Física.

NOTAS DE FÍSICA is a preprint of original works unpublished in Physics.

Pedidos de cópias desta publicação devem ser enviados aos autores ou à:

Requests for copies of these reports should be addressed to:

Centro Brasileiro de Pesquisas Físicas
Área de Publicações
Rua Dr. Xavier Sigaud, 150 - 49 andar
22.290 - Rio de Janeiro, RJ
BRASIL
CBPF-NF-016/93

Algebraic Computing in Torsion
Theories of Gravitation

by

Joel B. Fonseca-Neto*, M.J. Rebouças†
and M.A.H. MacCallum‡

†Centro Brasileiro de Pesquisas Físicas — CBPF/CNPq
Rua Dr. Xavier Sigaud, 150
22290-180 – Rio de Janeiro, RJ – Brasil

*Departamento de Física
Universidade Federal da Paraíba
Caixa Postal 5008
58059-900 – João Pessoa, PB – Brasil

‡ School of Mathematical Sciences
Queen Mary and Westfield College
Mile End Road, London E1 4NS – U.K

* bitnet: censf@cbpf.br
† bitnet: reboucas@brlncc
‡ internet: M.A.H.MacCallum@maths.qmw.ac.uk
Algebraic Computing in Torsion Theories of Gravitation

IMACS Symposium SC-1993 INVITED TALK.

Joel B. Fonseca-Neto*, M. J. Rebouças† and M. A. H. MacCallum‡

* CP 5008 Departamento de Física, Universidade Federal da Paraíba, 58059-900 João Pessoa-PB, Brazil
† Centro Brasileiro de Pesquisas Físicas, Rua Dr. Xavier Sigaud, 150, 22290-180 Rio de Janeiro-RJ, Brazil
‡ School of Mathematical Sciences, Queen Mary and Westfield College, Mile End Road, London E1 4NS, U.K.

June 10, 1993

Abstract: A suite of computer algebra programs for tensor and spinor calculations in torsion theories of gravitation is presented. Modules for the algebraic classification of curvature tensors in Riemann-Cartan space-times are discussed. The basic theoretical results and an algorithm for the equivalence problem for Riemann-Cartan space-times are presented. The package TCLASSI which implements this algorithm is also discussed.

Keywords: Computer Algebra, Gravitation, Torsion Theories, Equivalence Problem.

1 Introduction

It seems indisputable that most theoretical physicists spend a large amount of their time in carrying out routine nonnumeric calculations of one sort or another, most of which are algorithmic or semi-algorithmic. Perhaps there is no field in which this is more obvious than that of general relativity (GR).

In GR the underlying manifold $M$ where the calculations are made is that of a four-dimensional Riemannian manifold, i.e., a space-time manifold endowed with a Lorentzian metric and a metric-compatible symmetric connection (Christoffel's symbols $\Gamma^a_{bc}$).

However, it is well known that the metric tensor and the connection can be introduced as independent structures on a given space-time manifold $M$. In GR there is a unique torsion-free connection on $M$. In the framework of torsion theories of gravitation (TTG) we have space-time manifolds endowed with Lorentzian metrics and metric-compatible non-symmetric connections $\Gamma^a_{bc}$ (Riemann-Cartan manifolds). Therefore, in TTG the connection has a metric-independent part given by the torsion, and for a characterization of the local gravitational field one has to deal with both metric and connection.

All the main general-purpose computer algebra systems have some sort of facilities for calculation in general relativity. The most extensive set of programs useful in GR are available with REDUCE, MAPLE, MACSYMA, and with MATHEMATICA through the MATHTENSOR package. Nevertheless, there is room for specialized systems like SHEEP/CLASSI. The major reason for this is that they are more efficient for GR calculations than general-purpose systems. For a comparison of CPU times for a specific metric see MacCallum [1].

As far as we are aware the existing facilities in computer algebra systems for calculations in TTG are quite limited. Actually, we only know of some REDUCE programs for applications to Poincaré gauge field theory [2]. They are written using the REDUCE package EXCALC and basically aim at field equations of TTG [3].

The arbitrariness in the choice of coordinates is a commonly made basic assumption in GR and in TTG. Nevertheless, in these theories it gives rise to the problem of deciding whether or not two appar-
ently different space-time solutions of the field equations are locally the same — the equivalence problem. However, while in GR equivalence means local isometry of two Riemannian space-time manifolds, in TTG besides isometry \((g_{ab} = \tilde{g}_{ab})\) it means affine colineation \((\Gamma^c_{ab} = \tilde{\Gamma}^c_{ab})\) of two Riemann-Cartan space-time manifolds [1].

