MINIMAL POLYNOMIALS AND MULTIVECTOR INVERSES IN NON-DEGENERATE CLIFFORD ALGEBRAS

DIMITER PRODANOV[1,2]

PAML-LN, IICT, Bulgarian Academy of Sciences, Sofia, Bulgaria;
Neuroelectronics Research Flanders and EHS, IMEC, Leuven, Belgium

Abstract. Clifford algebras are an active area of mathematical research with numerous applications in mathematical physics and computer graphics among many others. The paper demonstrates an algorithm for the computation of inverses of such numbers in a non-degenerate Clifford algebra of an arbitrary dimension. This is achieved by the translation of the classical Faddeev-LeVerrier-Souriau (FVS) algorithm for characteristic polynomial computation in the language of the Clifford algebra. The FVS algorithm is implemented using the Clifford package in the open-source Computer Algebra System Maxima. Symbolic and numerical examples in different Clifford algebras are presented.

1. Introduction

Clifford algebras provide the natural generalizations of complex, dual and split-complex (or hyperbolic) numbers into the concept of *Clifford numbers*, i.e. general multivectors. The power of Clifford or, geometric, algebra lies in its ability to represent geometric operations in a concise and elegant manner. The development of Clifford algebras is based on the insights of Hamilton, Grassmann, and Clifford from the 19^{th} century. After a hiatus lasting many decades, the Clifford geometric algebra experienced a renaissance with the advent of contemporary computer algebra systems. Clifford algebras can be implemented in a variety of general-purpose computer languages and computational platforms. There are multiple actively developed applications in computer-aided design (CAD), computer vision and robotics, protein folding, neural networks, modern differential geometry, genetics, and mathematical physics. Recent years have seen renewed interest in Clifford algebra and its implementations in various computational platforms. There are implementation for the major computer algebra systems, such as Maple, Matlab, Mathematica, Maxima, as well as domain-specific applications – i.e. *Ganja.js* for JavaScript, *GaLua* for Lua, `http://spencerparkin.github.io/GALua/`, *Galgebra* for Python, `https://galgebra.readthedocs.io/`, *Grassmann* for Julia. `https://grassmann.crucialflow.com/`.

Computation of inverses of multivectors has drawn continuos attention in the literature as the problem was only gradually solved [7, 1, 6, 15]. In order to compute an inverse of a multivector, previous contributions used series of automorphisms of

E-mail address: `dimiter.prodanov@imec.be`.

special types discussed in Sec. 2.3. This allows one to write basis-free formulas with increasing complexity.

As a main application, the present contribution demonstrates an algorithm for multivector inversion, based on the based on Faddeev–LeVerrier–Souriau (FVS) algorithm. The algorithm is implemented in the Computer Algebra System *Maxima* using the Clifford package [12, 10]. The theory enabling the FVS algorithm has been presented in a preliminary form at the Computer Graphics International conferee, CGI 2023, Shanghai, Aug 28^{th} – Sept 1 $^{\text{st}}$ 2023 (in press). Unlike the original FVS algorithm, which computes the characteristic polynomial and has a fixed number of steps, the present Clifford FVS algorithm involves only Clifford multiplications and subtractions of scalar parts and has a variable number of steps, depending on the spanning subspace of the multivector. The correctness of the algorithm is proven using an algorithmic, constructive representation of a multivector in the matrix algebra over the reals, but it by no means depends on such a representation. The present FVS algorithm is in fact a proof certificate for the existence of an inverse. To the present author's knowledge the FVS algorithm has not been used systematically to exhibit multivector inverses.

The paper is organized in the following way. Sec. 2 introduces the notation. Sec. 3 exhibits a real matrix representation of the algebra. Sec. 4 discusses the characteristic and minimal polynomials of multivectors. Sec. 5 discusses the multivector inverse and derives the FVS multivector algorithm. Sec. 6 introduces the notion of rank of a multivector. Sec. 7 demonstrates the algorithm. Sec. 8 gives the outlook of the work.

2. Notation and Preliminaries

2.1. Notation.

$C\ell_n$ will denote a Clifford algebra of order n but with unspecified signature. Clifford multiplication is denoted by simple juxtaposition of symbols. Algebra generators will be indexed by Latin letters. Multi-indices will be considered as index **lists** and not as sets and will be denoted with capital letters. The operation of taking k-grade part of an expression will be denoted by $\langle . \rangle_k$ and in particular the scalar part will be denoted by $\langle . \rangle_0$. Set difference is denoted by $\triangle$. Matrices will be indicated with bold capital letters, while matrix entries will be indicated by lowercase letters. The *scalar product* of the blades will be denoted by $*$; $\widehat{}$ in superscript will denote the grade negation operation, while $\widetilde{}$ – the reversion of Clifford products. The degree of the polynomial P will be denoted as $\deg P$.

2.2. General Definitions.

Definition 1. *The generators of the Clifford algebra will be denoted by indexed symbol e. It will be assumed that there is an ordering relation $\prec$, such that for two natural numbers $i < j \Longrightarrow e_i \prec e_j$. The* ***extended basis*** *set of the algebra will be defined as the ordered power set $\mathbf{B} := \{P(E), \prec\}$ of all generators $E = \{e_1, \dots, e_n\}$ and their irreducible products.*

Definition 2 (Scalar product)**.** *Define the diagonal scalar product matrix as*

$$\mathbf{G} := \{\sigma_{IJ} = e_I * e_J |\ e_I, e_J \in \mathbf{B}, I \prec J\} \tag{1}$$

A multivector will be written as $A = a_0 + \sum_{k=1}^{r} \langle A \rangle_k = a_0 + \sum_J a_J e_J$, where J is a multiindex, such that $e_J \in \mathbf{B}$. In other words, J is subset of the power set of the first n natural numbers $P(n)$. The maximal grade of A will be denoted by $\mathrm{gr}[A]$. The pseudoscalar will be denoted by I.

A multivector for which all the coefficients are non zero will be called full-grade or generic multivector.

Definition 3 (Span of a multivector)**.** *The span of the multivector A, written as* $\mathrm{span}[A]$*, is defined as the minimal ordered set of generators* $\mathrm{span}[A] := \{e_i\}$ *for which*

$$(A - \langle A \rangle_0) \wedge e_J = 0, \quad e_J \in \mathbf{B}$$

It is clear that $\mathrm{span}[A] \subseteq E$ and $\mathrm{span}[A] = E$ only for a full grade multivector.

Definition 4 (Sparsity property)**.** *A (square) matrix has the sparsity property if it has exactly one non-zero element per column and exactly one non-zero element per row. Such a matrix we call sparse.*

Here it is useful to remind the definition of a permutation matrix, which is a square binary matrix that has exactly one entry of 1 in each row and each column, with all other entries being 0. Therefore, a sparse matrix in the sense of the above definition generalizes the notion of a permutation matrix.

2.3. **Automorphisms.** Consider the general multivector M. Most authors define two (principal) automorphisms: inversion

$$\widehat{M} := \sum_{k=0}^{n} (-1)^k \langle M \rangle_k \tag{2}$$

and grade reversion

$$M^{\sim} := \sum_{k=0}^{n} (-1)^{k(k-1)/2} \langle M \rangle_k \tag{3}$$

They can be further composed into Clifford conjugation:

$$\overline{M} := \widehat{M^{\sim}} = (\widehat{M})^{\sim} = \sum_{k=0}^{n} (-1)^{k(k+1)/2} \langle M \rangle_k \tag{4}$$

Another, less used, automorphisms are the Hitzer-Sangwine involution [7]:

$$h_J(M) := \sum_{k \in J}^{n} (-1)^k \langle M \rangle_k \tag{5}$$

for the multiindex J and the inverse (or Hermitian) automorphism

$$M^{\#} := \sum_{k=0}^{n} M_k^{-1} \tag{6}$$

which have no standard notation.

3. Clifford algebra real matrix representation map

In the present we will focus on non-degenerate Clifford algebras, therefore the non-zero elements of $\mathbf{G}$ are valued in $\{-1, 1\}$.

Lemma 1 (Sparsity lemma)**.** *If the matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *are sparse then so is* $\mathbf{C} = \mathbf{AB}$. *Moreover,*

$$c_{ij} = \begin{cases} 0 \\ a_{iq} b_{qj} \end{cases}$$

(no summation!) for some index q.

