Spinors: a Mathematica package for doing spinor calculus in General Relativity

Alfonso García-Parrado Gómez-Lobo,*, José M. Martín-García

*Centro de Matemática, Universidade do Minho. 4710-057 Braga, Portugal
bLaboratoire Univers et Théories, Observatoire de Paris, CNRS, Univ. Paris Diderot, 5 place Jules Janssen, 92190 Meudon, France
1Institut d’Astrophysique de Paris, Univ. Pierre et Marie Curie, CNRS, 98bis boulevard Arago, 75014 Paris, France

Abstract

The Spinors software is a Mathematica package which implements 2-component spinor calculus as devised by Penrose for General Relativity in dimension 3+1. The Spinors software is part of the xAct system, which is a collection of Mathematica packages to do tensor analysis by computer. In this paper we give a thorough description of Spinors and present practical examples of use.

Keywords: General Relativity; Spinor calculus; Tensor analysis; Tensor computer algebra; PACS: 04.20.-q;02.40.-k; 04.20.Gz.

1. Introduction

The concept of spinor plays an important role in certain areas of mathematical and theoretical physics. Roughly speaking a spinor is a field which transforms under a spinor representation of a given symmetry group in our system. For example, if we are working in a pseudo-Riemannian manifold with a metric of signature (p,q) (p represents the number of +1 and q
the number of −1 entries in the canonical form of the metric), then a natural symmetry group is the group which transforms orthonormal frames into orthonormal frames. This group is \( O(p, q) \) (\( SO(p, q) \) if we restrict ourselves to transformations preserving the frame orientation). The spin group is then the universal covering of \( SO(p, q) \) which, as is well-known, is \( Spin(p, q) \) and hence spinors transform under irreducible representations of this group.

The above considerations are completely general and they enable us to introduce the notion of spinor field in any pseudo-Riemannian manifold admitting a spin structure. However, in the case of a 4-dimensional Lorentzian manifold (the space-time model in General Relativity) a more algebraic approach is desired. This approach was pioneered by Penrose \([1]\) where he studied the main properties of the spinor algebra of those spinors arising from the spin group of \( SO(1, 3) \) and in addition he developed a calculus adapted to the particular spin vector bundle which one can define in a 4-dimensional Lorentzian manifold admitting a spin structure. Penrose’s spinor calculus revealed very useful in certain contexts of General Relativity (GR) where the use of tensor methods results in very cumbersome computations. Perhaps the best known example is the spinor formulation of the algebraic classification of the Weyl tensor (Petrov classification). The spinor form of the Weyl tensor is a totally symmetric 4-rank spinor and it is very easy to show that such a spinor can only admit four different algebraic types which are in correspondence with the four distinct Petrov types.

In this article we describe the Mathematica package Spinors which implements the spinor calculus in four dimensional Lorentzian geometry as conceived by Penrose. In this conception, spinors are tensor fields on a certain tensor bundle and therefore one can use the general ideas of tensor bundles to work with spinors. In particular the notion of spin covariant derivative, the curvature spinors or the relation between spinors and space-time tensors find here a natural formulation. An important part of this formulation is the notion of abstract index used to represent tensor fields on any tensor bundle. This representation of tensor fields has been adopted in the system xAct \([2]\), which Spinors is part of. The system xAct is a system to do tensor analysis by computer in Mathematica, both by working with tensors as linear combination of basis tensors (component calculus) and by working with tensors as symbolic names with certain properties like rank or symmetry (abstract calculus). The system xAct consists of different modules tailored for different tasks and Spinors is one of these modules.

Other computer algebra systems support computations with spinors. For example in the context of Particle Physics we may quote the package \( \text{Spinors@Mathematica} \) \([3]\) which can be used in the evaluation of scattering amplitudes at tree and loop level. The stand-alone package Cadabra \([4]\) handles generic abstract spinor quantities in any dimension, with emphasis in Field Theory, but no special support for General Relativity or component computations. The Maple built-in package DifferentialGeometry has extensive support for component computations of multiple types, in particular the NP formalism, but no support for abstract tensor computations. Another Maple package handling the NP formalism is \( \text{NPSpinor} \) \([5]\).

The paper is organised as follows: in section 2 we give a mathematical introduction to spinor calculus. The aim of this introduction is to set the notation and conventions which are followed by the Spinors implementation. Section 3 explains how the Spinors software fits into the xAct framework and section 4 presents a practical session with Spinors in which the main features of the program are shown by means of practical examples. The paper is finished in section 5 where a practical computation involving the Nester-Witten spinor and the Sparling identity is carried out with Spinors.

2. Mathematical preliminaries

In this section we give an overview of the spinor calculus in General Relativity, following a practical approach to introduce the subject and omitting most of the proofs (detailed studies can be found in e.g. \([3, 7]\)). Let \( \mathbf{L} \) be a 4-dimensional real vector space endowed with a real scalar product \( g, \) of Lorentzian signature and let \( \mathbf{S} \) be a 2-dimensional complex vector space (complex conjugate of scalars will be denoted by an overbar). The vector space \( \mathbf{S} \) is related to another complex vector space \( \overline{\mathbf{S}} \) by an anti-linear, involutive transformation.

The vector space \( \mathbf{L} \) and its dual \( \mathbf{L}^* \) can be used as the starting point to build a tensor algebra in the standard fashion. Similarly a tensor algebra is built from \( \mathbf{S}, \overline{\mathbf{S}} \) and their respective duals \( \mathbf{S}^*, \overline{\mathbf{S}}^* \). We denote these algebras by \( \mathfrak{T}(\mathbf{L}), \mathfrak{T}(\mathbf{S}) \) and \( \mathfrak{T}(\overline{\mathbf{S}}) \) respectively \([3]\). In this work abstract indices will be used throughout to denote tensorial quantities: in this way lowercase Latin indices \( a, b, \ldots \) will denote abstract indices on elements of \( \mathfrak{T}(\mathbf{L}) \).