The LISP-based system SHEEP was partially devised with the equivalence problem for Riemannian manifolds in mind. Frick [5] designed the system and Karbœde [6] and Amann [7] developed the algorithm and the first set of programs required for checking the equivalence of vacuum solutions [8, 9]. Since then, algorithms and programs to extend the treatment to nonvacuum case have been developed by MacCallum, Joly, Amman, and others. They are described in references [1] and [10]–[12]. See also [11] and references therein.

The equivalence problem for Riemann-Cartan space-time manifolds was taken up in 1989 as part of a collaboration between the group at the School of Mathematical Sciences, QMW, led by Prof. M. A. H. MacCallum, and Dr. M. J. Rebouças’s group at the Brazilian Center for Physics Research (CBPF) in Rio. The problem was tackled bearing in mind three basic aspects. First, the mathematical problem of finding the necessary and sufficient conditions for equivalence. Second, the problem of finding practical algorithms for effectively carrying out the calculation. Third, the implementation of the algorithms in the SHEEP/CASSI computer algebra system.

With the equivalence problem in TTG in view, we have developed a preliminary working version of the basic modules, which form the package called TCLASS1, extending the facilities of SHEEP/CASS1 to the Riemann-Cartan space-time manifolds of TTG. This is the first report of our work on computer-aided methods for torsion theories of gravitation.

Before actually tackling the equivalence problem, we had to implement several mathematical techniques which are used in the algorithm for the resolution of the equivalence problem. First, we have extended the tensor formalism to deal with calculations in coordinates, and noncoordinate frames in which the metric has constant components. Second, we have implemented the spinor formalism for Riemann-Cartan space-time manifolds. Finally, we have considered the Petrov and Segre classifications, and the algebraic classifications of vectors and bivectors.

In the following sections we will use the Einstein summation convention, square brackets for antisymmetrization and round brackets for symmetrization. Small capital letters are used throughout this work for the names of the modules, whereas for the internal names of the tensors we use upper-case letters. To indicate that a specific module of our package TCLASS1 is an extension of a SHEEP/CASS1 module we use the letter \(T\) followed by the corresponding SHEEP/CASS1 name. Thus, for example, our TCLASS module \(TCORD\) extends the facilities of the module \(CORD\) of SHEEP/CASS1 to the Riemann-Cartan space-time manifolds of TTG.

## 2 Implementation of Mathematical Techniques

### 2.1 Tensor Formalism

Although calculations using frames in which the metric has constant components are usually simpler than using coordinates, sometimes one might need to perform coordinate calculations. So, for the sake of completeness, we have developed programs not only for frame but also for coordinate calculations in TTG.

The names of the tensors externally (for the user) are the same, regardless of whether the calculation is performed using coordinates or a frame. However, our package uses the appropriate internal names pointed at by tensor aliases automatically set up. The torsion tensor, for example, is referred to as \(TOR\) for the user, but internally is denoted as either \(TORC\) or \(TORF\) depending on whether it is being calculated at by tensor aliases automatically set up. The torsion part of the curvature tensor we shall call the Cartan-Cartan part of the curvature (GCCD) and \(T^a_{bc}\) of the torsion (TORC).

Using \(TCORD\) one can calculate the torsion’s irreducible parts, namely the components \(T^a_{bc} = T_{abc}^\gamma\) of the torsion trace (TTRACE), \(P_{bc} = 1/2\eta_{abc} T^{\gamma}_{abc}\) of the torsion pseudo-trace (PTORC) and \(S_{cd} = T_{\gamma,cd} - 2/3 \delta_{\gamma}^{\nu} T_{\nu,d} + 1/3 \eta^{\nu}_{cd} P_{\nu}\) of the torsion part with no trace and no pseudo-trace (TORC). Here \(\eta_{abcd}\) is the totally antisymmetric pseudo-tensor with \(\eta_{0123} = 1\).