Proof. Consider two sparse square matrices $\mathbf{A}$ and $\mathbf{B}$ of dimension n. Let $c_{ij} = \sum_\mu a_{i\mu} b_{\mu j}$. Then as we vary the row index i then there is only one index $q \leq n$, such that $a_{iq} \neq 0$. As we vary the column index j then there is only one index $q \leq n$, such that $b_{qj} \neq 0$. Therefore, $c_{ij} = (0; a_{iq} b_{qj})$ for some q by the sparsity of $\mathbf{A}$ and $\mathbf{B}$. As we vary the row index i then $c_{qj} = 0$ for $i \neq q$ for the column j by the sparsity of $\mathbf{A}$. As we vary the column index j then $c_{iq} = 0$ for $j \neq q$ for the row i by the sparsity of $\mathbf{B}$. Therefore, $\mathbf{AB}$ is sparse. $\square$

Lemma 2 (Multiplication Matrix Structure)**.** *For the multi-index disjoint sets* $S \prec T$ *the following implications hold for the elements of* $\mathbf{M}$ *:*

$$\begin{array}{ccccc}
m_{\mu\lambda}\, e_S & \xrightarrow{\exists \lambda' > \lambda} & m_{\mu\lambda'} e_T & & \\
\exists \big\downarrow & & \big\downarrow & & \\
m_{\lambda\mu} e_S & \xrightarrow{\exists} & m_{\lambda\mu} m_{\mu\lambda'} e_{S \triangle T} & \xrightarrow{\exists \lambda'' = \lambda'} & m_{\lambda\lambda''} e_{S \triangle T}
\end{array}$$

so that $m_{\lambda\lambda'} = m_{\lambda\mu} \sigma_\mu m_{\mu\lambda'}$ *for some index* μ.

Proof. Suppose that the ordering of elements is given in the construction of $C\ell_{p,q,r}$. To simplify presentation, without loss of generality, suppose that e_s and e_t are some generators. By the properties of $\mathbf{M}$ there exists an index $\lambda' > \lambda$, such that $e_M e_{L'} = m_{\mu\lambda'}\, e_t$, $L' \backslash M = T$ for $L \prec L'$. Choose M, s.d. $L \prec M \prec L'$. Then for $L \prec M \prec L'$ and $S \prec T$

$$\begin{aligned}
e_M e_L &= m_{\mu\lambda}\, e_s, \quad L \triangle M = S \Leftrightarrow e_L e_M = m_{\lambda\mu}\, e_s \\
e_M e_{L'} &= m_{\mu\lambda'}\, e_t, \quad L' \triangle M = T
\end{aligned}$$

Suppose that $e_s e_t = e_{st}, st = S \cup T = S \triangle T$. Multiply together the diagonal nodes in the matrix

$$e_L \underbrace{e_M e_M}_{\sigma_\mu} e_{L'} = m_{\lambda\mu} m_{\mu\lambda'}\, e_{st}$$

Therefore, $s \in L$ and $t \in L'$. We observe that there is at least one element (the algebra unity) with the desired property $\sigma_\mu \neq 0$.

Further, we observe that there exists unique index λ'' such that $m_{\lambda\lambda''} e_{st}$. Since λ is fixed. This implies that $L'' = L' \Rightarrow \lambda'' = \lambda'$. Therefore,

$$e_L e_{L'} = m_{\lambda\lambda'} e_{st}, \quad L' \triangle L = \{s, t\}$$

which implies the identity $m_{\lambda\lambda'}\, e_{st} = m_{\lambda\mu} \sigma_\mu m_{\mu\lambda'}\, e_{st}$. For higher graded elements e_S and e_T we should write $e_{S \triangle T}$ instead of e_{st}.

$\square$

Proposition 1. *Consider the multiplication table* $\mathbf{M}$. *All elements* m_{kj} *are different for a fixed row* k. *All elements* m_{iq} *are different for a fixed column* q.

Proof. Fix k. Then for $e_K, e_J \in \mathbf{B}$ we have $e_K e_J = m_{kj} e_S, \quad S = K \triangle J$. Suppose that we have equality for 2 indices j, j'. Then $K \triangle J' = K \triangle J = S$. Let $\delta = J \cap J'$; then

$$K \triangle (J \cup \delta) = K \triangle J = S \Rightarrow K \triangle \delta = S \Rightarrow \delta = \emptyset$$

Therefore, $j = j'$. By symmetry, the same reasoning applies to a fixed column q. □

Proposition 2. *For $e_s \in \mathbf{E}$ the matrix $\mathbf{A}_s = C_s(\mathbf{M})$ is sparse.*

Proof. Fix an element $e_s \in \mathbf{E}$. Consider a row k. By Prop. 1 there is a j, such $e_{kj} = e_s$. Then $a_{kj} = m_{kj}$, while for $i \neq j$ $a_{ki} = 0$.

Consider a column l By Prop. 1 there is a j, such $e_{jl} = e_s$. Then $a_{jl} = m_{jl}$, while for $i \neq j$ $a_{il} = 0$. Therefore, $\mathbf{A}_s$ has the sparsity property. □

Proposition 3. *For generator elements e_s and e_t $\mathbf{E}_s \mathbf{E}_t + \mathbf{E}_t \mathbf{E}_s = \mathbf{0}$.*

Proof. Consider the basis elements e_s and e_t. By linearity and homomorphism of the π map (Th. 3.1): $\pi : e_s e_t + e_t e_s = 0 \mapsto \pi(e_s e_t) + \pi(e_t e_s) = \mathbf{0}$. Therefore, for two vector elements $\mathbf{E}_s \mathbf{E}_t + \mathbf{E}_t \mathbf{E}_s = \mathbf{0}$. □

Proposition 4. $\mathbf{E_s E_s} = \sigma_s \mathbf{I}$

Proof. Consider the matrix $\mathbf{W} = \mathbf{GA_s GA_s}$. Then $w_{\mu\nu} = \sum_\lambda \sigma_\mu \sigma_\lambda a_{\mu\lambda} a_{\lambda\nu}$ element-wise. By Lemma 1 $\mathbf{W}$ is sparse so that $w_{\mu\nu} = (0; \sigma_\mu \sigma_q a_{\mu q} a_{q\nu})$.

From the structure of $\mathbf{M}$ for the entries containing the element e_S we have the equivalence

$$\begin{cases} e_M e_Q = a^s_{\mu q} e_S, & S = M \triangle Q \\ e_Q e_M = a^s_{q\mu} e_S, \end{cases}$$

After multiplication of the equations we obtain $e_M e_Q e_Q e_M = a^s_{\mu q} e_S a^s_{q\mu} e_S$, which simplifies to the *First fundamental identity*:

$$\sigma_q \sigma_\mu = a^s_{\mu q} a^s_{q\mu} \sigma_s \tag{7}$$

We observe that if $\sigma_\mu = 0$ or $\sigma_q = 0$ the result follows trivially. In this case also $\sigma_s = 0$. Therefore, let's suppose that $\sigma_s \sigma_q \sigma_\mu \neq 0$. We multiply both sides by $\sigma_s \sigma_q \sigma_\mu$ to obtain $\sigma_s = \sigma_q \sigma_\mu a^s_{\mu q} a^s_{q\mu}$. However, the RHS is a diagonal element of $\mathbf{W}$, therefore by the sparsity it is the only non-zero element for a given row/column so that $\mathbf{W} = \mathbf{E_s^2} = \sigma_s \mathbf{I}$. □

Definition 5 (Clifford coefficient map)**.** *Define the linear map acting element-wise $C_a : C\ell_n \mapsto \mathbb{R}$ by the action $C_a(ax + b) = x$ for $x \in \mathbb{R}, a, b \in \mathbf{B}$.*

*Define the Clifford **coefficient map** indexed by e_S as $\mathbf{A}_S := C_S(\mathbf{M})$, where $\mathbf{M}$ is the multiplication table of the extended basis $\mathbf{M} = \{e_M e_N \mid e_M, e_N \in \mathbf{B}\}$, and $\mathbf{A}_S$ action of the map.*

Definition 6 (Canonical matrix map)**.** *Define the map $\pi : \mathbf{B} \mapsto \mathbf{Mat}_{\mathbb{R}}(2^n \times 2^n),\ n = p + q + r$ as*

$$\pi : e_S \mapsto \mathbf{E}_s := \mathbf{GA}_s \tag{8}$$

where s is the ordinal of $e_S \in \mathbf{B}$ and $\mathbf{A}_S$ is computed as in Def. 5.

Proposition 5. *The π-map is linear.*

The proposition follows from the linearity of the coefficient map and matrix multiplication with a scalar.

