ExteriorExtensions: a package for Macaulay2

LUKE OEDING

ExteriorExtensions: a package for Macaulay2

Luke Oeding

ABSTRACT: We explain a Macaulay2 implementation of a construction of Holweck and Oeding (*J. Comput. Sci.* **82** (2024), art. id. 102431) of a graded algebra structure on the direct sum of a Lie algebra $\mathfrak{g}$ (typically $\mathfrak{sl}_n$) and a $\mathfrak{g}$-module (typically a subspace of an exterior algebra $\bigwedge^{\bullet} \mathbb{C}^n$). We implement brackets, a Killing form, matrix representations of adjoint operators, and ranks of blocks of their powers.

1. INTRODUCTION. The Lie algebra $\mathfrak{g} = \mathfrak{sl}_n(\mathbb{F})$ is a matrix algebra over a field $\mathbb{F}$ with bracket $[A, B] = AB - BA$ for traceless matrices $A, B \in \mathbb{F}^{n \times n}$. The package ExteriorExtensions extends this algebra to a larger one $\mathfrak{a} = \mathfrak{g} \oplus M$ for a $\mathfrak{g}$-module M. At this time, we have only implemented this for M a sum of exterior powers of the underlying vector space $V = \mathbb{F}^n$. We say that the extension algebra has GJD (Jordan decomposition consistent with the G-action) if the $G = \text{Lie}(\mathfrak{g})$-action on M is consistent with conjugation on adjoint operators $\text{Ad}_T : \mathfrak{a} \to \mathfrak{a}$ (see [4]) for $T \in \mathfrak{a}$. Our development is focused on making tools for studying elements $x \in M$. In particular, the conjugation invariants of an adjoint operator Ad_x, such as the rank, the trace, and characteristic polynomial, are also G invariants when the algebra $\mathfrak{a}$ has GJD. In addition, the adjoint operators inherit a block structure from the grading on $\mathfrak{a}$. This package enables one to compute these invariants for many situations of interest in particular for the invariant theory of tensors, and may be useful for applications in quantum information and algebraic statistics, as well as for the study of geometric invariants of special varieties like hyper-Kähler manifolds, K3 surfaces and more. We explained these connections in [4]. In principle, the choice in field $\mathbb{F}$ could be quite general, but the majority of the testing for this package used $\mathbb{F} = \mathbb{Q}$, and imagined extensions to $\mathbb{F} = \mathbb{C}$. See Section 1.2.2 for an explicit field dependency, and Section 1.2.4 for more discussion.

The main features of the package are the following: the function exteriorExtension constructs a new graded algebra denoted $\mathfrak{a}$, called an ExteriorExtension, stored as a mutable hash table, that includes products implemented as bracket, and adjoint operators ad. The package computes standard matrix representations of adjoint operators on $\mathfrak{a}$, and facilitates the tabulation of the ranks of blocks of powers of adjoint operators via blockPowerRanks. We also compute the KillingMatrix, which may help identify the algebra $\mathfrak{a}$.

The remainder of the introduction provides a brief mathematical account of each of these objects. In Section 2 we provide examples of the functionality of the package, specifically these functions, and comment on the meaning of the computations. We note that M2 starts numbering by 0 by default, whereas

we often start numbering at 1, and we hope that this shift between explanations and implementations is not too confusing for the reader.

1.1. *Exterior algebras.*

The exterior algebra on a vector space V is the direct sum of exterior powers $\bigwedge^\bullet V = \bigoplus_{k \geq 0} \bigwedge^k V$, with product given by concatenation, which respects a $\mathbb{Z}$-grading. The exterior algebra on V is Artinian when V is finite-dimensional. However, we can obtain a $\mathbb{Z}_n$-grading if $n = \dim V$ by choosing a volume form $0 \neq \Omega \in \bigwedge^n V$ and utilizing the correspondingly defined Hodge star $\star$ operation to replace products that would have degree greater than n. We explain this more fully in Section 1.2.1.

The vector space V carries the standard group action by $G = \mathrm{SL}(V)$ and Lie algebra action by $\mathfrak{g} = \mathfrak{sl}(V)$. These actions are induced up to the exterior powers, making any direct sum of exterior powers G (or $\mathfrak{g}$)-modules.