One can use \(TCORD\) to calculate the components of the contorsion (CONTORC), the connection (TGAMC) and the curvature (TRIEC). They are given by

\[
R_{abcd} = (T_{abcd} + T_{dac} - T_{adb})/2, \quad \Gamma_{cd} = \Gamma_{[cd]} + \eta^{\nu}_{cd},
\]

and

\[
R_{abcd} = 2 g_{ad}(\Gamma_{[cd]} + \Gamma_{[cd]}^\nu \Gamma_{\nu[bd]}),
\]

where \{1, 2\} are the Christoffel symbols (GAMC) coded in \(CORD\).

Using \(TCORD\) one can also calculate the components: \(\Lambda_{ab} = R_{abcd}^{\nu} T_{\nu,d}^{\nu}\) of its anti-symmetric part (ATRICC), \(S_{ab} = R_{abcd} - 1/4 g_{ab} R\) of its symmetric trace-free part (STRICC) and the curvature scalar \(R = R_{\nu}^{\nu}\) (TRSCC).

The module \(TWIELC\) can be used to calculate the part of the curvature tensor we shall call the Cartan-Weyl-tensor (CWEYLC), whose components are

\[
C_{abcd} = R_{abcd} - R_{[a}g_{b]d} + \frac{1}{3} R g_{[a}g_{b]} g_{c]}.
\]
It is well known that $C_{abcd}$ is trace-free. Nevertheless, in Riemann-Cartan spaces times one can calculate its pseudo-trace (PTTRIC), whose components are $D_{ab} = 1/2 \eta_{abcd} R^{cd}$. The trace (PTSCLC) and the symmetric trace-free part (STRICC) of $D_{ab}$, namely $D = D_1$ and $S_{ab} = D_{ab} - 1/2 \eta_{ab} D$, can also be obtained. We call $D$ the pseudo-scalar of curvature. The programs for calculating $D_{ab}$ and its irreducible parts are in the module TCORD.

Finally, the module TWYLT can be used to calculate what we call the Weyl tensor (TWYLT), which is trace-and-pseudo-trace-free, and whose components are $W_{abcd} = C_{abcd} - \eta_{abcd} D_{ab} + 1/6 g_{abcd} D$. As far as the frame calculations are concerned we have developed the modules ITORSION, TORSION and TWYLT. Besides the components IZUDF of the frame 1-forms $\xi^a = 2 A^a$, the user's input for these modules is the input frame (IFRAME) and the components of either the torsion (ITORF) or the irreducible parts of torsion, internally denoted by IT-TORDF (torsion trace), IPTORDF (torsion pseudo-trace) and ILTORDF (torsion trace-and-pseudo-trace free). In these internal names the letter 'T' stands for input. Whenever the torsion is given its irreducible parts can be calculated and conversely. The programs for the torsion input are in the module ITORSION and for the frame input are in the SHEEP/CLASSI module IZUD, which is automatically loaded by ITORSION.

The module TORI can be used to calculate the same set of tensors one can find through the module TCORD, namely TORF, TTORDF, PTORDF, ILTORDF, TGMF, TRIF, TRICF, PTRICF, STRICF, TORSICF and TPRSCLF. Similarly, TWYLT can give the frame components of CWEYL and TWYLT.

It should be mentioned that the frame calculations are not performed with respect to IFRAME, but to another frame internally denoted as FRAME. They coincide by default, but can be different. The module ITORSION provides facilities to perform the transformation of the torsion from IFRAME to FRAME automatically.

Finally, we mention that the possible choices for IFRAME and FRAME are: Lorentz (LORENTZ), Cartesian (CARTESIAN), null (NULLT), half-null (HNULL), null cartesian (EUNULL) and half-cartesian (EUNULL).

2.2 Spinor Formalism

The system SHEEP/CLASSI has a package for spinor manipulation in GR which was written basically to take advantage of the simplification achieved in the practical algorithm for the equivalence problem.

In order to extend SHEEP/CLASSI facilities for spinor calculations to TTG we have developed the modules TSPINOR, TPSIPHI and SPTCURV. One can use TSPINOR to calculate the spinor components $\Gamma_{ABCD}$ of the spinorial connection (STGAM) and the spin coefficients [15]. The user's input is a null torsion frame and the components of the torsion tensor. The SHEEP/CLASSI module SPINOR is automatically loaded by TSPINOR in order to change from null frame to spinor components.

The module TPSIPHI can be used to calculate the irreducible parts of the torsion and the curvature spinors. The user's input is either a Lorentz or a null tetrad and the frame components of the torsion.