Theorem 3.1 (Semigroup property). *Let e_s and e_t be generators of $C\ell_{p,q,r}$. Then the following statements hold*

(1) *The map π is a homomorphism with regard to the Clifford product (i.e. π distributes over the Clifford products): $\pi(e_s e_t) = \pi(e_s)\pi(e_t)$.*
(2) *The set of all matrices $\mathbf{E}_s$ forms a multiplicative semigroup.*

Proof. Let $\mathbf{E}_s = \pi(e_s), \mathbf{E}_t = \pi(e_t), \mathbf{E}_{st} = \pi(e_s e_t)$. We specialize the result of Lemma 2 for $S = \{s\}$ and $T = \{t\}$ and observe that $m_{\lambda\lambda'}\, e_{st} = m_{\lambda\mu}\sigma_\mu m_{\mu\lambda'}\, e_{st}$ for $\lambda, \lambda', \mu \leq n$ and $\sigma_\lambda m_{\lambda\lambda'} = \sigma_\lambda m_{\lambda\mu}\sigma_\mu m_{\mu\lambda'}$. In summary, the map π acts on $C\ell_{p,q}$ according to the following diagram:

$$\begin{array}{ccc} e_s & \xrightarrow{\quad\pi\quad} & \mathbf{E}_s \\ \big\downarrow {\scriptstyle e_t} & & \big\downarrow {\scriptstyle \mathbf{E}_t} \\ e_s e_t \equiv e_{st} & \xrightarrow{\quad\pi\quad} & \mathbf{E}_{st} \equiv \mathbf{E}_s\mathbf{E}_t,\ st = s \cup t \end{array}$$

Therefore, $\mathbf{E}_{st} = \mathbf{E}_s\mathbf{E}_t$. Moreover, we observe that $\pi(e_s e_t) = \mathbf{E}_{st} = \mathbf{E}_s\mathbf{E}_t = \pi(e_s)\pi(e_t)$.

For the semi-group property observe that since π is linear it is invertible. Since π distributes over Clifford product its inverse π^{-1} distributes over matrix multiplication:

$$\pi^{-1}(\mathbf{E}_s\mathbf{E}_t) \equiv \pi^{-1}(\mathbf{E}_{st}) = e_{st} \equiv e_s e_t = \pi^{-1}(\mathbf{E}_s)\ \pi^{-1}(\mathbf{E}_t)$$

However, $C\ell_{p,q}$ is closed by construction, therefore, the set $\{\mathbf{E}\}_s$ is closed under matrix multiplication. □

Proposition 6. *Let $\mathbf{L} := \{l_i | \ l_i \in \mathbf{B}\}$ be a column vector and $\mathbf{R}_s$ be the first row of $\mathbf{E}_s$. Then $\pi^{-1} : \mathbf{E}_s \mapsto \mathbf{R}_s\mathbf{L}$.*

Proof. We observe that by the Prop. 2 the only non-zero element in the first row of $\mathbf{E}_s$ is $\sigma_1 m_{1s} = 1$. Therefore, $\mathbf{R}_s\mathbf{L} = e_s$. □

Theorem 3.2 (Complete Real Matrix Representation). *Define the map $g : \mathbf{A} \mapsto \mathbf{GA}$ as matrix multiplication with $\mathbf{G}$. Then for a fixed multiindex s $\pi = C_s \circ g = g \circ C_s$. Further, π is an isomorphism inducing a Clifford algebra representation in the real matrix algebra according to the diagram:*

$$C\ell_{p,q}(\mathbb{R}) \underset{\pi^{-1}}{\overset{\pi}{\rightleftarrows}} \mathbf{Mat}_{\mathbb{R}}\,(2^n \times 2^n)$$

Proof. The π-map is a linear isomorphism. The set $\{\mathbf{E}_s\}$ forms a multiplicative group, which is a subset of the matrix algebra $\mathbf{Mat}_{\mathbb{R}}(N \times N), N = 2^n$. Let $\pi(e_s) = \mathbf{E}_s$ and $\pi(e_t) = \mathbf{E}_t$. It is claimed that

(1) $\mathbf{E}_s\mathbf{E}_t \neq \mathbf{0}$ by the Sparsity Lemma 1.
(2) $\mathbf{E}_s\mathbf{E}_t = -\mathbf{E}_t\mathbf{E}_s$ by Prop. 3.
(3) $\mathbf{E}_s\mathbf{E}_s = \sigma_s\mathbf{I}$ by Prop. 4.

Therefore, the set $\{\mathbf{E}_S\}_{S=\{1\}}^{P(n)}$ is an image of the extended basis $\mathbf{B}$. Here $P(n)$ denotes the power set of the indices of the algebra generators. □

What is useful about the above representation is the relationship between the trace of the multivector matrix and the scalar part of the preimage

$$\mathrm{tr}\mathbf{A} = 2^n \langle A \rangle_0 \tag{9}$$

for the image $\pi(A) = \mathbf{A}$ of a general multivector element A. This will be used further in the proof of FVS algorithm.

Remark 1. *The above construction works if instead of the entire algebra $C\ell_{p,q}$ we restrict a multivector to a sub-algebra of a smaller grade* $\max \mathrm{gr}[A] = r$*. In this case, we form grade-restricted multiplication matrices $\mathbf{G}_r$ and $\mathbf{M}_r$.*

4. Characteristic and minimal polynomials of a multivector

Let us first introduce the notions of characteristic and minimal polynomials of a multivector.

Definition 7 (Characteristic polynomial)**.** *The characteristic polynomial p_A of the multivector A is the preimage of the characteristic polynomial $P_A(x) := \det(x\boldsymbol{I} - \boldsymbol{A})$ of its matrix representation by the map π.*

From the properties of the π map it is clear that

$$\pi^{-1} : P_A(\mathbf{A}) = \mathbf{0} \mapsto p_A(A) = 0 \tag{10}$$

so that the above definition is consistent with the usual notion of a characteristic polynomial. Therefore, the notion of an eigenvalue λ of a multivector can be also defined having its usual meaning – that is, a member of the list of real or complex numbers $\{\lambda\}_i$, such that the equation

$$p_A(A) = \prod_{i}^{2^n} (A - \lambda_i) = 0 \tag{11}$$

holds true for the multivector A.

Definition 8 (Minimal polynomial)**.** *The minimal polynomial m is the monic polynomial μ of minimal degree, such that*

$$\mu(A) = \sum_{k=0}^{m} c_k A^k = 0, \quad c_m = 1 \tag{12}$$

for a given multivector A.

The coefficients of the polynomial will be assumed to be real numbers although this is not strictly necessary, as discussed in [5].

Proposition 7. *The minimal polynomial μ is unique for a given multivector $A \in C\ell_{p,q}$.*

Proof. The proof is given in [5]. Suppose that f(x) and g(x) are two monic polynomials of minimal degree m such that f(y) = g(y) = 0, then h(x) = f(x) - g(x) is a polynomial of smaller degree (m-1) such that h(y) = 0. This contradicts the minimality of f and g. Therefore, h = 0. □

Proposition 8. *Under the mapping π the polynomial μ is the minimal polynomial of the complete real matrix representation of the generic multivector A.*

Proof. Consider the generic multivector A and fix the value of the coefficients of its minimal polynomial μ. Then by the properties of the π map we can compose the diagram

$$\begin{array}{ccc} \mu(A) & \xrightarrow{\ \pi\ } & \mu(\pi(A)) = \mu(\mathbf{A}) \\ \Big\downarrow{=} & & \Big\uparrow{=} \\ 0 & \xrightarrow{\ \pi\ } & \mathbf{0} \end{array}$$

which is true. ☐

Lemma 3. *Suppose that $p_A(x)$ and $\mu(x)$ are the characteristic and minimal polynomials of the multivector A, respectively; and furthermore $\mu(0) \neq 0$. Then*

- *μ divides p_A: $\mu | P_A$;*
- *p_A and μ share the same roots;*
- *Finally, p_A can be written as*

$$p_A(x) = \sum_{k=1}^{n=\lfloor N/m \rfloor} a_k \mu^k(x), \quad a_n = 1 \tag{13}$$

  where $\deg[p] = N$ *and* $m|N$.

Proof. Suppose that p_A is of degree N and is divided by μ (of degree m) as

$$p_A(x) = \mu(x) g(x) + r(x),$$

where $g(x)$ is a polynomial of degree $N - m$ and $r(x)$ is the reminder polynomial of maximal degree $k < m$ (by the definition of μ). Then we evaluate A at any of its roots to obtain $0 = r(A)$. Therefore, $r = 0$ since μ by hypothesis is the minimal polynomial.

Suppose that $\lambda \neq 0$ is a root of P_A with multiplicity 1. Then there exits a non-null eigenvector v, such that $\mathbf{A}v = \lambda v$. Furthermore, by associativity, for any natural number m we have $\mathbf{A}^m v = \lambda^m v$. Hence

$$\mu(\mathbf{A}) v = \mu(\lambda) \mathbf{I} v = \mathbf{0}$$

by Prop. 8. Therefore, by the above diagram P_A and hence p_A, and μ share the same roots.