1.2. *Extending the Lie algebra by appending an exterior algebra.*

Now fix positive integers k and $n = \dim V$. The exterior algebra $\bigwedge^\bullet V = \bigoplus_{k \geq 0} \bigwedge^k V$ is graded by degree, however, one obtains a (possibly) different grading by declaring $\bigwedge^j V$ to be the grade-1 piece and defining grading on the rest of the graded pieces inductively. More specifically, consider the graded vector space

$$\mathfrak{a} = \mathfrak{g} \oplus \bigwedge^k V \oplus \bigwedge^{2k} V \oplus \cdots = \mathfrak{g} \oplus \bigoplus_{ik \not\equiv 0 \bmod n} \bigwedge^{ik} V,$$

where we replace the scalars from the exterior algebra with $\mathfrak{a}_0 = \mathfrak{g}$ and $\mathfrak{a}_i = \bigwedge^{ik \bmod n} V$ for each i from $i = 1$ up to but not including the first i such that $ik \equiv 0 \bmod n$. Note that if k divides n, we are appending part of a subalgebra of the exterior algebra of the parts whose degrees are multiples of k. For example, when $k = 3$, and $n = 9$ we construct

$$\mathfrak{a} = \mathfrak{sl}_9 \oplus \bigwedge^3 \mathbb{F}^9 \oplus \bigwedge^6 \mathbb{F}^9,$$

but when $k = 3, n = 10$ we construct

$$\mathfrak{a} = \mathfrak{sl}_{10} \oplus \bigwedge^3 \mathbb{F}^{10} \oplus \bigwedge^6 \mathbb{F}^{10} \oplus \bigwedge^9 \mathbb{F}^{10} \oplus \bigwedge^2 \mathbb{F}^{10} \oplus \bigwedge^5 \mathbb{F}^{10} \oplus \bigwedge^8 \mathbb{F}^{10} \oplus \bigwedge^1 \mathbb{F}^{10} \oplus \bigwedge^4 \mathbb{F}^{10} \oplus \bigwedge^7 \mathbb{F}^{10}.$$

This vector space $\mathfrak{a}$ is constructed by the function `exteriorExtension` and the pieces that comprise it are accessed by `LieAlgebra` (for $\mathfrak{a}_0$) and `appendage` (for $\mathfrak{a}_{>0}$). More specifically, we construct $\mathfrak{a}_0 = \mathfrak{sl}(V)$ by making a common choice basis for the traceless $n \times n$ matrices, and we access $\mathfrak{a}_{>0}$ by calling a ring with skew-commuting variables, e_i. One can access the bases we choose for these algebras by `bases`, which produces a hash table with key `i` calling the basis of $\mathfrak{a}_i$.

Already the graded vector space space $\mathfrak{a}$ has brakets (products) defined for $\mathfrak{a}_i \times \mathfrak{a}_j \to \mathfrak{a}_{i+j}$ when $1 < i, j, i + j < n$ (the skew-commuting product on the ring $\mathbb{F}[\mathbb{F}^n]$). In the next section, we explain how to extend the definition of the bracket to the entire graded vector space in an equivariant fashion.

One of the main results from [4] tells precisely when it is possible to make $\mathfrak{a}$ into a graded algebra respecting the group structure.

Theorem 1.1. [4, Theorem 3.18] The vector space $\mathfrak{a} = \mathfrak{sl}_n \oplus \bigoplus_{k=1 \ldots n-1} \bigwedge^k \mathbb{C}^n$ has a $\mathbb{Z}_n$-graded algebra structure with a Jordan decomposition consistent with the $G = \mathrm{SL}(V)$-action. There is a unique (up to

scale) equivariant bracket product that agrees with the $\mathfrak{g}$-action on each $\mathfrak{g}_i$. If $n = 2k$, the equivariant bracket must satisfy the property that the restriction to $\bigwedge^k \mathbb{C}^n \times \bigwedge^k \mathbb{C}^n \to \mathfrak{g}$ must be commuting when k is odd and skew-commuting when k is even. For any k such that $2k \neq n$ the bracket $\bigwedge^k \mathbb{C}^n \times \bigwedge^k \mathbb{C}^n \to \bigwedge^{2k \bmod n} \mathbb{C}^n$ must be skew-commuting when k is odd and commuting when k is even.

This result here is that a natural equivariant bracket exists, so it makes sense to try to implement it in the first place, and moreover, since the structure tensor B for the algebra $\mathfrak{a}$ is essentially unique, there aren't many choices in its formulation. In particular, the only choices we have are choices of scalar multiples of the blocks $B_{i,j,i+j}$ of the tensor. We implement that product as a method function in M2 called `bracket`. Next, we give a detailed description of the essentially unique structure tensor for this algebra.

1.2.1. *The structure tensor defining the bracket.* If V is a vector space over a field $\mathbb{F}$, we let V^* denote the dual vector space of $\mathbb{F}$-linear functionals. Following the perspective of [5] a tensor $T \in V_1 \otimes \cdots \otimes V_m$ can be equivalently viewed in many ways, such as a linear functional $T \colon V_1^* \otimes \cdots \otimes V_m^* \to \mathbb{F}$, as a linear map $T \colon V_1^* \otimes \cdots \otimes \widehat{V_i^*} \otimes \cdots \otimes V_m^* \to V_i$ (where $\widehat{}$ denotes omission), or as a multilinear map $T \colon V_1^* \times \cdots \times V_m^* \to \mathbb{F}$.