The curvature spinor (STPCURV) can be decomposed into irreducible parts according to [15]

\[
R_{ABCD} = \epsilon_{GHI} [\Psi_{ABCD} + \epsilon_{ACBD} (\Lambda + i\Omega)] + \epsilon_{ACBD} (\Lambda + i\Omega) + \epsilon_{ADBC} (\Lambda + i\Omega),
\]

(2.2)

The module TPSIPHI can be used to calculate the components of $\Psi_{ABCD} = \Phi_{ABCD}$ (TPSII), $\Phi_{ABCD} = \Phi_{ABCD} (TPII)$, $\Sigma_{AB} = \Sigma_{AB} (SIGMA)$, $\Theta_{ABCD} = \Theta_{ABCD} (THETA)$, $\Lambda (TLAMBDA)$ and $\Omega (OMEGA)$. Hereafter we shall call $\Psi_{ABCD}$ and $\Phi_{ABCD}$ the Weyl and symmetric Ricci spinors, respectively.

The torsion spinor, on the other hand, can be decomposed into its irreducible parts as follows

\[
T_{ABC} = L_{ABC} + \frac{1}{2} (\epsilon_{ABCD} T_{CD} + \epsilon_{ACBD} T_{AB}) + \frac{1}{2} i \epsilon_{ABCD} S_{CD} + \epsilon_{ACBD} S_{AB}.
\]

(2.3)

The module TPSIPHI can also be used to calculate the components $T_{ABC}$ of the torsion trace spinor (SPTO), $\Sigma_{AB}$, of the torsion pseudo-trace spinor (SPTOR) and the components $L_{ABC}$ of the trace-and-pseudo-trace free part of the torsion (SPLTOR).

When using the module TPSIPHI the irreducible parts of the curvature and torsion spinors are calculated from the corresponding tensors. However, they can also be calculated by using another module called SPTCURV when the curvature tensor is not known by the system. The user's input is a null tetrad frame and the components of the torsion. The module TSPINOR is automatically loaded by SPTCURV.

The SPTCURV algorithm uses the curvature spinor $R_{ABCD} = 0$, which is calculated from the spinorial connection (STGAM), and the torsion spinor $T_{ABC}$, implemented in TSPINOR.

Dyad transformations are also available to reduce the spinors to their canonical forms. The package TCLASSI implements two different ways of making dyad transformations of spinors: (a) when TPSIPHI is used the dyad transformation DVTR is automatically applied when IFRAME is transformed to FRAME, (b) when SPTCURV is used there is a different dyad transformation DVTRSP which only acts on spinors.
2.3 Algebraic Classification

The essential idea underlying all classifications is the concept of equivalence. Objects may be grouped into different equivalence classes according to the criteria one chooses. A further refinement is to choose exactly one element, as simple as possible, from each equivalence class. The collection of all such simples constitutes a set of canonical forms for the objects one is dealing with. Every object in any equivalence class is then equivalent to exactly one canonical form. In GR an algebraic classification of the curvature tensor can be cast in terms of the existence problems one can construct from its irreducible parts namely the Weyl and Ricci tensors. These eigenvalue problems split the space-times into equivalence classes and give rise to what is known as Petrov and Segre types. In TTT, besides the classifications of objects with the same symmetries as the Weyl and symmetric Ricci spinors, the irreducible parts of the curvature and the torsion give rise to classifications of vectors and bivectors. So, in TTT the space-times can be classified by Petrov and Segre types as well as according to the character of the vectors and the types of bivectors.

To deal with algebraic classifications in TTT we have developed the modules SEGPIETROV, VECTCLA and BIVTCLA.

In the module SEGPIETROV we have implemented the commands PETROV and TSEGRE, which perform the Petrov and Segre classifications and can operate on spinors with the same symmetries as the Weyl and the symmetric Ricci spinors. These functions use the SHEEP/CLASS modules PTPETROV and SEGRE, automatically loaded by SEGPIETROV. These modules implement the algorithms for Petrov and Segre classifications discussed in references [13, 16]. The commands PETROV and TSEGRE can be used to determine the Petrov type of \( \Psi_{ABCD} \) and the Segre types of \( \Phi_{ABXZ} \) and \( \Theta_{ABXZ} \). In modules VECTCLA and BIVTCLA we have implemented the commands VECTCLA and BIVTCLA which classify spinors corresponding to vectors and bivectors. The algorithms implemented in these modules classify the vectors as timelike, spacelike or null, and the bivectors as null (radiation) or non-null (general). The commands VECTCLA and BIVTCLA can be used to determine the types of \( T_{AX} \) and \( S_{AX} \), and of \( \Sigma_{AX} \). 