Now suppose that λ is a root of μ with multiplicity 1. To establish the validity of eq. 13 we write it first as

$$p_A(x) = \sum_{k=0}^{n=\lfloor N/m \rfloor} a_k \mu^k(x) + r(x),$$

where r is the reminder term. However, we have already established that $h = 0$. Then we use a result on the condition when one polynomial is a polynomial of another one, as stated in [13, Prop. 1] since p_A and μ share the same roots as established above they do fulfil the technical condition. Furthermore, we observe that $a_n = 1$ since p_A is monic. To determine n we observe that the coefficients can be determined by the analytical formula

$$a_k = \frac{1}{k!}\left(\frac{1}{\mu'(x)}\frac{\partial}{\partial x}\right)^k p_A(x)\bigg|_{x=\lambda} \tag{14}$$

Therefore, at the final step $nm = \deg[p] = N$. The series terminates for $k > n$. The proof of eq. 14 follows by induction observing that for a differentiable function μ we have that

$$\frac{\partial}{\partial \mu} p_A = \frac{dx}{d\mu}\frac{\partial}{\partial x} p_A = \frac{1}{\mu'(x)}\frac{\partial}{\partial x} p_A$$

while also $\mu'(\lambda) \neq 0$. □

To optimize the inverse calculation the following needs to be considered. On the first place, for the case whenever $p_A(x) = \mu(x)^n$ one could determine the coefficients of μ by equating the equal powers from both sides of the equation. The exponent n in the formula can be determined by the polynomial Greatest Common Divisor algorithm. The algorithm how to compute μ from p_A for this case is presented in Listing 1.

On the other hand, if $\mu(0) = 0$ then $\mu(x) = xh(x)$, where $h(x)$ corresponds to a zero divisor, since $\det A = 0$ in that case. Suppose that $h(0) \neq 0$. In such a case we proceed as follows. Write p_A as

$$p_A(x) = \sum_{k=0}^{n=\lfloor N/m \rfloor} a_k h^k(x) \tag{15}$$

Then by the chain rule we obtain

$$\frac{\partial}{\partial h} p_A = \frac{dx}{dh}\frac{\partial}{\partial x} p_A = \frac{1}{h'(x)}\frac{\partial}{\partial x} p_A$$

Therefore,

$$a_k = \frac{1}{k!}\left(\frac{1}{h'(x)}\frac{\partial}{\partial x}\right)^k p_A(x)\bigg|_{x=\lambda} \tag{16}$$

by induction, in a similar way as above. The above discussion is valid also for the case when $\mu(x) = x^q h(x)$ for some natural number q. In such case h is computed in corresponding manner as $h(x) = \mu(x)/x^q$, so we only need to determine the multiplicity of the root $x = 0$ by successive differentiations.

From this discussion it is apparent that in the general case the minimal polynomial can not be determined solely from the characteristic one. However, this is not interesting for applications, since in the case $\det A = 0$ the inverse will not exist.

5. Multivector inverses and related notions

5.1. **Low dimensional formulas for the inverse.** The following formulas for the inverse element have been shown to hold[7]: For n=1,2

$$M^{-1} = \frac{\overline{M}}{M\overline{M}} \tag{17}$$

For $n = 3$

$$M^{-1} = \frac{\overline{M}\widehat{M}M^{\sim}}{M\overline{M}\widehat{M}M^{\sim}} \tag{18}$$

For $n = 4$

$$M^{-1} = \frac{\overline{M} h_{3,4}(M\overline{M})}{M\overline{M} h_{3,4}(M\overline{M})} \tag{19}$$

For $n = 5$

$$M^{-1} = \frac{\overline{M}\widehat{M}M^{\sim} h_{1,4}(M\overline{M}\widehat{M}M^{\sim})}{M\overline{M}\widehat{M}M^{\sim} h_{1,4}(M\overline{M}\widehat{M}M^{\sim})} \tag{20}$$

Other, but equivalent formulas have been derived by different authors [8, 14, 3].

5.2. **The FVS multivector inversion algorithm.** Multivector inverses can be computed using the matrix representation and the characteristic polynomial.

The matrix inverse is given as $\mathbf{A}^{-1} = \operatorname{adj}\mathbf{A}/\det\mathbf{A}$, where $\det\mathbf{A}$ is the determinant and adj denotes the adjunct. The formula is not practical, because it requires the computation of $n^2 + 1$ determinants. By Cayley-Hamilton's Theorem, the inverse of $\mathbf{A}$ is a polynomial in $\mathbf{A}$, which can be computed at the last step of the FVS algorithm [4]. This algorithm has a direct representation in terms of Clifford multiplications as follows.

Theorem 5.1 (Reduced-grade FVS algorithm)**.** *Suppose that $A \in C\ell_{p,q}$ is a multivector of span s, such that $A \subseteq \operatorname{span}[e_1, \ldots, e_s]$. The Clifford inverse, if it exists, can be computed in $k = 2^{\lceil s/2 \rceil}$ Clifford multiplication steps as*

$$\begin{array}{l|ll} m_1 = A & c_1 = -kA * 1, & t_1 := -c_1 \\ m_2 = Am_1 - t_1 & c_2 = -\frac{k}{2} A * m_1, & t_2 := -c_2 \\ \ldots & \ldots & \\ m_k = Am_{k-1} - t_k & c_k = -A * m_{k-1}, & t_k := -c_k \end{array}$$

until the step where $m_k = 0$ so that

$$A^{-1} = -m_{k-1}/c_k. \tag{21}$$

The inverse does not exist if $c_k = -\det A = 0$.

There is a polynomial of A of maximal grade k

$$\chi_A(\lambda) = \lambda^k + c_1\lambda^{k-1} + \ldots c_{k-1}\lambda + c_k, \tag{22}$$

such that $\chi_A(A) = 0$. This polynomial will be called reduced characteristic polynomial.

Proof. The proof follows from the homomorphism of the π map. We recall the statement of FVS algorithm:

$$p_A(\lambda) = \det(\lambda \mathbf{I}_n - \mathbf{A}) = \lambda^n + c_1\lambda^{n-1} + \ldots c_{n-1}\lambda + c_n, \quad n = \dim(\mathbf{A}),$$

where

$$\begin{array}{l|ll} \mathbf{M}_1 = \mathbf{A}, & t_1 = \ \operatorname{tr}[\mathbf{M}_1], & c_1 = -t_1 \\ \mathbf{M}_2 = \mathbf{A}\mathbf{M}_1 - t_1\mathbf{I}_n, & t_2 = \frac{1}{2}\operatorname{tr}[\mathbf{A}\mathbf{M}_1], & c_2 = -t_2 \\ \ldots & \ldots & \ldots \\ \mathbf{M}_n = \mathbf{A}\mathbf{M}_{n-1} - t_n\mathbf{I}_n, & t_n = \frac{1}{n}\operatorname{tr}[\mathbf{A}\mathbf{M}_{n-1}], & c_n = -t_n. \end{array}$$

The matrix inverse can be computed from the last step of the algorithm as $\mathbf{A}^{-1} = \mathbf{M}_{n-1}/t_n$ under the obvious restriction $t_n \neq 0$.

Therefore, for the k$^{\text{th}}$ step of the algorithm application of π^{-1} leads to

$$\pi^{-1} : \mathbf{M}_k = \mathbf{A}\mathbf{M}_{k-1} - t_k\mathbf{I} \mapsto m_k = Am_{k-1} - t_k.$$

Furthermore, $\mathrm{tr}[\mathbf{M}_k] = n \langle m_k \rangle_0 = t_k$ by eq. 9. Moreover, the FVS algorithm terminates with $\mathbf{M}_n = \mathbf{0}_n$, which corresponds to the limiting case $n = 2^{p+q}$ wherever A contains all grades. Here $\mathbf{0}_n$ denotes the square zero matrix of dimension n.

On the other hand, examining the matrix representations of different Clifford algebras, Acus and Dargys [2] make the observation that according to the Bott periodicity the number of steps can be reduced to $2^{\lceil n/2 \rceil}$. This can be proven as follows. Consider the isomorphism $C\ell_{p,q} \supset C\ell^+_{p,q} \cong C\ell_{q-1,p-1}$. Therefore, if a property holds for an algebra of dimension $n = p+q$ it will hold also for the algebra of dimension $n-2$. Therefore, suppose that for n even the characteristic polynomial is square free: $p_A(v) \neq q(v)^2$ for some polynomial. We proceed by reduction. For $n = 2$ in $C\ell_{2,0}$ and $A = a_1 + e_1 a_2 + e_2 a_3 + e_{12} a_4$ we compute

$$p_A(v) = \left(a_1^2 - a_2^2 - a_3^2 + a_4^2 - 2a_1 v + v^2\right)^2$$

and a similar result holds also for the other signatures of $C\ell_2$. Therefore, we have a contradiction and the reduced polynomial is of degree $k = 2^{n/2}$ and the number of steps can be reduced accordingly. In the same way, suppose that n is odd and the characteristic polynomial is square-free. However, for $n = 3$ in $C\ell_{3,0}$ and $A = a_1 + e_1 a_2 + e_2 a_3 + e_3 a_4 + a_5 e_{12} + a_6 e_{13} + a_7 e_{23} + a_8 e_{123}$ it is established that p_A factorizes as $p_A(v) = q(v)^2$ for $q(v) =$

$$(a_1^2 - a_2^2 - a_3^2 - a_4^2 + a_5^2 + a_6^2 + a_7^2 - a_8^2 + 2i(a_3 a_6 - a_4 a_5 - a_2 a_7 + a_1 a_8) - 2(a_1 + i a_8) v + v^2)$$
$$(a_1^2 - a_2^2 - a_3^2 - a_4^2 + a_5^2 + a_6^2 + a_7^2 - a_8^2 + 2i(a_4 a_5 - a_3 a_6 + a_2 a_7 - a_1 a_8) - 2(a_1 - i a_8) v + v^2).$$

The above polynomial is factored due to space limitations. Similar results hold also for the other signatures of $C\ell_3$. Therefore, we have a contradiction and the reduced polynomial is of degree $k = (n+1)/2$. Therefore, overall, one can reduce the number of steps to $k = 2^{\lceil n/2 \rceil}$.