\( ^{\text{strictly speaking only the algebras } \mathfrak{T}(\mathbf{L}) \text{ of tensors } r\text{-contravariant } s\text{-covariant can be defined (and the same applies to } \mathfrak{T}(\overline{\mathbf{S}})). \text{ To lessen the notation we will suppress the labels } r, s \text{ in the notation and they will only be made explicit when confusion may arise.} \)
and capital Latin indices $A, B, \ldots$ (resp. primed capital Latin indices $A', B', \ldots$) will be used for abstract indices of elements in $\mathfrak{T}(S)$ (resp. $\mathfrak{T}(\bar{S})$). The union of the tensor algebras $\mathfrak{T}(S), \mathfrak{T}(\bar{S})$ will be referred to as the spin algebra and its elements will be called spinors. One can also build tensor algebras by taking tensor products of elements in $\mathfrak{T}(L), \mathfrak{T}(S)$ and $\mathfrak{T}(\bar{S})$. Quantities in these tensor algebras will be referred to as mixed quantities and they will carry abstract indices of tensor and spinor type. All tensor algebras shall be regarded as complex vector spaces.

Since $S$ is 2-dimensional, we deduce that the vector space of antisymmetric 2-spinors is 1-dimensional and therefore we can pick up a non-vanishing representative $\epsilon_{AB}$ which generates such a vector space. We define next a spinor $\bar{\epsilon}^{AB}$ by the relation

$$\epsilon_{AB} \epsilon^{CB} = \delta_A^C, \tag{1}$$

where $\delta_A^C$ is the identity tensor (also known as the Kronecker delta) on the vector space $S$. Indeed the spinors $\epsilon_{AB}, \bar{\epsilon}^{AB}$ can be used to relate elements in $S$ and elements in $S'$ in the following way

$$\xi^A \epsilon_{AB} = \xi_B, \quad \bar{\xi}^A = \epsilon^{AB} \xi_B, \tag{2}$$

where $\xi^A$ is an arbitrary spinor in $S$. Hence, the spinors $\epsilon_{AB}$ and $\bar{\epsilon}^{AB}$ can be understood as a metric on $S$ (symplectic metric) and its inverse and the operation shown in (2) is the standard “raising and lowering” of indices. These operations are extended to the full spinor algebra without difficulty. In particular we can raise the indices of $\epsilon_{AB}$ getting $\epsilon^A = \bar{\epsilon}^{AB}$ and from now on only the symbol $\epsilon$ will be used for the symplectic metric and its inverse. Note also the property

$$\delta^A_B = -\delta^B_A. \tag{3}$$

Here the quantity $\delta^A_B$ is the Kronecker delta on $S$ and $\delta^A_B$ is a derived quantity obtained from it by the raising and lowering of indices. In particular this implies $\delta^A_A = 2$. The spinors $\epsilon_{AB}, \epsilon^{AB}$ and $\delta^A_B$ all have counterparts (complex conjugates) defined in the algebra $\mathfrak{T}(\bar{S})$.

It is possible to relate tensors and spinors by means of the soldering form. This is a mixed quantity $\sigma_{AA'}^{AB}$ fulfilling the algebraic properties

$$\sigma_{AA'}^{AB} \epsilon_{BB'} = \delta_{BA} \bar{\epsilon}_{BB'}, \quad \sigma_{AA'}^{AB} \epsilon^{BB'} = \delta^A_B, \quad \sigma_{AA'}^{AB} = \sigma_{AA'}^{BA'}.$$  

The last of these properties implies that $\sigma_{AA'}$ is hermitian. This is only compatible with the metric signature $(1, -1, -1, -1)$. Choosing $\sigma_{AA'}^{AB}$ anti-hermitian would be only compatible with the signature $(-1, 1, 1, 1)$ [7]. These properties enable us to relate tensors and spinors in the following way

$$T^{AA'}_{B'B''} \epsilon_{B'B''} = T_{B'B''}^{AA'} \epsilon^{B'B''} \epsilon_{B'B''} = T_{B'B''}^{AA'} \epsilon^{B'B''}, \tag{4}$$

$$T^{AB}_{B'B''} = T_{B'B''}^{AB} \epsilon_{B'B''} \epsilon^{B'B''} \epsilon_{B'B''}, \quad T_{B'B''}^{AB} \epsilon^{B'B''} \epsilon_{B'B''} = T_{B'B''}^{AB} \epsilon^{B'B''} \epsilon_{B'B''}, \tag{5}$$

where $T^{AB}_{B'B''}$ is an arbitrary tensor and $T^{AA'}_{B'B''}$ its spinor counterpart.

Another important algebraic property of the soldering form is

$$\sigma_{AA'}^{AB} \sigma_{B'C'}^{AC'} + \sigma_{AA'}^{AC} \sigma_{B'B'}^{A'C'} = g_{ab} \mathfrak{T}_{AC'C'}. \tag{6}$$

This equation is a direct consequence of the irreducible decomposition of the product $\sigma_{AA'}^{AB} \sigma_{B'C'}^{AC'}$ according to theorem [11] below and the algebraic properties of the soldering form. Starting from (6) we can derive formulas for the products of soldering forms with all their spinor indices contracted (these are useful to translate spinor expressions into tensor ones). For example

$$\sigma_{AA'}^{AB} \sigma_{B'C'}^{AC'} \sigma_{C'D'}^{DC'} \sigma_{d'd'}^{A'} = \frac{1}{2} (\eta_{dab} + g_{ab} g_{cd} + g_{cb} g_{da} - g_{ca} g_{db}), \tag{7}$$

where $\eta_{abcd}$ is the volume form of the metric $g_{ab}$. It is possible to generalise this formula for the case of a product of more soldering forms. These can be written as contracted products of the quantity

$$G_{abcd} \equiv \frac{1}{2} (\eta_{abcd} + g_{ab} g_{cd} - g_{ac} g_{bd} - g_{ab} g_{cd}). \tag{8}$$

Combining eq. (7) and its complex conjugate we obtain the spinor counterpart of $\eta_{abcd}$, written as follows

$$\eta_{abcd} \sigma_{AA'}^{AB} \sigma_{BB'}^{CC} \sigma_{dd'}^{DD} = i (\epsilon_{AC} \epsilon_{BD} \mathfrak{T}_{A'B'} \mathfrak{T}_{B'C'} - \epsilon_{AD} \epsilon_{BC} \mathfrak{T}_{A'B'} \mathfrak{T}_{B'C'}). \tag{9}$$

We finish this review about spinor algebra by recalling an important result dealing with the decomposition of an arbitrary spinor into irreducible parts under the Lorentz group [4].