3 The Equivalence Problem

3.1 Basic Theoretical Results

The most appropriate setting to deal with the equivalence problem, according to Cartan [20], is that of a base manifold \( M \) together with all possible choices of orthonormal frames. This structure also constitutes a differential manifold and is known as the Lorentz frame bundle \( L(M) \). The manifold \( L(M) \) incorporates the freedom in the choice of Lorentz frames and has a uniquely-defined set of linearly independent 1-forms \( \{ \theta^A, \omega^A_B \} \) forming a basis for the cotangent space \( T^*_p(L(M)) \) at an arbitrary point \( P \in L(M) \).

We can determine equivalence of Riemann-Cartan manifolds by considering that if two Riemann-Cartan manifolds \( M \) and \( \tilde{M} \) are the same then so are their Lorentz frame bundles. We, therefore, say that \( M \) and \( \tilde{M} \) are locally equivalent when there exists a local mapping \( F \) of \( L(M) \) onto \( L(\tilde{M}) \) such that

\[
F^*\theta^A = \theta^A \quad \text{and} \quad F^*\omega^A_B = \omega^A_B. \tag{3.4}
\]

Here \( F^* \) is the well known pull-back map.

The equivalence problem for Riemann-Cartan manifolds was recently solved [1] and the results can be summarized by stating that two \( n \)-dimensional Riemann-Cartan manifolds \( M \) and \( \tilde{M} \) are locally equivalent if a local map (diffeomorphism) between their corresponding Lorentz frame bundles \( L(M) \) and \( L(M) \) exists, such that the algebraic equations

\[
T^A_{BC} = T^A_{BC}, \quad T^A_{BC,M} = T^A_{BC,M}, \quad R^A_{BCD} = R^A_{BCD}, \quad R^A_{BCD,M} = R^A_{BCD,M}, \quad T^A_{BC,M,M} = T^A_{BC,M,M}, \quad \ldots
\]

\[
R^A_{BCD,M_1M_2 \ldots} = R^A_{BCD,M_1M_2 \ldots}, \quad T^A_{BC,M_1M_2 \ldots} = T^A_{BC,M_1M_2 \ldots}
\]

are compatible as equations in the coordinates \( \{ x^a, \xi^A \} \) at an arbitrary point in the Lorentz frame bundle manifold \( L(M) \) (note that \( x^a \) are coordinates on the base manifold while \( \xi^A \) parametrize the group of allowed frame transformations). Reciprocally, equations (3.6) imply equivalence between the space-time manifolds. The \((p + 2)\textsuperscript{th} \) derivative of torsion and the \((p + 1)\textsuperscript{th} \) derivative of curvature are the lowest derivatives which are functionally dependent on all the previous derivatives.

As no real equations were used to show the conditions (3.6) for equivalence of Riemann-Cartan manifolds, the results hold irrespective of the torsion theory of gravitation one may be concerned with. Thus, a comprehensive local description of a Riemann-Cartan manifold is given by the set of coordinate invariants

\[
I_p = \{ T^A_{BC}, T^A_{BC,N_1}, R^A_{BCD}, R^A_{BCD,N_1}, T^A_{BC,M_1M_2 \ldots}, T^A_{BC,N_1N_2M_1M_2 \ldots} \} \quad \text{or} \quad \{ R^A_{BCD,N_1 \ldots}, T^A_{BC,M_1M_2 \ldots}, R^A_{BCD,N_1 \ldots} \}
\]

\[\text{Actually, the cotangent space can be naturally decomposed as a direct sum of the} n\text{-dimensional horizontal subspace, spanned by the canonical 1-forms} \{ \omega^A \}, \text{and the} \{(q + 1)/2\}\text{-dimensional vertical subspace, spanned by the linear connection 1-forms} \{ \omega^A_{\#} \}, \text{defined on} L(M). \text{ See, for example, S. Kobayashi and K. Nomizu [17].}\]
where the derivatives have to be calculated up to the 10th order at most.