As a second case, let $E_s = \mathrm{span}[A]$ be the set of all generators, represented in A and s their number. We compute the restricted multiplication tables $\mathbf{M}(E_s)$ and respectively $\mathbf{G}(E_s)$ and form the restricted map π_s. Then

$$\pi_s(AA^{-1}) = \pi_s(A)\pi_s(A^{-1}) = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_n, \quad n = 2^s.$$

Therefore, the FVS algorithm terminates in $k = 2^s$ steps. Observe that π^{-1} : $\mathbf{AM}_k \mapsto A m_k$. Therefore, $\mathrm{tr}[\mathbf{AM}_k]$ will map to $2^s A * m_k$ by eq. 9. Now, suppose that $t_k \neq 0$; then for the last step of the algorithm we obtain:

$$A m_{k-1} - t_k = 0 \Rightarrow A m_{k-1}/t_k = 1 \Rightarrow A^{-1} = m_{k-1}/t_k.$$

Therefore, by the argument of the previous case, the number of steps can be reduced to $k = 2^{\lceil s/2 \rceil}$. □

Corollary 1. *The adjunct of a multivector A can be computed as*

$$\mathrm{adj}\, A = m_{k-1}, \quad k = 2^{\lceil s/2 \rceil}$$

Corollary 2. *χ_A is the minimal polynomial of the generic multivector A of span s. The maximal grade of the μ is $m = 2^{\lceil s/2 \rceil}$. The algebra signature determines uniquely χ_A.*

Proof. The first statement follows from Prop. 8. The second statement follows from Prop. 7. □

Remark 2. *To avoid possible confusion the name "reduced characteristic polynomial" will be kept for the minimal polynomial of the algebra.*

6. Multivector rank

The above proof demonstrates that the degree of the minimal polynomial determines the number of steps (i.e. Clifford multiplications) in the computation. If the degree of the minimal polynomial is smaller than the degree of the characteristic polynomial some optimization of the algorithm is possible but then we have to determine the minimal polynomial of a specific, possibly sparser multivector. To do so one could use the following result.

Proposition 9. *Suppose that μ is of degree m. Then the multivector FVS algorithm will terminate in m steps.*

Proof. By Lemma eq. 3 $p_A(x)$ is a polynomial in μ. Since in $C\ell_{p,q}$ there are no nilpotents

$$p_A(A) = 0 \Longrightarrow \mu(A) = 0 \Longrightarrow \pi[\mu(A)] = \mu(\pi(A)) = \mu(\mathbf{A}) = \mathbf{0}$$

So we obtain a matrix polynomial. On the other hand, similarly to the method of Horner we calculate

$$\mathbf{0} = \mathbf{B}_m = \mathbf{A}(\mathbf{B}_{m-1} - p_{m-1}\mathbf{I}) = \mathbf{A}(\mathbf{A}(\mathbf{B}_{m-2} - p_{m-2}\mathbf{I}) - p_{m-1}\mathbf{I}) = \\ \ldots = \mathbf{A}(\mathbf{A}^m \ldots - p_{m-2}\mathbf{A} - p_{m-1}\mathbf{I}) = \mathbf{A}\mu(\mathbf{A}).$$

Therefore, we recover the structure of the FVS matrix algorithm with the identification $p_k = -c_{m-k-1}$.

□

Based on Th. 5.1 we can tabulate the numbers for steps necessary for the determinant computation (Table 1) in view of the algebra dimension. The table can be extended in an obvious way for the higher dimensional Clifford algebras. However, here it is truncated to $n = 8$ considering the Bott periodicity.

dimension	s	1	2	3	4	5	6	7	8
maximal number of steps	$2^{\lceil s/2 \rceil}$	2	2	4	4	8	8	16	16

Table 1. Number of steps of the reduced-grade FVS algorithm

Proposition 10 (Rank algorithm)**.** *Consider the multivector A, such that* $\det A \neq 0$*, having span $E_s = \mathrm{span}[A] = \{e_1, \ldots, e_s\}$ of s dimensions. Consider the elements of the Krylov sequence*

$$\mathcal{W} = \{1, A, A^2, \ldots A^k\}, \quad k = 2^{\lceil s/2 \rceil}$$

Populate a matrix $\mathbf{W}$ by the action, generating its k-th row

$$W_{(k)} = \{e_J^{-1} * (A^k)\}, \quad e_J \in P(E_s)$$

Then the rank of $\mathbf{W}$ is equal to the degree of $\mu(A)$.

Proof.

$$\mathbf{W}_{(k)} = \mathbf{G}^{-k}\mathbf{A}^k\mathbf{1}^T = \{e_J^{-1} * (A^k)\}, \quad e_J \in P(E_s)$$

Therefore, $\mathrm{rank}(\mathbf{W})$ is equal to the degree of $\mu(\mathbf{A})$ by definition from where the result follows. □

This allows us to define the rank of a multivector in the following way:

Definition 9 (Rank of a multivector). *The rank of the multivector A, $r(A)$, is the degree of its minimal polynomial $\mu(A)$.*

Therefore, we claim that

Proposition 11. *The determinant of the multivector A (and hence its inverse if it exists) can be computed in at least $r(A)$ number of steps.*

Remark 3. *If the matrix rank is determined by direct computation the above result stated in Prop. 10 may not be very practical. On the other hand, the computation of the set $\mathcal{W}$ can be parallelized which can lead to some time saving. For example this could be done by adapting the Samuleson-Berkowitz algorithm. On the second place, there maybe more economic algorithms for determining the rank of A.*

From the above results we can conclude that the rank of a multivector is a measure of its complexity. Once could expect that the multivector rank will have impact also on other algorithms.

7. Experiments

Computations are performed using the *Clifford* package in Maxima, which was first demonstrated in [12]. The present version of the package is 2.5 and it is available for download from a Zenodo repository [10]. The function fadlevicg2cp returns the inverse (if it exists) and the characteristic polynomial $p_A(v)$ of a multivector A (Appendix A). Experiments were performed on a Dell® 64-bit Microsoft Windows 10 Enterprise machine with configuration – Intel® Core™ i5-8350U CPU @ 1.70GHz, 1.90 GHz and 16GB RAM. The computations were performed using the Clifford package version 2.5 on Maxima version 5.46.0 using Steel Bank Common Lisp version 2.2.2.

7.1. Symbolical experiments.

Example 1. *For $C\ell_{2,0}$ and a multivector $A = a_0 + a_1 e_1 + a_2 e_2 + a_3 e_{12}$ the reduced grade algorithm produces*

$$t_1 = -2a_1, \quad m_1 = a_1 + e_1 a_2 + e_2 a_3 + a_4 e_{12},$$

resulting in $A^{-1} = (a_1 - e_1 a_2 - e_2 a_3 - a_4 e_{12})/(a_1^2 - a_2^2 - a_3^2 + a_4^2)$ and the reduced characteristic polynomial is $\chi_A(v) = a_1^2 - a_2^2 - a_3^2 + a_4^2 - 2a_1 v + v^2$.

Example 2. *For $C\ell_{1,1}$ and a multivector$A = a_0 + a_1 e_1 + a_2 e_2 + a_3 e_{12}$ the reduced grade algorithm produces*

$$t_1 = -2a_1, \quad m_1 = a_1 + e_1 a_2 + e_2 a_3 + a_4 e_{12},$$

resulting in $A^{-1} = (-a_1 + e_1 a_2 + e_2 a_3 + a_4 e_{12})/(-a_1^2 + a_2^2 - a_3^2 + a_4^2)$ and the reduced characteristic polynomial is $\chi_A(v) = a_1^2 - a_2^2 + a_3^2 - a_4^2 - 2a_1 v + v^2$.