Recall that a structure tensor $B \in \mathfrak{a}^* \otimes \mathfrak{a}^* \otimes \mathfrak{a}$ representing the bilinear product $[\ ,\ ] \colon \mathfrak{a} \times \mathfrak{a} \to \mathfrak{a}$ is equivalent information to the algebra structure on $\mathfrak{a}$. We denote by $B_{i,j,k} \in \mathfrak{a}_i^* \otimes \mathfrak{a}_j^* \otimes \mathfrak{a}_k$ the graded pieces of B. To respect the grading we must require that $B_{i,j,k} = 0$ whenever $i + j \not\equiv k \bmod n$.

For $B_{0,0,0}$ we have the standard skew-commuting bracket on $\mathfrak{sl}_n$, namely $[A, B] = AB - BA$. For $B_{0,j,j}$ (and for $B_{j,0,j}$) with $j > 0$ we use the standard Lie algebra action of $\mathfrak{sl}_n$ on the fundamental representation $\bigwedge^{jk} \mathbb{C}^n$, namely for $A \in \mathfrak{sl}_n$ and $T \in \bigwedge^{jk} V$ we take $[A, T] := A.T$, where $.$ denotes the Lie algebra action. Similarly $[T, A] = -A.T$.

For $i > 0$ we have two possibilities for $B_{i,j,(i+j)\bmod n}$ depending on whether or not $(i+j) \bmod n = 0$. When $(i+j) \equiv 0 \bmod n$ we separate two further cases depending on whether $i \leq n/2$. When $0 < i + j < n$ we use the standard product in the exterior algebra as the bracket.

If $i + j \equiv 0 \bmod n$, for S of degree $i \leq n/2$ and T of degree $j > n/2$ we utilize the Hodge star $\star T$ to change degrees and compute as follows. We declare an orthonormal basis $e_1, \ldots, e_n$ of $V = \mathbb{F}^n$ and note that the Hodge star is determined by a volume form which we choose to be $\Omega = e_1 \wedge \cdots \wedge e_n$. The standard Hodge star operator for $T \in \bigwedge^j V$ producing $\star T \in \bigwedge^{n-j} V$ is implemented as follows:

$$\star T = \sum_{\alpha \in \binom{[n]}{j}} \mathrm{sgn}(\alpha) \left(\left(\prod_{i \in \alpha} e_i \right) \lrcorner T \right) \prod_{i \in \alpha^c} e_i,$$

where $\binom{[n]}{j}$ denotes the ordered subsets of size j of $[n] = \{1, \ldots, n\}$, the products are taken in the exterior algebra $\bigwedge^\bullet [e_1, \ldots, e_n]$, and $\lrcorner$ denotes contraction. We note that while it is certainly the case that $\bigwedge^{n-j} V \cong \bigwedge^j V^*$ (the dual vector space), for $T \in \bigwedge^j V$ we still consider $\star T \in \bigwedge^{n-j} V$ in the not-dual coordinates since in this case $i + j = n$ with $n > j > n/2$ we have $n - j = i$, so the pair $(S, \star T)$ consists of elements of equal degree (i). Taking a pair of partial derivatives with respect to elements of degree 1 takes a pair $(S, \star T)$ of elements of degree i to a pair $\left(\partial S / \partial e_p, \partial \star T / \partial e_q \right)$ of elements both of degree $i - 1$. Contracting the resulting pair produces a number, which we store as the (p, q)-entry of a matrix.

Projection (denoted π) to traceless matrices produces an element of $\mathfrak{sl}_n$,

$$\pi: \ \mathbb{F}^{n \times n} \to \mathfrak{sl}_n, \quad A \mapsto A - \frac{\mathrm{tr}(A)}{n} I_{n \times n} \ .$$

We have chosen a global scalar to calibrate and match the outcomes of [7] (see the end of example 4) which considered the case $\mathfrak{sl}_9 \oplus \bigwedge^3 \mathbb{C}^9 \oplus \bigwedge^3 \mathbb{C}^9$ as a model for the exceptional Lie algebra $\mathfrak{e}_8$. In summary, the map that takes two elements of the exterior extension to a matrix in $\mathfrak{sl}_n$ is

$$\mathfrak{a}_i \times \mathfrak{a}_j \to \mathfrak{a}_0, \quad (S, T) \mapsto [S, T] = -k\pi \left(\left(\frac{\partial S}{\partial e_p} \lrcorner \frac{\partial \star T}{\partial e_q} \right)_{1 \le p, q \le n} \right) \quad \text{when } i + j = n \text{ and } j > n/2.$$

For $i \ge n/2$ and $j < n/2$ we reverse the role of the Hodge dual and proceed as in the previous paragraph.

Finally, when $i + j > n$ we use Hodge star to convert to a product that we have already computed. If S has degree $i > n/2$ or T has degree $j > n/2$ then we take $[S, T] = \star[\star S, \star T]$ and use the bracket previously defined for $i + j \le n$.