One might think at first sight that, although satisfactory, the conditions for equivalence (3.6) are rather daunting from a practical point of view since the curvature $R_{ABCD}$, the torsion $T^A_{BC}$, and their covariant derivatives are functions on $\mathcal{F}(M)$. However, in the next section we will discuss a practical procedure for testing equivalence in which all calculations are performed on the base manifold $M$.

### 3.2 Practical Algorithm

To deal with equivalence in practice it is necessary to calculate the elements of the set $L_q$. However, even when these calculations are performed on the space-time base manifold $M$, and the Bianchi and Ricci identities and their differential concomitants are taken into account, in the worst case one still has 11064 independent elements to calculate. Thus, a practical procedure for carrying out these calculations and a computer algebra implementation are desirable.

We have developed a practical procedure for testing equivalence of Riemann-Cartan space-times [18], extending Karlhede's [6] results for Riemannian space-times.

In our procedure the maximum order of derivatives is 7 for the curvature and 8 for the torsion [18]. The maximum number of algebraically independent elements is then reduced to 6016. For the sake of comparison we mention that the corresponding number in Karlhede's procedure is 3156.

The basic idea behind our procedure is separate handling of frame rotations and space-time coordinates, fixing the frame at each stage of differentiation of the curvature and torsion tensors by aligning it as far as possible with invariantly-defined directions.

An important point in the procedure for Riemann-Cartan manifolds is that, according to (2.1), the components of the curvature are algebraically related to the components of both the torsion and its first derivative. Similarly, the $q$th derivatives of curvature are algebraically related to the $q$th and $(q + 1)$th derivatives of torsion.

The algorithm starts by setting $q = 0$ and has the following steps:

1. Calculate the set $L_q$, i.e., the derivatives of the curvature up to the $q$th order and of the torsion up to the $(q + 1)$th order.
2. Fix the frame, as much as possible, by putting the elements of $L_q$ into canonical forms.
3. Find the frame freedom given by the residual isotropy group $H_q$ of transformations which leave the canonical forms invariant.
4. Find the number $t_q$ of functionally independent functions of space-time coordinates in the elements of $L_q$, brought into the canonical forms.
5. If the isotropy group $H_q$ is the same as $H_{(q-1)}$ and the number of functionally independent functions $t_q$ is equal to $t_{(q-1)}$, then let $q = p + 1$ and stop. Otherwise, increment $q$ by 1 and go to step 1.

This procedure provides a comprehensive characterization of Riemann-Cartan space-times in terms of the following discrete properties: the set of canonical forms in $L_q$, the isotropy groups $\{H_0, \ldots, H_p\}$ and the number of independent functions $t_0, \ldots, t_p$. To check the equivalence of two Riemann-Cartan space-times one first compares these discrete properties and only when they match is it necessary to determine the compatibility of equations (3.6).

Another important point is that one can explicitly specify a minimal set of algebraically independent quantities from which all elements of the set $L_q$ can be obtained by algebraic operations. Before proceeding to the discussion of this minimal set we remark that rather than using tensors as such, the algorithms and programs for the equivalence problem in TIG were devised and written in terms of spinors.

A complete minimal set for Riemann-Cartan space-time manifolds, recursively defined in terms of totally symmetrized $q$th and $(q + 1)$th derivatives of the curvature and torsion spinors, can be specified [19] by:

1. The torsion's irreducible parts:
   - (a) $T_{AX}$, (b) $S_{AX}$, (c) $L_{ABCD}$
2. The totally symmetrized $q$th derivatives of:
   - (a) $\Lambda$,
   - (ii) $T = \nabla_{NN'} T_{NN'}$,
   - (iii) $S = \nabla_{NN'} S_{NN'}$,
   - (b) $\Sigma_{AB}$,
   - (ii) $T_{AB} = \nabla_{NN'} (A T_{B})_{NN'}$,
   - (iii) $S_{AB} = \nabla_{NN'} (A S_{B})_{NN'}$,
   - (c) $\Phi_{ABCD}$,
   - (ii) $\Phi_{ABCD} = -\nabla_{NN'} (A L_{BCD})_{NN'}$,
   - (d) $\Theta_{ABXZ}$,
   - (ii) $\Theta_{ABXZ}$,
   - (iii) $\Phi_{ABXZ} = \frac{1}{2} (L_{ABXZ} + T_{ABXZ})$,
   - where $L_{ABXZ} = -\nabla_{NN'} (A T_{X} L_{Z})_{AB}$.