**Theorem 1.** Any spinor $\xi_{A_1 \ldots A_k B_1 \ldots B_k}, p, q \in \mathbb{N}$ can be written as the sum of a totally symmetric spinor $\hat{\xi}_{(A_1 \ldots A_k B_1 \ldots B_k)}$ plus terms which are products of the spin metric $\epsilon_{AB}$ (or its complex conjugate $\bar{\epsilon}_{AB}$) times totally symmetric spinors of lower rank.
2.1. Spinor calculus

So far all our considerations were algebraic in nature, but we can also perform our construction for the case of a Lorentzian manifold \((M,g)\) as follows: the construction performed in previous paragraphs is carried out taking as vector space \(L\), the tangent space \(T_p(M)\) of an arbitrary point \(p \in M\) which is endowed with the Lorentzian scalar product \(g_{\mu\nu}\). In this way it is possible to introduce a complex vector space \(S_p\) and a quantity \(\sigma^a_{\mu\nu}\). Now the set \(S(M) \equiv \bigcup_{p \in M} S_p\) is a vector bundle with the manifold \(M\) as the base space and the group of linear transformations on \(\mathbb{C}^2\) as the structure group. We will call this vector bundle the spinor bundle with \(M\) as the base manifold. These bundles are tensor bundles and we denote each of these tensor bundles by \(\mathcal{S}^{\mathcal{R},\mathcal{L}}(M)\), where the meaning of the labels \(r, R, s, S\) is the obvious one. In general we will suppress these labels and use just the notation \(\mathcal{S}(M)\) as a generic symbol for these tensor bundles. Sections on \(\mathcal{S}(M)\) are written using abstract indices and we follow the same conventions as in the case of the vector spaces \(L\) and \(S\). Sections of any of the bundles \(\mathcal{S}^{(0,1)}(M)\) are called spinor fields or simply spinors. As usual there is a complex conjugate counterpart of this bundle, denoted by \(\overline{\mathcal{S}}(M)\).

**Definition 1** (Spin structure). *If the quantity \(\sigma^a_{\mu\nu}\) varies smoothly on the manifold \(M\), then one can define a smooth section, denoted by \(\sigma^a_{\mu\nu}\). When this is the case we call the smooth section \(\sigma^a_{\mu\nu}\) a smooth spin structure on the Lorentzian manifold \((M,g)\).*

Clearly a spin structure can be always defined in a neighborhood of any point \(p \in M\) but further topological restrictions are required if the spin structure is to be defined globally (see e.g. [3]).

We turn now to the study of covariant derivatives defined on the bundles \(\mathcal{S}(M), \overline{\mathcal{S}}(M)\). Let \(D_a\) denote such a covariant derivative. Then the operator \(D_a\) can act on any quantity with tensor indices and/or spinor indices. As a result, when \(D_a\) is restricted to quantities having only tensor indices we recover the standard notion of covariant derivative acting on tensor fields of \(M\). If \(D_a\) is restricted to quantities having only spinor indices then \(D_a\) is the covariant derivative acting on spinor fields. The consequence of this is that the connection coefficients and the curvature of \(D_a\) will be divided in two groups: quantities arising from the tensorial part and quantities arising from the spinorial part. The group arising from the tensorial part consists of the Christoffel symbols/Ricci rotation coefficients and the Riemann tensor of the covariant derivative restricted to the tangent bundle \(T(M)\). The group coming from the spinorial part contains the connection components and the curvature tensor of the covariant derivative restricted to the spin bundle \(S(M)\) (or \(\overline{S}(M)\)). We will refer to these as the *inner connection* and the *inner curvature* respectively. See [2, 5] for an in-depth discussion of these concepts.

**Definition 2** (Spin covariant derivative). *Suppose that \(\mathcal{S}(M)\) admits a spin structure \(\sigma^a_{\mu\nu}\). We say that a covariant derivative \(D_a\) defined on \(\mathcal{S}(M)\) is compatible with the spin structure \(\sigma^a_{\mu\nu}\) if it fulfills the property*

\[
D_a\sigma^a_{\mu\nu} = 0.
\]

*The covariant derivative \(D_a\) is then called a spin covariant derivative with respect to the spin structure \(\sigma^a_{\mu\nu}\).*

Given that any quantity antisymmetric in two spinor indices must contain the spin metric as a factor we have

\[
D_a\epsilon_{AB} = \lambda_a\epsilon_{AB}, \quad \lambda_a \equiv \epsilon_{AB}D_a\epsilon_{AB}.
\]

where \(D_a\) is any covariant derivative defined on the bundle \(\mathcal{S}(M)\). When \(D_a\) is in addition a spin covariant derivative then, combining (10) and (11) we easily deduce the additional properties

\[
D_0\sigma^a_{\mu\nu} = 0, \quad D_0g_{ab} = (\lambda_c + \bar{\lambda}_c)g_{ab}.
\]

The last equation implies that any spin covariant derivative gives rise to a *semi-metric connection* when restricted to the space-time tensor bundle. If furthermore \(D_a\) has no torsion, then it is known as a *Weyl connection*. The spin covariant derivative gets fixed if we demand additional properties on it (see e.g. [6]).