Finally, we remark that the products that have degree sum $i + j > n$ on the level of basic elements $S = e_I$ and $T = e_J$ essentially result in $[S, T] \sim e_{I \cap J}$, however one has to pay attention to signs. Likewise, when the degree sum is $i + j < n$ on the level of basic elements $S = e_I$ and $T = e_J$ essentially results in $[S, T] \sim e_{I \cup J}$, however again one has to pay attention to signs.

1.2.2. *Adjoint operators.* The standard matrix representation of the adjoint operator on $\mathfrak{a}$ defined by $\mathrm{Ad}_x(y) = [x, y]$ is constructed by choosing an ordered basis of $\mathfrak{a}$, and expressing the results of $\mathrm{Ad}_x(y)$ for each basis vector y in terms of the standard basis as the corresponding column. One caveat is that we have to choose a basis of $\mathfrak{sl}_n$, and there is not a "standard" choice for the basis of the Cartan subalgebra of traceless diagonal matrices. We choose a common one $\{E_{ii} - E_{i+1,i+1} \mid i = 1 \ldots n - 1\}$. Then we have to solve a system of equations to find the coordinates for the cartan rather than just extracting the entries of the matrix. Solving this system of equations can depend on the field $\mathbb{F}$ because we invert a matrix over $\mathbb{F}$ to find the solution.

Not that the adjoint operator is a linear map $\mathrm{Ad}_x: \mathfrak{a} \to \mathfrak{a}$, which is represented by a (large) matrix. The structure tensor is graded so that $B_{i,j,k}: \mathfrak{a}_i \times \mathfrak{a}_j \to \mathfrak{a}_k$. The map respects the grading on $\mathfrak{a}$ implies that $B_{i,j,k} = 0$ unless $k \equiv i + j \bmod n$. This, in turn, induces a block structure on the matrix representing Ad_x since $\mathrm{Ad}_x(y) = \bigoplus_{i,j,k}(B_{i,j,k}(x, y))$. For example, in the 2-graded case, with $x \in \mathfrak{a}_1$ we have the blocking

$$\mathrm{Ad}_x = \begin{pmatrix} 0 & * \\ * & 0 \end{pmatrix},$$

where the 0's and $*$'s respectively represent block matrices, and column blocks (and row blocks) are respectively of sizes ($\dim \mathfrak{a}_0, \dim \mathfrak{a}_1$). In the 3-graded case, with $x \in \mathfrak{a}_1$ we have the blocking

$$\mathrm{Ad}_x = \begin{pmatrix} 0 & 0 & * \\ * & 0 & 0 \\ 0 & * & 0 \end{pmatrix},$$

with column blocks (and row blocks) respectively of sizes ($\dim \mathfrak{a}_0, \dim \mathfrak{a}_1, \dim \mathfrak{a}_2$).

The ranks of the blocks of Ad_x are invariants of x, as are the ranks of blocks of the powers of Ad_x. We collect these ranks in a table and display them with `prettyBlockPowerRanks`.

We note that we implement the ranks of powers of blocks "under the hood" by extracting blocks of $(\mathrm{Ad}_x)^k$ as submatrices of the appropriate sizes (determined by the dimensions of the graded pieces, and the respective ordered bases we constructed to carry out our computations) and computing their ranks, and moreover, we compute $(\mathrm{Ad}_x)^k$ sequentially. We note that it is, in theory, possible to make a generic adjoint operator since the map $\mathfrak{a} \to \mathrm{End}\,\mathfrak{a}$ is linear, and that would produce Ad_x as a matrix of linear forms on $\mathfrak{a}$. Computing with a large matrix of linear forms can quickly become costly computationally, so we prefer to only work with specific numerical instances, where computing powers and ranks of powers is computationally inexpensive.

The Jacobi identity is stated via adjoint operators as $[\mathrm{Ad}_x, \mathrm{Ad}_y] = \mathrm{Ad}_{[x,y]}$ [3]. We can check whether this identity holds for the algebras we construct either on random elements of each grade, or on bases.

1.2.3. *The Killing form.* Recall that the Killing form for an algebra is a quadratic form on $\mathfrak{a}$ defined by $B(x, y) = \mathrm{trace}(\mathrm{Ad}(x) \cdot \mathrm{Ad}(y))$ (unfortunately the symbol B is being reused.) A Lie algebra is an algebra that has a skew-commuting bracket that satisfies the Jacobi identity. Cartan's criterion states that a Lie algebra is semisimple (a direct sum of simples) if and only if the Killing form is nondegenerate. Angles between weight basis vectors are obtained via the inner product induced from the Killing form, which in turn defines a Dynkin diagram. This information can be used to identify the Lie algebra.