Example 3. *For $C\ell_{0,2}$ and a multivector $A = a_0 + a_1 e_1 + a_2 e_2 + a_3 e_{12}$ the reduced grade algorithm produces*

$$t_1 = -2a_1, \quad m_1 = a_1 + e_1 a_2 + e_2 a_3 + a_4 e_{12},$$

resulting in $A^{-1} = (a_1 - e_1 a_2 - e_2 a_3 - a_4 e_{12})/(a_1^2 + a_2^2 + a_3^2 + a_4^2)$ and the reduced characteristic polynomial is $\chi_A(v) = a_1^2 + a_2^2 + a_3^2 + a_4^2 - 2a_1 v + v^2$.

Bespoke computations are practically instantaneous on the testing hardware configuration.

Example 4. *The real matrix representation of a generic multivector A in the quaternion algebra $C\ell_{0,2}$ is*

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ -a_2 & a_1 & -a_4 & a_3 \\ -a_3 & a_4 & a_1 & -a_2 \\ -a_4 & -a_3 & a_2 & a_1 \end{pmatrix}$$

Suppose that we wish to determine if a particular Hadamard 4×4 matrix (discussed for example in [9]*)*

$$A_q = \begin{pmatrix} q_1 & q_2 & -q_3 & q_4 \\ -q_2 & q_1 & q_4 & q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ -q_4 & -q_3 & -q_2 & q_1 \end{pmatrix}$$

encodes the same algebra. One could proceed as follows. Applying the matrix FVS algorithm to compute the characteristic polynomial of A_q, we obtain

$$\chi_{A_q}(v) = \left(q_1^2 + q_2^2 + q_3^2 + q_4^2 - 2q_1 v + v^2\right)^2$$

On the other hand, the reduced multivector FVS algorithm applied to A results in the polynomial

$$\chi_A(v) = a_1^2 + a_2^2 + a_3^2 + a_4^2 - 2a_1 v + v^2$$

Therefore, by simple inspection one could conclude that A_q encodes the same Clifford (i.e. quaternion) algebra, which was also recognized in [9] *using the full multiplication table of the matrix algebra. The identification using the multiplication table requires 16 matrix multiplications, while the multivector FVS algorithm only 2 multivector multiplications as shown above.*

Example 5. *Consider $C\ell_{3,0}$. Let*

$$A = a_1 + e_1 a_2 + e_2 a_3 + e_3 a_4 + a_5 (e_1 e_2) + a_6 (e_1 e_3) + a_7 (e_2 e_3) + a_8 (e_1 e_2 e_3)$$

Then the application of the FVS algorithm yields

$$A^{-1} = \frac{S + V + BV + Q}{\Delta}$$

where the determinant is given by

$$\begin{aligned}\Delta = {}& a_1^4 - 2a_1^2 a_2^2 + a_2^4 - 2a_1^2 a_3^2 + 2a_2^2 a_3^2 + a_3^4 \\ & -2a_1^2 a_4^2 + 2a_2^2 a_4^2 + 2a_3^2 a_4^2 + a_4^4 + 2a_1^2 a_5^2 - 2a_2^2 a_5^2 - 2a_3^2 a_5^2 + 2a_4^2 a_5^2 + a_5^4 - 8a_3 a_4 a_5 a_6 + 2a_1^2 \\ & a_6^2 - 2a_2^2 a_6^2 + 2a_3^2 a_6^2 - 2a_4^2 a_6^2 + 2a_5^2 a_6^2 + a_6^4 + 8a_2 a_4 a_5 a_7 - 8a_2 a_3 a_6 a_7 + 2a_1^2 a_7^2 + 2a_2^2 a_7^2 \\ & - 2a_3^2 a_7^2 - 2a_4^2 a_7^2 + 2a_5^2 a_7^2 + 2a_6^2 a_7^2 + a_7^4 - 8a_1 a_4 a_5 a_8 + 8a_1 a_3 a_6 a_8 - 8a_1 a_2 a_7 a_8 \\ & + 2a_1^2 a_8^2 + 2a_2^2 a_8^2 + 2a_3^2 a_8^2 + 2a_4^2 a_8^2 - 2a_5^2 a_8^2 - 2a_6^2 a_8^2 - 2a_7^2 a_8^2 + a_8^4\end{aligned}$$

and

$$S = \left(a_1^3 - a_1 a_2^2 - a_1 a_3^2 - a_1 a_4^2 + a_1 a_5^2 + a_1 a_6^2 + a_1 a_7^2 - 2a_4 a_5 a_8 + 2a_3 a_6 a_8 - 2a_2 a_7 a_8 + a_1 a_8^2\right)$$

While the vector part is given by

$$V = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$
$$\begin{pmatrix} -a_1^2 a_2 + a_2^3 + a_2 a_3^2 + a_2 a_4^2 - a_2 a_5^2 - a_2 a_6^2 + 2a_4 a_5 a_7 - 2a_3 a_6 a_7 + a_2 a_7^2 - 2a_1 a_7 a_8 + a_2 a_8^2 \\ -a_1^2 a_3 + a_2^2 a_3 + a_3^3 + a_3 a_4^2 - a_3 a_5^2 - 2a_4 a_5 a_6 + a_3 a_6^2 - 2a_2 a_6 a_7 - a_3 a_7^2 + 2a_1 a_6 a_8 + a_3 a_8^2 \\ -a_1^2 a_4 + a_2^2 a_4 + a_3^2 a_4 + a_4^3 + a_4 a_5^2 - 2a_3 a_5 a_6 - a_4 a_6^2 + 2a_2 a_5 a_7 - a_4 a_7^2 - 2a_1 a_5 a_8 + a_4 a_8^2 \end{pmatrix} \tag{23}$$

the bi-vector part by

$$BV = \begin{pmatrix} e_{12} \\ e_{23} \\ e_{13} \end{pmatrix}$$
$$\begin{pmatrix} -a_1^2 a_5 + a_2^2 a_5 + a_3^2 a_5 - a_4^2 a_5 - a_5^3 + 2a_3 a_4 a_6 - a_5 a_6^2 - 2a_2 a_4 a_7 - a_5 a_7^2 + 2a_1 a_4 a_8 + a_5 a_8^2 \\ 2a_3 a_4 a_5 - a_1^2 a_6 + a_2^2 a_6 - a_3^2 a_6 + a_4^2 a_6 - a_5^2 a_6 - a_6^3 + 2a_2 a_3 a_7 - a_6 a_7^2 - 2a_1 a_3 a_8 + a_6 a_8^2 \\ -2a_2 a_4 a_5 + 2a_2 a_3 a_6 - a_1^2 a_7 - a_2^2 a_7 + a_3^2 a_7 + a_4^2 a_7 - a_5^2 a_7 - a_6^2 a_7 - a_7^3 + 2a_1 a_2 a_8 + a_7 a_8^2 \end{pmatrix} \tag{24}$$

and the pseudoscalar part by

$$Q = I\left(2a_1 a_4 a_5 - 2a_1 a_3 a_6 + 2a_1 a_2 a_7 - a_1^2 a_8 - a_2^2 a_8 - a_3^2 a_8 - a_4^2 a_8 + a_5^2 a_8 + a_6^2 a_8 + a_7^2 a_8 - a_8^3\right)$$

The inverse exists if the determinant $\Delta \neq 0$.

Up to sign permutations the above results hold also for $C\ell_{2,1}$, $C\ell_{1,2}$, and $C\ell_{0,3}$ but are not given in view of space limitations.

7.2. **Numerical experiments.** Note that the trivial last steps will be omitted because of space limitations. To demonstrate the utility of FVS algorithm here follow some high-dimensional numerical examples. Examples for higher dimensional algebras are not particularly instructive as they result in very long expressions. These, can be nevertheless useful for hardcoding formulas in particular niche applications.

Example 6. *In $C\ell_{2,2}$ let*

$$A = 1 + e_1 + e_{134} - 2\,(e_2\, e_3)$$

Let $B = e_{134}$, $C = e_{123}$

$$\begin{aligned} t_1 &= -4, \quad m_1 = 1 + e_1 + B - 2e_{23} \\ t_2 &= -2, \quad m_2 = 1 - 2e_1 - 4C - 2B + 4e_{23} + 2e_{34} \\ t_3 &= 12, \quad m_3 = -9 + 3e_1 + 4C - B + 2e_{23} - 2e_{34} \end{aligned}$$

so that

$$A^{-1} = 1 + e_1 + \frac{4}{3}C - \frac{1}{3}B + \frac{2}{3}e_{23} - \frac{2}{3}e_{34}$$

The reduced characteristic polynomial is

$$\chi_A(v) = -3 + 12v - 2v^2 - 4v^3 + v^4$$

and is also minimal.