**Theorem 2.** *There is one and only one torsion-free spin covariant derivative \(\nabla_a\) with respect to the spin structure \(\sigma^a_{\mu\nu}\) which fulfills the property*

\[
\nabla_a\epsilon_{AB} = 0.
\]

*Acting with such \(\nabla_a\) on [5] gives*

\[
\nabla_a\epsilon_{AB} = 0,
\]

*which shows that the restriction of \(\nabla_a\) to quantities with tensorial indices is just the Levi-Civita covariant derivative of \(g_{ab}\).*
Consider now any spinor field $\xi^A$ and any spin covariant derivative $D_a$. Then the commutation of $D_a$, $D_b$ acting on $\xi^A$ is given by [7]

$$D_a D_b \xi^B - D_b D_a \xi^B = T^r_{ab} \xi^C$$

(15)

where $T^r_{ab}$ is the torsion of $D_a$. The mixed quantity $F_{ab}^B$ is the inner curvature mentioned above. It is antisymmetric in the tensorial indices and it fulfills the Bianchi identity [7]

$$D[a F_{bc}]_A^B + T^r_{[ab} F_{c]r}^B = 0.$$  

(16)

The spinor counterpart of the inner curvature is represented by $F_{D,C}^B$ and it can be decomposed as follows

$$F_{CC,DD}^A = X_{ABCD} \varepsilon^C D^D + \Phi_{ABC}^D \varepsilon_{CD}.$$  

(17)

The spinors $X_{ABCD}$ and $\Phi_{ABC}^D$ are called curvature spinors and they enjoy the symmetries

$$X_{ABCD} = X_{ABCD}, \quad \Phi_{ABC}^D = \Phi_{ABC}^D.$$  

(18)

We can also introduce a spinor $T^{AB}_{BBCC}$, representing the torsion. Its irreducible decomposition reads

$$T^{AB}_{BBCC} = \Omega^{AB}_{BC} \varepsilon^C + \Omega^{AB}_{BC} \varepsilon^C.$$  

(19)

where $\Omega^{AB}_{BC}$ is the torsion spinor and it fulfills the symmetries

$$\Omega^{AA'}_{BC} = \Omega^{AA'}_{BC}.$$  

(20)

The inner curvature and the Riemann tensor of $D_a$ are indeed related. To find the relation between them one computes the Ricci identity for an arbitrary vector $V^a$ and then particularises it for the special case in which $V^a = \sigma^{AA'} \varepsilon^A$. The result is

$$R_{abc}^d = (F_{abc}^D \varepsilon^C D^D + F_{abc}^D \varepsilon^C D^D) \sigma_{CC}^{dD}.$$  

(21)

In the important particular case of a torsion-free connection which is compatible with the metric (Levi-Civita connection) the curvature spinors gain further symmetries. These are

$$X_{ABCD} = X_{ABCD}, \quad X_{ABCD} = X_{ABCD}, \quad \Phi_{ABC}^D = \Phi_{ABC}^D.$$  

Given these additional symmetries we find that the spinor $\Phi_{ABC}^D$ is already in its irreducible form and it is called the Ricci spinor. The irreducible decomposition of the spinor $X_{ABCD}$ yields

$$X_{ABCD} = \Psi_{ABCD} + \Lambda (\varepsilon_{AD} \varepsilon_{BC} + \varepsilon_{AC} \varepsilon_{BD}),$$  

(22)

where $\Psi_{ABCD}$ is a totally symmetric spinor called the Weyl spinor and $\Lambda$ is related to the scalar curvature by the formula $\Lambda = R/24$. The curvature spinors and the torsion spinor are defined up to a constant scalar factor.

When working with a spin covariant derivative $D_a$ it is convenient to introduce the differential operator

$$D_{AA'} \equiv \sigma^A_{AA'} D_a,$$  

(23)

which enables us to render any expression containing spin covariant derivatives as an expression containing only spin indices. Also the commutation $D_{AA'} D_{BB'} - D_{BB'} D_{AA'}$ can be formally decomposed into irreducible parts as follows

$$D_{AA'} D_{BB'} - D_{BB'} D_{AA'} = \bar{\Omega}_{AB} \square_{AB} + \epsilon_{AB} \square_{AB'},$$  

(24)

where

$$\square_{AB} \equiv \frac{D_{AB'}}{D_{A'B'}}, \quad \square_{A'B'} \equiv \frac{D_{A'B'}}{D_{AAB'}},$$  

(25)

are linear differential operators. The action of these operators on a spinor of any rank is obtained from the spinor expression of the Ricci identity of $D_a$ and the expression of the Riemann tensor in terms of the curvature spinors. The results for the case of a rank-1 spinor are

$$\square_{BC} \varepsilon^D = X_{D,BC}^A \varepsilon^D + \Omega_{BC}^{VA} \varepsilon^D.$$  

(26)

3. The package Spinors and its relation to xAct

The package Spinors is a Mathematica package which implements the spinor calculus as described in the previous section. Spinors is part of xAct [8], which is a system to do tensor analysis by computer written mostly in the Mathematica programming language with a smaller part in C. The composition is roughly 16000 lines of Mathematica code and 2700 lines of C code. As of October 2011 the version of xAct is 1.0.3. and the complete system is free software available under the terms of the GPL license.

The system xAct is organised as a suite of interdependent Mathematica packages which can be regarded as different-purpose modules loadable on-demand. The packages and the relations among them are depicted in figure 1. The arrows indicate which packages of the suite a given package relies upon and we can see that there are three packages (xCore, xPerm and xTensor) which act as kernel for the whole implementation. This means that these packages yield the basic framework to set up any computation requiring tensor analysis. In addition the module sperm.c is a piece of code in C language devised to speed-up the group
Finally the formulae relating spinors and tensors had to be studied and coded from scratch (this was perhaps one of the most time-consuming tasks in the development of Spinors). The conclusion of all of this is that spinor theory is complex enough to develop a new package for the xAct suite.

The package spinors has already been used by a number of authors in their research. An example of this is the invariant construction of Kerr initial data in [15, 16] (see also [17] for a generalisation of these results). The present authors have also used Spinors in the investigation of the invariant properties of type D vacuum solutions of the Einstein’s field equations [18].

4. Working with Spinors

Assuming xAct has been installed one loads Spinors in a Mathematica session by typing

```
In[1]:= << xAct`Spinors`
```

This will load the package Spinors together with the other packages of the xAct suite which Spinors relies on. These are xCore, xPerm and xTensor (see section 3 and figure 1). In this work we will only explain the features of these packages which are required for our implementation and we refer the reader to their documentation for further details.