1.2.4. *On fields of definition.* This package was written with the fields $\mathbb{C}$ or $\mathbb{R}$ in mind because of our knowledge of results in representation theory over the real and complex numbers. However, much of the functionality of M2 is better suited for exact arithmetic, over $\mathbb{Q}$ or $\mathbb{Z}_p$, for instance. In one place the package attempts to find an expression of a matrix in terms of a specially chosen basis of $\mathfrak{sl}_n$, which we carry out by solving a system of equations. Solving this linear system inverts a matrix (the locally defined matrix Qmat in the package), and in M2 this operation has different behavior based on the ground field. At this time of development, the author has not tested behavior over other rings or fields, and so the functions are implemented for exact arithmetic, and M2 will throw an error when imprecise fields like $\mathbb{R}$ or $\mathbb{C}$ are passed to the exteriorExtension command. It is still possible to have some of this functionality, and to do this is typically best to start working over an exact field, and only lift to $\mathbb{R}$ or $\mathbb{C}$ when necessary or when it makes sense to do so. By not working over inexact fields until necessary, we reduce the chances for numerical errors.

2. EXAMPLES.   The algebras $\mathfrak{a} = \mathfrak{sl}_{2m} \oplus \bigwedge^m \mathbb{C}^{2m}$ for $m = 2$ and $m = 4$ are actually simple Lie algebras, respectively $\mathfrak{sl}_4 \oplus \bigwedge^2 \mathbb{C}^4 \cong \mathfrak{sp}_6$ (a symplectic Lie algebra) and $\mathfrak{sl}_8 \oplus \bigwedge^4 \mathbb{C}^8 \cong \mathfrak{e}_7$ (an exceptional Lie algebra). An excellent overview of fine gradings of simple Lie algebras is given in [2]. It is not immediately clear to the author if this example falls within those fine gradings classified in [2, Example 6].

Example 1. We show the use of the functions exteriorExtension, makeTraceless, bracket, ad, appendage, and KillingMatrix, to illustrate the known sporadic isomorphism $\mathfrak{sl}_4 \oplus \bigwedge^2 \mathbb{C}^4 \cong \mathfrak{sp}_6$. Recall from [3, Ch. 15.2 and Ch.17.1] the structure of the two algebras. The weights of $\bigwedge^2 \mathbb{C}^4$ are the vertices of an octahedron, while the weights of $\mathfrak{sl}_4$ are the midpoints of the edges of a cube. These geometries are also present in the weights of a basis of $\mathfrak{sp}_6$, see [3, p220, p254].

In this example we construct $\mathfrak{a} = \mathfrak{sl}_4(\mathbb{F}) \oplus \bigwedge^2 \mathbb{F}^4$ for $\mathbb{F} = \mathbb{Q}$, and note some of the salient features of this algebra extension, even though we may be mainly interested in the theory of the complex Lie algebras. Near the end of the example, we note that the Killing form of the algebra has full rank and all real eigenvalues, and this means that the isomorphism $\mathfrak{a} \cong \mathfrak{sp}_6$ holds as an isomorphism of real Lie algebras.

We set the algebra as an `ExteriorExtension` via:

```
loadPackage"ExteriorExtensions"
ea24 = exteriorExtension(2,4,e,QQ)
```

We will see that $\mathfrak{sl}_4 \oplus \bigwedge^2 \mathbb{C}^4$ with the (essentially unique) bracket coming from the GJD criterion forms a Lie algebra. We can check (at least on random points with rational coefficients) if the bracket satisfies the Lie algebra axioms in each graded piece of the algebra. If we had the patience we could also check these identities for all pairs of basis vectors. For illustration we only perform tests for randomly chosen elements. For grade 0 times grade 0, we first make 2 random traceless matrices:

```
A0 = makeTraceless random(QQ^4,QQ^4)
B0 = makeTraceless random(QQ^4,QQ^4)
```

To access the `bracket` and adjoint operator `ad` for the algebra `ea24` that we have constructed, we do so with the dot, which we see in the following random check of the skew symmetry and Jacobi identities:

```
ea24.bracket(A0,B0) + ea24.bracket(B0,A0)
    0
ea24.ad(ea24.bracket(A0,B0)) - ea24.bracket(ea24.ad(A0),ea24.ad(B0))
    0
```

We shouldn't be surprised that we get 0 for both since the grade 0 piece is a Lie algebra.

For a pair of grade 1 elements we call for random elements of the `appendage` and check their brackets are skew and satisfy the Jacobi identity as follows:

```
A1 = random(2, ea24.appendage)
B1 = random(2, ea24.appendage)
ea24.bracket(A1,B1) + ea24.bracket(B1,A1)
    0
ea24.ad(ea24.bracket(A1,B1)) - ea24.bracket(ea24.ad(A1),ea24.ad(B1))
    0
```

For a point in grade 0 and a point in grade 1, we see the axioms hold, but we recall the caveat that it is important to make the matrix traceless (we implemented this projection as `makeTraceless`), otherwise the Jacobi identity fails. We proceed as before:

```
ea24.bracket(A0,B1) + ea24.bracket(B1,A0)
    0
ea24.ad(ea24.bracket(A0,B1)) - ea24.bracket(ea24.ad(A0),ea24.ad(B1))
    0
```