Example 7. *Let us compute a rational example in* $C\ell_{2,5}$ *. Let* $A = 1 - 2B + 5C$, *where* $B := e_{15}$ *and* $C := e_1e_3e_4$. *Then* $\mathrm{span}[A] = \{e_1, e_3, e_4, e_5\}$ *and for the maximal representation we have* $k = 2^4 = 16$ *steps:*

$$
\begin{array}{ll}
t_1 = -16, & m_1 = -15 + 5C - 2B; \\
t_2 = 288, & m_2 = 252 - 70C + 28B; \\
t_3 = -2912, & m_3 = -2366 + 1190C - 476B; \\
t_4 = 29456, & m_4 = 22092 - 10640C + 4256B; \\
t_5 = -213696, & m_5 = -146916 + 99820C - 39928B; \\
t_6 = 1509760, & m_6 = 943600 - 634760C + 253904B; \\
t_7 = -8250496, & m_7 = -4640904 + 4083240C - 1633296B; \\
t_8 = 43581024, & m_8 = 21790512 - 19121280C + 7648512B; \\
t_9 = -181510912, & m_9 = -79411024 + 89831280C - 35932512B; \\
t_{10} = 730723840, & m_{10} = 274021440 - 307223840C + 122889536B; \\
t_{11} = -2275435008, & m_{11} = -711073440 + 1062883360C - 425153344B; \\
t_{12} = 6900244736, & m_{12} = 1725061184 - 2492483840C + 996993536B; \\
t_{13} = -15007376384, & m_{13} = -2813883072 + 6132822080C - 2453128832B; \\
t_{14} = 32653412352, & m_{14} = 4081676544 - 7936593280C + 3174637312B; \\
t_{15} = -39909726208, & m_{15} = -2494357888 + 12471789440C - 4988715776B.
\end{array}
$$

Therefore, $A^{-1} = (1 - 5C + 2B)/22$ *and* $p_A(v) = (22 - 2v + v^2)^8$. *Evaluation takes 0.0469 s using 12.029 MB memory on Maxima. On the other hand, the reduced algorithm will run in* $k = 2^{\lceil 4/2 \rceil} = 4$ *steps:*

$$
\begin{array}{ll}
t_1 = -4, & m_1 = 1 + 5C - 2B; \\
t_2 = 48, & m_2 = -24 - 10C + 4B; \\
t_3 = -88, & m_3 = 66 + 110C - 44B;
\end{array}
$$

and $\chi_A(v) = 484 - 88v + 48v^2 - 4v^3 + v^4 = (22 - 2v + v^2)^2$. *Evaluation takes 0.0156 s using 2.512 MB memory on Maxima. Note, that in this case* $\det A = AA^{\sim} = 22$. *Therefore, in accordance with Shirokov's approach* $A^{-1} = A^{\sim}/22$.

Example 8. *The example was presented in* [1]. *In* $C\ell_{5,0}$ *consider the multivector* $A = 1 + 2e_1 + 3e_{23} + 4e_{2345}$. *The reduced algorithm will run in* $k = 2^3 = 8$ *steps. Let* $B = e_{2345}$, $C = e_{123}$ *and* $D = e_{145}$.

$$
\begin{array}{ll}
t_1 = -8 & m_1 = 1 + 2e_1 + 3e_{23} + 4B, \\
t_2 = -16 & m_2 = 4 - 12e_1 + 12C + 16I - 18e_{23} - 24B - 24e_{45}, \\
t_3 = 208 & m_3 = -78 - 8e_1 - 60C - 80I - 144D + 66e_{23} - 112B + 120e_{45}, \\
t_4 = 1064 & m_4 = -532 + 112e_1 + 624C - 768I + 576D - 144e_{23} + 608B - 96e_{45}, \\
t_5 = -5792 & m_5 = 3620 - 3768e_1 - 1632C + 2624I + 192D + 3084e_{23} + 912B - 192e_{45}, \\
t_6 = 20416 & m_6 = -15312 + 7280e_1 - 7536C - 10048I - 1536D - 5928e_{23} - 3104B - 14880e_{45}, \\
t_7 = -28608 & m_7 = 25032 - 96e_1 + 8592C + 8256I + 28992D + 53832e_{23} - 47424B + 15072e_{45}
\end{array}
$$

The inverse is

$$
A^{-1} = -\frac{1}{14790}\left(-149 - 4e_1 + 358e_{123} + 344e_{12345} + 1208e_{145} + 2243e_{23} - 1976e_{2345} + 628e_{45}\right)
$$

and the reduced characteristic polynomial is

$$
\chi_A(v) = 354960 - 28608v + 20416v^2 - 5792v^3 + 1064v^4 + 208v^5 - 16v^6 - 8v^7 + v^8
$$

which is also minimal. The reduced rank matrix is $\mathbf{W} =$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 & 0 & 4 & 0 \\ 12 & 4 & 6 & -24 & 12 & 0 & 8 & 16 \\ 34 & 56 & 162 & -72 & 36 & -144 & 16 & 48 \\ -276 & 208 & 624 & -1056 & 1104 & -576 & 32 & -128 \\ -1604 & -4168 & 6228 & -4800 & 5280 & -6720 & -4496 & -960 \\ -46608 & -27056 & 31176 & -29664 & 32112 & -34560 & -27232 & -46784 \\ -303176 & -403744 & 74232 & -141792 & 151536 & -81984 & -396224 & -313152 \\ -2918256 & -2717312 & 34944 & 585984 & -583296 & -32256 & -2660608 & -2966528 \end{pmatrix}$$

which is of rank 8.

Example 9. *Consider* $C\ell_{5,2}$ *and let* $A = 1 - e_2 + I$. *The full-grade algorithm takes 128 steps and will not be illustrated due to space limitation. The reduced grade algorithm can be illustrated as follows. Let* $C = e_{134567}$. *Then*

$$\begin{aligned}
t_1 &= -16, & m_1 &= 1 - e_2 + I; \\
t_2 &= 120, & m_2 &= -15 + 14e_2 - 14I + 2C; \\
t_3 &= -560, & m_3 &= 105 - 89e_2 + 93I - 26C; \\
t_4 &= 1836, & m_4 &= -459 + 340e_2 - 388I + 156C; \\
t_5 &= -4560, & m_5 &= 1425 - 881e_2 + 1145I - 572C; \\
t_6 &= 9064, & m_6 &= -3399 + 1682e_2 - 2562I + 1454C; \\
t_7 &= -14960, & m_7 &= 6545 - 2529e_2 + 4557I - 2790C; \\
t_8 &= 20886, & m_8 &= -10443 + 3096e_2 - 6648I + 4296C; \\
t_9 &= -24880, & m_9 &= 13995 - 3051e_2 + 8091I - 5448C; \\
t_{10} &= 25480, & m_{10} &= -15925 + 2386e_2 - 8242I + 5694C; \\
t_{11} &= -22416, & m_{11} &= 15411 - 1475e_2 + 7007I - 4934C; \\
t_{12} &= 16716, & m_{12} &= -12537 + 596e_2 - 4932I + 3548C; \\
t_{13} &= -10480, & m_{13} &= 8515 - 35e_2 + 2795I - 1980C; \\
t_{14} &= 5400, & m_{14} &= -4725 - 50e_2 - 1150I + 850C; \\
t_{15} &= -2000, & m_{15} &= 1875 + 125e_2 + 375I - 250C,
\end{aligned}$$

resulting in $A^{-1} = (1 - e_2 - 3\,I + 2\,C)/5$. *The reduced characteristic polynomial can factorize as*

$$\chi_A(v) = (5 - 4v + 6v^2 - 4v^3 + v^4)^2 = (1 + v^2)^4 (5 - 4v + v^2)^4 \tag{25}$$

This is an indication that the rank of the multivector is lower as will be demonstrated below.

Example 10. *We use the same data $A = 1 - e_2 + I$ in $C\ell_{5,2}$ to compute the rank according to Prop. 10. The reduced (with zero columns removed) rank matrix is*

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 \\ 1 & -2 & 2 & -2 \\ 1 & -1 & 6 & -5 \\ -3 & 4 & 12 & -12 \\ -19 & 19 & 20 & -21 \\ -59 & 58 & 22 & -22 \\ -139 & 139 & -14 & 15 \\ -263 & 264 & -168 & 168 \\ -359 & 359 & -600 & 599 \\ -119 & 118 & -1558 & 1558 \\ 1321 & -1321 & -3234 & 3235 \\ 5877 & -5876 & -5148 & 5148 \\ 16901 & -16901 & -4420 & 4419 \\ 38221 & -38222 & 8062 & -8062 \\ 68381 & -68381 & 54346 & -54345 \\ 82417 & -82416 & 177072 & -177072 \end{pmatrix}$$

It can be triangularized to the matrix

$$\mathbf{W}' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

From the structure of the matrix it can be seen that it has rank 4. The determinant $\det A$ *can be computed by the sequence of operations $B = A\widehat{A} = 1 - 2I$, followed by* $\det A = BB^{\sim} = 5$. *This allows for writing the formula*

$$A^{-1} = \widehat{A}(A\widehat{A})^{\sim}/5$$

Example 11. *We use $A = 1 - e_2 + e_3 + e_{13456}$ in $C\ell_{5,2}$ to compute the rank according to Prop. 10. The span is a 6 dimensional vector space –* $\mathrm{span}[A] = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. *The reduced (with zero columns removed) rank matrix is*

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & -1 \\ 2 & -2 & 2 & -2 & 0 & -2 \\ 4 & -4 & 2 & -6 & 2 & -6 \\ 4 & -8 & 0 & -16 & 8 & -16 \\ -4 & -12 & -12 & -40 & 24 & -36 \\ -40 & -8 & -56 & -88 & 64 & -72 \\ -160 & 32 & -184 & -168 & 152 & -120 \\ -496 & 192 & -512 & -256 & 320 & -128 \end{pmatrix}$$

It can be triangularized to the matrix

$$\mathbf{W}' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 2 & 6 & -2 & 2 \\ 0 & 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

From the structure of the matrix it can be seen that it has rank 4.