Next we need to declare a 4-dimensional Lorentzian manifold by means of the standard xAct machinery:

```
In[2]:= DefManifold[M4, 4, {a, b, c, d, f, h, p}]
In[3]:= DefMetric[[1,3,0], g[-a,-b], CD]
```

The list \{a, b, c, d, f, h, p\} corresponds to the spacetime abstract indices which will be used in tensor expressions and the list \{1,3,0\} in \texttt{DefMetric} serves to indicate the canonical form of the metric tensor g (its canonical form contains once +1, three times -1 and zero times 0, thus it corresponds to a Lorentzian metric). The symbol \texttt{CD} represents the Levi-Civita connection compatible with the metric g and in addition a number of quantities (the Riemann tensor, the Ricci tensor, the Weyl tensor, etc) are also created automatically after issuing the command \texttt{DefMetric}.

So far we have used commands belonging to xTensor and we have now the set-up necessary to start working with Spinors. The first step is the introduction of a spin structure. This is achieved as follows.
Several new objects are defined alongside this command. These are the spin bundle $\text{Spin}$, its abstract indices $(A, B, C, D, F, P)$, the spin metric $\epsilon$, the soldering form $\sigma$ and the spin covariant derivative $\text{CD}$ compatible with both the space-time metric $g$ and the spin metric $\epsilon$. The spin bundle $\text{Spin}$ together with its structures and the curvature spinors are automatically defined with this command. For example the Weyl and Ricci spinors are

\begin{verbatim}
In[5]:= DefSpinStructure[g, Spin, {A, B, C, D, F, P}, \epsilon, \sigma, CD]
\end{verbatim}

Additional options controlling the displayed form of the different quantities automatically defined can be supplied to $\text{DefSpinStructure}$. For example we may add the options

$$\text{SpinorPrefix} \to \text{SP}, \text{SpinorMark} \to \text{"S"}.$$ 

The symbol $\text{SP}$ will be prepended to the tensor (spinor) counterpart of any spinor (tensor) and the string $\text{"S"}$ will be used in the displayed representation (see below for explicit examples). From now on it will be understood that these options were used in the command $\text{DefSpinStructure}$ above.

Primed indices are entered with the "dagger" symbol $\dagger$ (entered via the keyboard shortcut $\text{\textbackslash esc} \text{dg} \text{\textbackslash esc}$). Take now the Ricci tensor $\text{RicciCD}[-a, -b]$. Its spinor counterpart is represented by $\text{SP}_{\text{RicciCD}}[-A, -B]$, where $\text{SP}$ was the tag declared through the option SpinorPrefix in $\text{DefSpinStructure}$. Again, this tag is used to construct the symbol defining the spinor counterpart of any tensorial quantity, in the way illustrated by this example. In the mathematica notebook this is

\begin{verbatim}
In[6]:= SP_RicciCD[-A, -B] // PrintAs[\epsilon] \text{"S"};
\end{verbatim}

The linking symbol "\_" (entered through the key combinations $\text{\textbackslash esc}\_\text{\textbackslash esc}$) serves to link the symbol chosen to represent the spinor prefix with the symbol representing the tensor. This linking symbol is stored in the variable $\$\text{\_\text{\textbackslash LinkCharacter}}$ which can be freely modified. In addition to this, it is possible to modify the default printing options and obtain outputs similar to the formulae described in section 2. This is done as follows for the primed indices

\begin{verbatim}
In[7]:= PrintAs[A], \"\_A\"; PrintAs[B], \"\_B\";
PrintAs[C], \"\_C\"; PrintAs[D], \"\_D\"; PrintAs[F], \"\_F\";
\end{verbatim}

Also we can modify the printing output of any tensor or spinor

\begin{verbatim}
In[8]:= PrintAs[\_A], \"\_A\";
\end{verbatim}

The command $\text{Decomposition}$ can be used to find the decomposition into irreducible parts of any other curvature spinor (it is possible to indicate the curvature spinor being decomposed as an additional argument). For example

\begin{verbatim}
In[9]:= SP_RiemannCD[-A, -B, -P, -C, -F, -D]
\end{verbatim}

\begin{verbatim}
Out[9]= \text{\_A}_\text{\_P}_{\text{\_C}_{\text{\_F}_{\text{\_D}}}}
\end{verbatim}

The spin covariant derivative $\text{CD}$ can be handled as a 2-index covariant derivative ...

\begin{verbatim}
In[10]:= CDe[-F, -F] \& SPDe[-A, -B, -C, -D]
\end{verbatim}

\begin{verbatim}
Out[10]= \text{\_F}_{\text{\_F}}_{\text{\_A}_{\text{\_B}_{\text{\_C}_{\text{\_D}}}}}
\end{verbatim}

The command $\text{SeparateSolderingForm}$ enables us to transform spinor indices into tensor ones (cf. §4.1 for more details about this).

Spinors are defined by means of $\text{DefSpinor}$, which is just a special call to the $\text{xAct}$ command $\text{DefTensor}$ (by default the option $\text{Dagger->Complex}$ is assumed).

\begin{verbatim}
In[12]:= DefSpinor[x\{\_A, -\_A\}, M]
\end{verbatim}

**DefTensor: Defining tensor $\{x\_A, -\_A\}$.
**DefTensor: Defining tensor $\bullet \{\_A, -\_A\}$. 

If one wishes to work with Hermitian spinors then this is done by using the option $\text{Dagger \_> Hermitian}$ on the command $\text{DefSpinor}$. Under this assumption one has that $\epsilon\{\_A, -\_A\}$ is invariant under complex conjugation

\begin{verbatim}
In[13]:= \_A\_A; Dagger; InputForm
Out[13]= \_A\_A
\end{verbatim}

The canonicalizer of $\text{xAct}$ is $\text{ToCanonical}$ and it can deal with the canonicalization of spinor expressions without any additional user input. It is beyond the present article to explain the workings of the canonicalization procedure and the reader is referred to [11] and the $\text{xTensor}$ documentation for additional details. Similarly, the $\text{xAct}$ command $\text{ContractMetric}$ takes care automatically of the conventions for raising and lowering indices in spinor expressions. We present next some explicit examples about these issues
Also, a number of quantities are automatically defined in addition to the spin covariant derivative \(\nabla\). Since we used the option Torsion->True the torsion is among those and one can work with both its tensor and spinor forms.