We can also construct a Killing matrix and see that it is 21×21 and nondegenerate. The only possibility for a simple Lie algebra over $\mathbb{C}$ of dimension 21 is $\mathfrak{sp}_6$.

```
K = KillingMatrix ea24
    | 20  -10 0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | -10 20  -10 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   -10 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  10 0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  10 0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  0  10 0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  10 0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  10 0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  0  10 0  0   0  0   0  0  0  |
    | 0   0   0   10 0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  10 0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  10 0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  10 0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  10 0  0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  10 0  0  0  0  0  0  0  0  0   0  0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  0  -10 |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   0  10 0  |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  0   -10 0  0 |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   0  -10 0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   10 0   0  0  0  |
    | 0   0   0   0  0  0  0  0  0  0  0  0  0  0  0  0  -10 0  0  0   0  0  0  |
rank K
    21
```

We implemented the `KillingMatrix` as a function rather than something that gets constructed when the algebra is constructed since, without additional tricks, it may be time-consuming to construct this matrix. One checks that the Killing matrix has a full set of real eigenvectors, so the isomorphism $\mathfrak{a} \cong \mathfrak{sp}_6$ holds over $\mathbb{R}$.

```
toList eigenvalues K
    {34.1421, 20, 5.85786, 10, -10, 10, -10, 10, -10, 10, -10, 10,
    -10, 10, -10, 10, -10, 10, -10, 10, -10}
```

Here we note the first instance of a change in the field of definition. To compute eigenvalues M2 automatically upgraded the matrix K, or at least its eigenvalues, to work over $\mathbb{C}$.

The tensors $e_i \wedge e_j \pm e_k \wedge e_l$ for i, j, k, l distinct have diagonalizable adjoint forms (we call them semisimple or ad-semisimple). We can check that the geometric and algebraic multiplicities of the eigenvalues add up:

```
E = ea24.appendage
A = ea24.ad(E_0*E_1 - E_2*E_3)
tally apply(toList eigenvalues (A), xx-> round(2, realPart xx) +
```

```
round(2, imaginaryPart xx)*ii)
     Tally{-1.41 => 5}
           0 => 11
           1.41 => 5
rank A
     10
```

We also note that even though *A* is defined over `QQ`, we can let M2 work with square roots.

```
tally apply(toList eigenvalues (1/sqrt(2)*A), xx->
round(2, realPart xx) + round(2, imaginaryPart xx)*ii)
     Tally{-1 => 5}
           0 => 11
           1 => 5
```

The semisimple structure is also indicated by the block ranks computation, via `blockPowerRanks` (to produce a matrix as a result) or via `prettyBlockPowerRanks` for onscreen printing:

```
ea24.prettyBlockPowerRanks A
     +---+---+---+---+-----+
     |g00|g01|g10|g11|total|
     +---+---+---+---+-----+
     |0  |5  |5  |0  |10   |
     +---+---+---+---+-----+
     |5  |0  |0  |5  |10   |
     +---+---+---+---+-----+
     |0  |5  |5  |0  |10   |
     +---+---+---+---+-----+
     |5  |0  |0  |5  |10   |
     +---+---+---+---+-----+
```

The rows in the above table correspond to subsequent powers of the matrix, and we stop computation after sufficient repetition or zero total rank occurs.

We can check that neither the elements nor their adjoint forms commute, so they're not simultaneously diagonalizable.

```
ea24.bracket(E_0*E_1 + E_2*E_3,E_0*E_2 + E_1*E_3)
     {1} | 0  0 0 -1 |
     {1} | 0  0 1 0  |
     {1} | 0  1 0 0  |
     {1} | -1 0 0 0  |
```

We checked that the only elements of $\bigwedge^2 \mathbb{C}^4$ that commute with $e_i \wedge e_j + e_k \wedge e_l$ are scalar multiples of itself. So we can't hope for a Cartan subalgebra that intersects $\bigwedge^2 \mathbb{C}^4$ and includes more than one

of these forms. We did this computation by accessing the basis (of the grade 1 part) of the algebra via `ea24.bases`.

```
L = apply(ea24.bases#1, xx-> ea24.bracket(xx,e_0*e_1 + e_2*e_3))
```

Then we prepare additional variables, and solve the system of equations coming from the commutation conditions:

```
R = QQ[x_0.. x_5]
decompose ideal flatten entries sum(6, i-> sub(L_i, R)*x_i)
    ideal (x , x , x , x , x  - x )}
           4   3   2   1   0    5
```

Here we note that we have changed rings again only when necessary. Attention should be paid to the ring of definition for the matrices the package produces, and sometimes substitution commands are necessary.