The minimal polynomial is computed as $\mu(v) = 4 + 4v^2 - 4v^3 + v^4$ *and the inverse is*

$$A^{-1} = \frac{1}{2}\left(e_3 + e_{12456} - e_{13456} - e_{1456}\right) \tag{26}$$

In this case, the determinant can be computed by the sequence of steps

$$B = AA^{\sim}, \quad \det A = B\widehat{B} = -4$$

Therefore, the inverse can be computed by the formula

$$A^{-1} = -A^{\sim}\widehat{(AA^{\sim})}/4$$

in an obvious manner.

8. Discussion

From Table 1 we can conclude that the low dimensional formulas reported in the literature are optimal in terms of the number of Clifford multiplications. It is also apparent that looking for specific formulas of the general inverse element for higher dimensional Clifford algebras would offer little immediate insight.

The maximal matrix algebra construction exhibited in the present paper allows for systematic translation of matrix-based algorithms to Clifford algebra simultaneously allowing for their direct verification. For example, future work could focus on formulating the algorithm exhibited in Prop. 10 entirely in the language of the Clifford algebra.

The advantage of the multivector FVS algorithm is its simplicity of implementation. This can be beneficial for purely numerical applications as it involves only Clifford multiplications followed by taking scalar parts of multivectors, which can be encoded as the first member of an array. The Clifford multiplication computation can be reduced to $\mathcal{O}(N \log N)$ operations, since it involves sorting of a joined list of algebra generators. On the other hand, the FVS algorithm does not ensure optimality of the computation but only provides a certificate of existence of an inverse. Therefore, optimized algorithms can be introduced for particular applications, i.e. Space-Time Algebra $C\ell_{1,4}$, Projective Geometric Algebra $C\ell_{3,0,1}$, Conformal Geometric Algebra $C\ell_{4,1}$, etc. As a side product, the algorithm can compute the characteristic polynomial of a general multivector and, hence, its determinant also without any resort to a matrix representation. This could be used, for example, for computation of a multivector resolvent or some other analytical functions.

One of the main applications of the present algorithms could be in Finite Element Modelling where a Geometric algebra approach would improve the efficiency and accuracy of calculations by providing a more compact representation of vectors, tensors, and geometric operations. This can lead to faster and more accurate simulations of elastic deformations.

Acknowledgment

The present work is funded in part by the European Union's Horizon Europe program under grant agreement VIBraTE, grant agreement 101086815.

Appendix A. Program code

The Clifford package can be downloaded from a Zenodo repository [10]. The examples can be downloaded from a Zenodo repository and it includes the file climatrep.mac, which implements different instances of the FVS algorithm [11].

Listing 1. Minimal polynomial determination

```

minpoly(P, x):=block([dP:rat(P), md:0, %a, Q, Q1, n, k, cc:[],
sol, m, qq],
    local(%a, cc),
    P:expand(P),
    n:hipow(P,x),
    if _debug1=all then display (n),
    while gcd(dP, rat(P))#1 do (
        dP:diff(dP,x),
        md:md+1
    ),
    if md=1 then (
        print (P, "is minimal"),
    return(P)
    ),
    if _debug1=all then display (md),
    m: n/md,
    if not integerp (m) then return (false),
    Q:sum(%a[i]*v^i,i,0, m),

    Q1:expand(P-Q^(md)),

    for k:n thru 1 step -1 do (
        qq:ratcoeff(Q1, x, k),
        qq: subst(cc, qq),
        sol:solve(qq, %a[m]),
        if _debug1=all then display(qq, k, m, sol),
        if listp(sol) then
            cc: push(last(sol), cc)
            else return( [ subst(cc,Q), cc] ),
        m:m-1,
        if m <0 then return( [ subst(cc,Q), cc] )
    ),
    [ subst(cc, Q), cc]
);
```

Listing 2. FVS algorithm implementation in Maxima based on the Clifford package

```
fadlevicg2cp(A, v):=block(
[M:1, K, i:1, n, k:length(clv(A)), cq, c, ss],
    n:2^(ceiling(k/2)),
    array(c,n+1),  for r:0 thru n+1 do c[r]:1,
    A:rat(A),
    ss:c[1]*v^^n,
    while i<n and K#0 do (
        K:dotsimpc(expand (A.M)),
        cq:-n/i*scalarpart(K),
        if _debug1=all then print("t_{",i,"}=",cq,"
m_{",i,"}=",K,"\\\\"),
        if K#0 then
            M: rat(K + cq),
        c[i+1]:cq, ss:ss+c[i+1]*v^^(n-i),
        i:i+1
    ),
    K:dotsimpc(expand(A.M)),
    cq:-n/i*scalarpart(K),
    if _debug1=all then print("t_{",i,"}=",cq,"
m_{",i,"}=",K,"\\\\"),
    ss:ss+cq,
    if cq=0 then cq:1,  M:factor(-(M)/cq),
    [M, ss]
);
```

References

[1] Acus, A., Dargys, A.: The inverse of a multivector: Beyond the threshold $p + q = 5$. Adv. Appl. Clifford Algebras **28**(3) (2018). https://doi.org/10.1007/s00006-018-0885-4

[2] Acus, A., Dargys, A.: The characteristic polynomial in calculation of exponential and elementary functions in Clifford algebras. Math. Methods Appl. Sci. (2022). https://doi.org/10.22541/au.167101043.33855504/v1

[3] Dadbeh, P.: Inverse and determinant in 0 to 5 dimensional Clifford algebra (2011). https://doi.org/10.48550/ARXIV.1104.0067

[4] Faddeev, D.K., Sominskij, I.S.: Sbornik Zadatch po Vyshej Algebre. Nauka, Moscow–Leningrad (1949)

[5] Garibaldi, S.: The characteristic polynomial and determinant are not ad hoc constructions **111**(9), 761. https://doi.org/10.2307/4145188

[6] Hitzer, E., Sangwine, S.J.: Construction of multivector inverse for Clifford algebras over $2m + 1$ – dimensional vector spaces from multivector inverse for Clifford algebras over 2m-dimensional vector spaces. Adv. Appl. Clifford Algebras **29**(2) (2019). https://doi.org/10.1007/s00006-019-0942-7

[7] Hitzer, E., Sangwine, S.: Multivector and multivector matrix inverses in real Clifford algebras. Appl. Math. Comput. **311**, 375–389 (2017). https://doi.org/10.1016/j.amc.2017.05.027

[8] Lundholm, D.: Geometric (clifford) algebra and its applications (2006). https://doi.org/10.48550/ARXIV.MATH/0605280

[9] Petoukhov, S.V.: Symmetries of the genetic code, hypercomplex numbers and genetic matrices with internal complementarities **23**(3–4), 225 – 448

[10] Prodanov, D.: Clifford Maxima package v 2.5.4. https://doi.org/10.5281/ZENODO.8205828, `https://zenodo.org/record/8205828`

[11] Prodanov, D.: Examples for CGI2023. https://doi.org/10.5281/ZENODO.8207889

[12] Prodanov, D., Toth, V.T.: Sparse representations of Clifford and tensor algebras in Maxima. Adv. Appl. Clifford Algebras pp. 1–23 (2016). https://doi.org/10.1007/s00006-016-0682-x

[13] Rickards, J.: When is a polynomial a composition of other polynomials? **118**(4), 358. https://doi.org/10.4169/amer.math.monthly.118.04.358

[14] Shirokov, D.: Concepts of trace, determinant and inverse of Clifford algebra elements. Progress in Analysis **1**, 187 – 194 (2011), proceedings of the 8th congress ISAAC (ISBN 978-5-209-04582-3/hbk)

[15] Shirokov, D.S.: On computing the determinant, other characteristic polynomial coefficients, and inverse in Clifford algebras of arbitrary dimension. Comp. Appl. Math. **40**(5) (2021). https://doi.org/10.1007/s40314-021-01536-0