\begin{verbatim}
In[26]:= SeparateSolderingForm[]Tensor\[a,-b,-c]\nOut[26]= ST(D\[A\]CCB, C\[AB\]CCB, C\[AB\]CCB, C\[AB\]CCB)
In[27]:= PutSolderingFormDecomposition\[X\]
Out[27]= \[\epsilon\]\[AB\]\[CD\]X\[BD\]C + \[\Omega\]\[DA\]BC\[EC\]X\[AC\]
In[28]:= ContractMetric\[X\]
Out[28]= \[\Omega\]\[DA\]BC\[EC\]X\[AC\] + \[\Omega\]\[DA\]BC\[EC\]X\[AC\]
\end{verbatim}

The first step finds the relation between the torsion tensor and the torsion spinor and the second step computes its irreducible decomposition according to eq. (15). Any spin covariant derivative can be represented in single index and two-index notation.

\begin{verbatim}
In[29]:= nb\[-A,-A\]\[\Psi\][-C]\nOut[29]= D\[A\]C\[B\]
In[30]:= SeparateSolderingForm[]\[X,\text{nb}\]
Out[30]= C\[A\]C\[B\]D\[A\]C\[B\]
\end{verbatim}

Finally we remark that it is possible to define a spin structure for a metric connection with torsion. In this case DefSpinStructure defines the torsion spinors automatically.

4.1. Relations between tensors and spinors

One of the strongest points of Spinors is its ability to transform tensor expressions into spinor ones and back. The transformation rules are illustrated by (4) and to work out these expressions in explicit examples we need to repeatedly use (5). To illustrate how this works in Spinors let us consider the following example: suppose that we have the Riemann tensor associated to the Levi-Civita connection and we wish to find its spinor form by following (4). The procedure is then

\begin{verbatim}
In[31]:= PutSolderingForm\[Riemann\[\text{CD}\]-a,-b,-c,-d]\nOut[31]= R\[\nabla\]X\[A\]BC\[DE\]X\[AB\] + R\[\nabla\]X\[A\]BC\[DE\]X\[AB\]
In[32]:= ContractSolderingForm\[\%\]
Out[32]= SR\[\nabla\]X\[A\]BC\[DE\]X\[AB\]
\end{verbatim}

The Riemann spinor can be transformed back into a tensor as follows

\begin{verbatim}
In[33]:= SeparateSolderingFormula\[\%\][\%]
Out[33]= R\[\nabla\]X\[A\]BC\[DE\]X\[AB\]
In[34]:= PutSolderingForm\[\%\]
Out[34]= R\[\nabla\]X\[A\]
\end{verbatim}
As we see in this simple example a tensor (resp. a spinor) is transformed into a spinor (resp. a tensor) by contracting it with a number of soldering forms in the appropriate way. The insertion of soldering forms is achieved with the command \texttt{ContractSolderingForm} and the elimination of their dummy indices with \texttt{DefTensor}.

Example:

\texttt{In[36]:= DefTensor[M[-a,-b], M4].}
\texttt{Out[36]= \$5,-Q3,-Q5].}

\texttt{DefSpinorOfTensor[SPM[-A,-A], M[-a,-b], SeparateSolderingForm[M[-a,-b],-B].}

\texttt{σ[-Q, -B, -B, -A, -B] M[-B,-B]$3,-Q3,-Q5.}

\texttt{Sigma Gg[-A1]% /. TetraRule[g].}

The square of \(\Sigma_{\alpha\beta}^{ABCD}\) can be written as a transformation is shown in eq. (7) and products of soldering forms with more factors will arise when transforming complicated spinor expressions into tensor ones. The way of computing products of soldering forms in \textit{Spinors} is through the command \texttt{ContractSolderingForm}. For example, the simplest case is the product of two soldering forms

\texttt{In[39]:= \texttt{σ[a,-A,-A] σ[b,-B,A]/ContractSolderingForm.}}
\texttt{Out[39]= \texttt{Σ_{AB}.}}

The mixed quantity \(\Sigma_{\alpha\beta}^{AB}\) is entered through the keyboard as \texttt{Sigm\[a, b, -A, -B\]} and its square results in the tensor \(G_{abcd}\) introduced in eq. (8). The tensor \(G_{abcd}\) will be referred to as the tetra-metric and it is one of the quantities automatically defined by \texttt{DefMetric}. When the manifold dimension is four. In this way, if the metric name symbol is \(g\) then \(G_{abcd}\) is represented by the symbol \texttt{tetr} and eq. (8) by the rule \texttt{TetraRule[g].}

\texttt{In[40]:= Tetr[-a,-b,-c,-d].}
\texttt{Out[40]= \texttt{G_{abcd}.}}

\texttt{In[41]:= g /. TetraRule[g].}
\texttt{Out[41]= \texttt{\frac{1}{4} G_{abcd} + \frac{1}{2} G_{bc}\texttt{e} + \frac{1}{2} G_{cb}\texttt{e} + \frac{1}{2} G_{bc}\texttt{e}.}}

The square of \(\Sigma_{\alpha\beta}^{ABCD}\) is always automatically replaced by the tetra-metric

\texttt{In[42]:= Sigm\[a,-b,-A,-B\] Sigm\[-c,-d,A,B\].}
\texttt{Out[42]= \texttt{G_{abcd}.}}

The main interest of the tetra-metric is that any contracted product of soldering forms with no free spinor indices can be always expressed as a product of tetrametrics. This is precisely the kind of product which arises naturally when translating spinor expressions into tensor ones and back. Consider the following example: if \(W_{ABCD}\) is the Weyl spinor, we wish to write the scalar quantity \(\Psi_{ABCD} W_{ABCD}\) as an expression in terms of the Weyl tensor. The Weyl spinor and the Weyl tensor \(W_{abcd}\) are related through the relation

\[\Psi_{ABCD} = \frac{1}{4} W_{abcd} \sigma_{aA}^{\alpha} \sigma_{bB}^{\beta} \sigma_{cC}^{\gamma} \sigma_{dD}^{\delta},\]  

and hence the scalar \(\Psi_{ABCD} W_{ABCD}\) can be computed by replacing the Weyl spinor according to (27). Equation (27) can be written as a \(xAct\) rule in the following way.
This construct is called in xAct an index rule. Its difference with a Mathematica (delayed) rule is that dummy indices can be included in the right hand side of the index rule without caring about the collision of these indices with other dummy indices already present in the expression in which the replacement is being done. Dummy indices will be automatically re-named to avoid any index collision. The reader is referred to the xAct documentation for further details about this.