Example 2. We consider $\mathfrak{sl}_6 \oplus \bigwedge^3 \mathbb{C}^6$, which is not a Lie algebra as it fails to be skew-commuting (it is commuting on the grade 1 piece) and it fails the Jacobi identity.

```
ea36 = exteriorExtension(3,6,QQ);
A = random(3, ea36.appendage);
B = random(3, ea36.appendage);
ea36.bracket(A,B) + ea36.bracket(B,A) == 0
    false
ea36.bracket(A,B) - ea36.bracket(B,A) == 0
    true
ea36.ad(ea36.bracket(A,B)) - ea36.bracket(ea36.ad(A),ea36.ad(B)) == 0
    false
```

Interestingly, the trace of the 4th power of the adjoint operator still produces the single (degree 4) generator of the invariant ring $\mathbb{C}[\bigwedge^3 \mathbb{C}^6]^{\mathrm{SL}_6} \cong \mathbb{C}[f_4]$ in this case. Computing values of this invariant is accomplished as follows:

```
E = ea36.appendage
f4 = A-> trace(A^4)
N = E_0*E_1*E_2 + E_3*E_4*E_5
f4 ea36.ad(N)
    36
```

Example 3. Consider the case of $\mathfrak{sl}_8 \oplus \bigwedge^4 \mathbb{C}^8 \cong \mathfrak{e}_7$ considered in [1; 6]. Antonyan shows that a 7-dimensional Cartan subalgebra is given by 4-forms of type $e_I + e_J$ with the concatenation $I|J$ a permutation of [8], the 7 pairs of indices I, J are

$$\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}\}, \quad \{\{1, 3, 5, 7\}, \{6, 8, 2, 4\}\}, \quad \{\{1, 5, 6, 2\}, \{8, 4, 7, 3\}\},$$
$$\{\{1, 6, 8, 3\}, \{4, 7, 5, 2\}\}, \quad \{\{1, 8, 4, 5\}, \{7, 2, 6, 3\}\}, \quad \{\{1, 4, 7, 6\}, \{2, 3, 8, 5\}\},$$
$$\{\{1, 7, 2, 8\}, \{3, 5, 4, 6\}\}.$$

We can check the diagonalizability and commutativity as follows. First, we construct the algebra and our candidate for the basis of the Cartan subalgebra.

```
ea48 = exteriorExtension(4,8,e,QQ);
list2E = L-> sum(L, l-> product(l, i-> e_(i-1)));
aa = {{{1,2,3,4},{5,6,7,8}},{{1,3,5,7},{6,8,2,4}},{{1,5,6,2},{8,4,7,3}},
{{1,6,8,3},{4,7,5,2}},{{1,8,4,5},{7,2,6,3}},{{1,4,7,6},{2,3,8,5}},
{{1,7,2,8},{3,5,4,6}}};
```

Then we produce the adjoint operators for each, and compute respectively their sums of the algebraic and geometric multiplicities of eigenvalues to see that they do, in fact, agree, and hence these elements are diagonalizable.

```
for i to length(aa)-1 do(
adList_i = ea48.ad(list2E aa_i);
print(# for xx in eigenvalues adList_i list if
 not round(abs(xx)) ==0 then xx else continue, rank adList_i); )
    (66, 66)      [six more identical lines omitted]
```

Next, we check that these elements mutually pairwise commute by computing all their commutators. We store these results as a matrix whose entries are the ranks of the commutators, and note that we get the zero matrix:

```
matrix apply(7, i-> apply(7, j-> rank ea48.bracket(adList_i,adList_j)))
    0
```

Example 4. Larger algebras are also possible, such as $\mathfrak{e}_8 \cong \mathfrak{sl}_9 \oplus \bigwedge^3 \mathbb{C}^9 \oplus \bigwedge^6 \mathbb{C}^9$ from [7], and we have done extensive computations in [4]. Here are some example computations that check that the bracket satisfies the Lie-bracket conditions (skew-symmetry and Jacobi). We check bracket conditions for pairs of random elements in each grade:

```
ea = exteriorExtension(3,9)
E = ea.appendage
A0 = makeTraceless random(QQ^9,QQ^9);
B0 = makeTraceless random(QQ^9,QQ^9);
A1 = random(3, E);
B1 = random(3, E);
A2 = random(6, E);
B2 = random(6, E);
ea.bracket(A0,B0) + ea.bracket(B0,A0)
    0
```

Each of the following expressions also evaluates to 0:

```
ea.ad(ea.bracket(A0,B0)) -ea.bracket(ea.ad(A0),ea.ad(B0))
ea.bracket(A0,B1) + ea.bracket(B1,A0)
ea.ad(ea.bracket(A0,B1)) -ea.bracket(ea.ad(A0),ea.ad(B1))
ea.bracket(A0,B2) + ea.bracket(B2,A0)
ea.ad(ea.bracket(A0,B2)) -ea.bracket(ea.ad(A0),ea.ad(B2))
ea.bracket(A1,B1) + ea.bracket(B1,A1)
ea.ad(ea.bracket(A1,B1)) -ea.bracket(ea.ad(A1),ea.ad(B1))
ea.bracket(A1,B2) + ea.bracket(B2,A1)
ea.ad(ea.bracket(A1,B2)) -ea.bracket(ea.ad(A1),ea.ad(B2))
ea.bracket(A2,B2) + ea.bracket(B2,A2)
ea.ad(ea.bracket(A2,B2)) -ea.bracket(ea.ad(A2),ea.ad(B2))
```