3. The totally symmetrized $(q + 1)$th derivatives of:
   - (a) $T_{AX}$, (b) $S_{AX}$, (c) $L_{ABCD}$

4. For $q \geq 1$, the totally symmetrized $(q - 1)$th derivatives of:
   - (a) $\Xi_{ABCD} = \nabla_{NN'} (A \Theta_{B})_{NN'}$,
   - (b) $E_{ABCD} = \nabla_{NN'} (A \Phi_{B})_{NN'}$,
   - (c) $U_{AX} = \frac{1}{2} (\nabla_{NN'} \Sigma_{AX} + \nabla_{NN'} \Sigma_{AX})$,
   - (d) $V_{AX} = -\frac{1}{2} (\nabla_{NN'} \Sigma_{AX} - \nabla_{NN'} \Sigma_{AX})$. 

---

CBF-EN-3184-1
5. For $q \geq 2$:

(a) the d'Alembertian $\nabla^N \nabla_N Q$ applied to all quantities $Q$ calculated for the derivatives of order $q = 2$.

(b) the totally symmetric $(q = 2)$th derivatives of:

i. $\mathcal{T}_{ABCD} = -\nabla^N (\mathcal{E}_{ABCD})$,

ii. $U_{AB} = \nabla^N (\mathcal{U}_{AB})$.

The above complete minimal set is a generalization of the corresponding set found by MacCallum and Åman [2] for Riemannian space-time manifolds.

### 3.3 First Implementation: The Package TCLASSI

We have developed a suite of computer algebra programs called TCLASSI to carry out the procedure described in the previous section. In the development of the package we have extended SHEEP/CCLASSI algorithms and modules to allow for calculations involving torsion. It would be inappropriate to discuss here the details of the implementation of TCLASSI. So, in what follows we briefly discuss the basic aspects related to the algorithm.

The implementation of step 1 in terms of spinors required the development of the set of programs discussed in section 2.2 and two additional modules, namely TEQUSPI and TSYMSPI. These modules implement the programs for the calculation of the minimal set $I_q$ up to the third and fourth derivatives of the curvature and torsion, respectively.

Step 2 requires the choice of a canonical frame. Its implementation requires modules to perform the algebraic classification of the curvature and the torsion and their derivatives, and programs providing ways to fix the (canonical) frame by bringing the elements of $I_q$ into canonical form. The frame can be fixed interactively by using the module TYDTSYM for dyad transformations of symmetrized spinors. The user's input for this module is the dyad transformation DYTRX. The transformed spinors are denoted by SPTTOITH, TPSITH, and so forth. Once an appropriate dyad transformation is interactively found, it can be used in either DYTR or DYTRSP to bring the spinors into a canonical form.

For step 3, the SHEEP/CCLASSI module ISOTST and CLABAS were extended as the modules ISO3STOR and TCLABAS to include bivectors and spinors with the same symmetries as the Weyl spinor. These latter modules implement the command ISOTST, which determines the isotropy group and verifies whether the frame is standard, generalizing the similar command in ISO3ST. It should be mentioned that ISO3ST loads CLABAS and ISOTST, which in turns loads CLABAS.

A second aspect of the implementation of steps 2 and 3 is that for $q \geq 1$ it works as does CCLASSI, where no algebraic classification is made. It only verifies whether the isotropy group and the number of independent functions obtained for $q = 1$ remain the same as for $q$, by using the SHEEP/CCLASSI command FUTSST.

No addition to the SHEEP/CCLASSI system was required for the steps for $q = 0, 1, 2$ separately. This is done through whether the isotropy group and the elements of the minimal set $I_q$ to determine the number of functionally independent functions.

Finally, the implementation of step 5 required the coding of all possible isotropy groups to allow the necessary comparisons. The coding of these groups is in module CLABAS, and is used in ISOTST to identify the isotropy groups.

The module TCLASSI (not the package) implements the whole practical procedure. This is done through the command TCLASSIFY, which performs the procedure from $q = 0$ to $q = 2$. It also has the commands TCLASSIFY0, TCLASSIFY1 and TCLASSIFY2 to do the steps for $q = 0, 1, 2$ separately.

A summary output of the discrete parts of the classifications can be generated by the command TCLASSUM in module TCLASSUM for comparison of Riemann-Cartan space-times.

### References