We can use now the rule defined above to find the tensor expression of any scalar invariant written in terms of the Weyl spinor. In our example the actual computation runs as follows:

The option IndicesOf[Spin] Used in ContractSolderingForm indicates that only dummy (dummy) indices in the product of soldering form have to be taken into account in the contraction. In this way the final result does not contain any spinor index and it is thus a tensor expression as desired. One can now use the TetraRule[ε] discussed before to transform the tetra-metrics into ordinary metrics and epsilon symbols (volume elements).

The spinor $\Xi_{AA',BB'}$ is called the Nester-Witten spinor and it has the algebraic property $\Xi_{AA',BB'} + \Xi_{BB',AA'} = 0$. Hence, its tensor counterpart, defined by

$$F_{ab} \equiv \sigma_a^{AA'} \sigma_b^{BB'} \Xi_{AA',BB'}$$

is an antisymmetric tensor and it can be regarded as a 2-form.

**Theorem 3.** The 2-form $F_{ab}$ fulfills the relation (Sparling identity)

$$3 \nabla_{[a} F_{bc]} = \eta_{abc} \left( Z_d^{[df]} - \frac{1}{4} G_{d[l} \xi^{f]} \right),$$

where $\xi^a$ is the tensor representing the spinor $\lambda^A \lambda^{A'}$, $G_{ab}$ is the Einstein tensor, $\eta_{abc}$ the volume 4-form (both with respect to the space-time metric) and $Z_{abc}$ is a tensor fulfilling the “dominant property”, namely for any three causal future-directed null vectors $k_1^a, k_2^a, k_3^a$ one has the property

$$Z_{abc} k_1^b k_2^c k_3^a \geq 0.$$  

**Proof:** We carry out the proof of this result using the tools introduced in section 4 (we work in the same Spinors session as the one used in that section). First of all, we need to define the spinors and tensors intervening in our problem.

The spino $\lambda^A$ be any rank-1 spinor and define the following quantity

$$\Xi_{AA',BB'} \equiv \frac{1}{2} (\lambda^A \nabla_{BB'} A^A - \lambda^B \nabla_{AA'} A^B).$$  

The set-up is as follows: let $\lambda^A$ be any rank-1 spinor and define the following quantity

$$\Xi_{AA',BB'} \equiv \frac{1}{2} (\lambda^A \nabla_{BB'} A^A - \lambda^B \nabla_{AA'} A^B).$$  

We introduce a short form for the xTensor command IndexSolve (see the documentation of xTensor for further details about IndexSolve).
We also define the shortcut canonicalization function `TC` (combination of the `xAct` commands `ContractMetric` and `ToCanonical`)

Also we define a function named `EqualTimes` to multiply by a quantity both sides of an equation and canonicalize the result in just one step.

With all these preparations we introduce the Nester-Witten spinor definition, as given by (28), in our `Spinors` session.

The resulting expression (not shown due to lack of space) consists of second covariant spin derivatives of \( \lambda \) and terms formed out of the product \( \nabla_{\lambda\lambda} A_B \). The second covariant derivatives can be eliminated by means of the spinor Ricci identity (24) and (26). In `Spinors` the procedure for doing this is as follows

The arising curvature spinors have to be decomposed into irreducible parts

We split the previous equation into two parts: terms containing covariant derivatives of \( \lambda \) and terms which do not contain any covariant derivative.

The aim is now to find the explicit tensor form of each part. The first part \( d\Xi_1 \) is an expression which is linear in the curvature spinors. We write the curvature spinors \( \Phi_{BA\bar{B}} \) and \( \Psi_{ABC} \) in terms of the trace-free Ricci tensor and the Weyl tensor respectively (the rule `WSToWT` was defined from eq. (29), see explanations coming after that equation.)

Also we need to replace the products \( \lambda_A \lambda_B \) by \( \xi_{ab} \).

Finally we eliminate the spinor indices in the previous expression.

By definition

We use now this rule in the expression for \( d\Xi_2 \) getting

We combine now the values just found for \( d\Xi_1 \) and \( d\Xi_2 \) and expand the tetra-metrics (see subsection 4.1). The final result is

The right hand side of this expression is a complicated tensor expression of 26 terms. It can be simplified though if we compute its double dual (we carry out the computation in two steps)

```
In[66]:= d\Xi2 /. WSToWT /.
   IndexRule[Φ[BA\bar{B}][a, -A, -A, -B] \to \frac{1}{2} \Phi_{BA\bar{B}}, \Psi_{ABC}[a, -A, -A, -B] \to \frac{1}{2} TFRiccic[BA, -A, -B, -A, -B]]

In[67]:= % /. IndexRule[\{B\} \to \{\bar{B}\}, \xi[\{-a\}][a, B, \bar{B}]] // TC
```

This is the tensor we define next.

We study next the part containing the covariant derivatives of \( \lambda_A \) (the expression \( d\Xi_2 \)). This expression is a linear combination of spinors of the form \( \nabla_{\lambda\lambda} A_B \nabla_{\lambda\lambda} A_B \) whose tensor counterpart has rank 3. This is the tensor we define next.