The following match equations 2.5 and 2.6 from [7].

```
pt = E_0*E_1*E_2;
tmp = ea.bracket(pt, ea.HodgeStar(pt))
    {2} | -2/3 0     0     0   0   0   0   0   0   |
    {2} | 0    -2/3 0     0   0   0   0   0   0   |
    {2} | 0    0     -2/3 0   0   0   0   0   0   |
    {2} | 0    0     0     1/3 0   0   0   0   0   |
    {2} | 0    0     0     0   1/3 0   0   0   0   |
    {2} | 0    0     0     0   0   1/3 0   0   0   |
    {2} | 0    0     0     0   0   0   1/3 0   0   |
    {2} | 0    0     0     0   0   0   0   1/3 0   |
    {2} | 0    0     0     0   0   0   0   0   1/3 |
ea.bracket(tmp, pt)
    -2e e e
      0 1 2
```

3. THE FUTURE. What we anticipate for the future of this package is to provide functionality for theoretical work on tensors. For example, it is possible to compute an adjoint form of a tensor of small format in a short amount of time on a laptop computer. From here one may use the adjoint form to compare tensors via their adjoint block ranks, adjoint spectra, etc. This can have applications whenever one wants to separate tensor orbits, for example. This strategy was explored in [4]. The author has work in progress with collaborators where this strategy is applied to tensors that arise in phylogenetic networks. In particular, we are very interested in the cases beyond the "tame" tensor formats, where there are not finitely many orbits, and the algebra we construct $\mathfrak{a}$ is not a semisimple Lie algebra. One very interesting case is $\mathfrak{a} = \mathfrak{sl}_{12} \oplus \bigwedge^4 \mathbb{C}^{12} \oplus \bigwedge^8 \mathbb{C}^{12}$, which contains a graded subalgebra

$$\mathfrak{a}_{4,4,4} = (\mathfrak{sl}_4 \oplus \mathfrak{sl}_4 \oplus \mathfrak{sl}_4) \oplus (\mathbb{C}^4 \otimes \mathbb{C}^4 \otimes \mathbb{C}^4) \oplus (\textstyle\bigwedge^2 \mathbb{C}^4 \otimes \bigwedge^2 \mathbb{C}^4 \otimes \bigwedge^2 \mathbb{C}^4) \oplus (\bigwedge^3 \mathbb{C}^4 \otimes \bigwedge^3 \mathbb{C}^4 \otimes \bigwedge^3 \mathbb{C}^4),$$

which allows us to utilize these tools to study tensors of format $4 \times 4 \times 4$. This format is the smallest cubic format for tensors where there is not a complete irredundant classification of orbits. We are quite interested in studying this case further with these new tools.

We have tried to be as efficient as possible in the implementation of the functions in this package, but there is certainly room for improvement, both mathematically and algorithmically. On the mathematical side, if one knows already how to compute the Killing form in a combinatorial fashion it might be possible to skip the lengthy process that computes each entry of the Killing matrix as the trace of a product of adjoint operators. Algorithmically, it might also be possible to speed up the computation of the adjoint operators in the first place, and it also seems very possible to utilize sparse arrays to be more efficient with memory. In addition, it could be quite interesting to attempt to construct similar extensors for other Lie algebras (replacing $\mathfrak{a}_0 = \mathfrak{sl}_n$), or for other modules, replacing $\mathfrak{a}_{<0}$.

SUPPLEMENT.   The online supplement contains version 1.0 of `ExteriorExtensions.m2`.

REFERENCES.

[1] L. V. Antonyan, "Classification of four-vectors of an eight-dimensional space", *Trudy Sem. Vektor. Tenzor. Anal.* 20 (1981), 144–161. In Russian; translated in [6].

[2] C. Draper and A. Elduque, "An overview of fine gradings on simple Lie algebras", *Note Mat.* **36** (2016), 15–34.

[3] W. Fulton and J. Harris, *Representation theory: a first course*, Graduate Texts in Mathematics **129**, Springer, 1991.

[4] F. Holweck and L. Oeding, "Toward Jordan decompositions for tensors", *J. Comput. Sci.* **82** (2024), art. id. 102431.

[5] J. M. Landsberg, *Tensors: geometry and applications*, Graduate Studies in Mathematics **128**, Amer. Math. Soc., Providence, RI, 2012.

[6] L. Oeding, "A translation of "Classification of four-vectors of an 8-dimensional space", by Antonyan, L. V., with an appendix by the translator", *Trans. Moscow Math. Soc.* **83** (2022), 227–250.

[7] E. B. Vinberg and A. G. Elashvili, "A classification of the three-vectors of nine-dimensional space", *Trudy Sem. Vektor. Tenzor. Anal.* **18** (1978), 197–233.

LUKE OEDING:

oeding@auburn.edu
Department of Mathematics, Auburn University, Auburn, AL 36830, United States