By definition

We combine now the values just found for \( d\Xi_1 \) and \( d\Xi_2 \) and expand the tetra-metrics (see subsection 4.1). The final result is

The right hand side of this expression is a complicated tensor expression of 26 terms. It can be simplified though if we compute its double dual (we carry out the computation in two steps)

```
In[68]:= d\Xi2 //合同SolderingForm[X, Indicesof\$\$Spin] // TC;
```

We use now this rule in the expression for \( d\Xi_2 \) getting

We combine now the values just found for \( d\Xi_1 \) and \( d\Xi_2 \) and expand the tetra-metrics (see subsection 4.1). The final result is

The right hand side of this expression is a complicated tensor expression of 26 terms. It can be simplified though if we compute its double dual (we carry out the computation in two steps)

```
In[74]:= EqualTimes[X, epsilong[-p, a, b, c]]
```

```
Out[74]= η_{\text{abc}} \partial_x \phi_p = Z_{\text{abc}} - Z_{\text{abc}}^\gamma - \frac{1}{2} \delta_{\gamma[\text{abc}]}} + 3 \lambda[\text{abc}] \delta_{\gamma[\text{abc}]}}
```
This last equation coincides with \eqref{eq:spinor-form-identity} and thus we conclude its validity. In addition from the spinor expression for $Z_{abc}$ we easily deduce the algebraic property \eqref{eq:spinor-form-identity} if we express the null vectors $k^a_1$, $k^a_2$, and $k^a_3$ as tensor products of spinors of rank-1 (see e.g. theorem 2.3.6 of \cite{Stewart}).

The spinor form of \eqref{eq:spinor-form-identity} has been used as the starting point of a proof of the \textit{positive mass theorem}. The rough idea is to prove that the integrals of the Einstein and Sparling 3-form over suitable hypersurfaces extending to \textit{infinity} yield a positive quantity. This is straightforward for the Einstein 3-form if the \textit{dominant energy condition} on the matter is assumed, but it requires more efforts for the sparling 3-form. In fact one needs to make a special choice of the spinor $\lambda_A$ in order to ensure the positivity and there is more than one way of achieving this (a good account of the different choices tried can be found in \cite{Bergqvist}).

Acknowledgements

AGP is supported by the Research Centre of Mathematics of the University of Minho (Portugal) through the “Fundação para a Ciência e a Tecnologia (FCT) Pluriannual Funding Program” and through project CERN/FP/116377/2010. AGP also thanks the Erwin Schrödinger Institute in Vienna (Austria) where part of this work was carried out for hospitality and financial support under the program “Dynamics of General Relativity”. JMM was supported by the French ANR Grant BLAN07-1-201699 entitled “LISA Science”, and also in part by the Spanish MICINN projects 2008-06078-C03-03 and FIS2009-11893. Both authors thank Dr. Thomas Bäckdahl for his tests of prior versions of the Spinors package and constructive criticism.

References

[1] R. Penrose, A spinor approach to general relativity, Ann. of Phys. 10 (1960) 171–201.
[2] J. M. Martin-García, xAct: efficient tensor computer algebra, \url{http://www.xact.es}
[3] D. Maître, P. Mastroloia, S@m, a Mathematica implementation of the spinor-helicity formalism, Comput. Phys. Comm. 179 (7) (2008) 501–534.
[4] K. Peeters, Introducing Cadabra: a symbolic computer algebra system for field theory problems, \url{http://www.arxiv.org/abs/hep-th/0701238}
[5] S. K. Czap, K. G. McLoughlin, J. Carminati, The automatic conversion of spinor equations to dyad form in MAPLE, Gen. Relativity Gravitation 24 (9) (1992) 911–928.
[6] R. Penrose, W. Rindler, Spinors and space-time. Vol. 1, Cambridge Monographs on Mathematical Physics, Cambridge University Press, Cambridge, 1987.
[7] A. Ashtekar, Lectures on non-perturbative canonical gravity, Advanced Series in Astrophysics and Cosmology, World Scientific, Singapore, 1991.
[8] M. Nakahara, Geometry, topology and physics, Graduate Student Series in Physics, Adam Hilger Ltd., Bristol, 1990.
[9] A. Ashtekar, G. T. Horowitz, A. Magnon-Ashtekar, A generalization of tensor calculus and its applications to physics, Gen. Relativity Gravitation 14 (5) (1982) 411–428.
[10] J. A. Schouten, Ricci Calculus, Die Grundlehren der Matematischen Wissenschaften, Springer Verlag, Berlin, 1954.
[11] J. M. Martín-García, xPerm: fast index canonicalization for tensor computer algebra, Computer Physics Communications 179 (2008) 597–603.
[12] J. M. Martín-García, R. Portugal, L. Manssur, The Invar tensor package, Comp. Phys. Comm. 177 (8) (2007) 640–648.
[13] J. M. Martín-García, D. Yllanes, R. Portugal, The Invar tensor package: Differential invariants of Riemann, Comp. Phys. Comm. 179 (8) (2008) 586–590.
[14] D. Brizuela, J. M. Martín-García, G. A. Mena Marugán, xPert: computer algebra for metric perturbation theory, Gen. Relativity Gravitation 41 (10) (2009) 2415–2431.
[15] T. Bäckdahl, J. A. Valiente Kroon, On the construction of a geometric invariant measuring the deviation from Kerr data, Ann. Henri Poincaré 11 (2010) 1225–1271.
[16] T. Bäckdahl, J. A. Valiente Kroon, Geometric invariant measuring the deviation from Kerr data, Phys. Rev. Lett. 104 (2010) 231102, 4.
[17] T. Bäckdahl, J. A. Valiente Kroon, The ‘non-Kerms’ of domains of outer communication of black holes and exteriors of stars, Proc. Roy. Soc. A 467 (2011) 1701–1718.
[18] S. B. Edgar, A. García-Parrado, J. M. Martín-García, Petrov D vacuum spaces revisited: identities and invariant classification, Class. Quantum Grav. 26 (10) (2009) 105022, 13.
[19] E. Witten, A new proof of the positive energy theorem, Comm. Math. Phys 80 (3) (1981) 381–402.
[20] J. M. Nester, A new gravitational energy expression with a simple positivity proof, Phys. Lett. A 83 (6) (1981) 241–242.
[21] J. Frauendiener, Geometric description of energy-momentum pseudotensors, Classical Quantum Gravity 6 (12) (1989) L237–L241.
[22] L. B. Szabados, On canonical pseudotensors, Sparling’s form and Noether currents, Class. Quantum Gravity 9 (11) (1992) 2521–2541.
[23] J. Stewart, Advanced general relativity, Cambridge Monographs on Mathematical Physics, Cambridge University Press, Cambridge, 1991.
[24] G. Bergqvist, Simplified spinorial proof of the positive energy theorem, Phys. Rev. D (3) 48 (2) (1993) 628–630.