mathematics

MDPI

Article

Computation of Minimal Polynomials and Multivector Inverses in Non-Degenerate Clifford Algebras

Dimiter Prodanov [1,2]

[1] PAML-LN, Institute for Information and Communication Technologies (IICT), Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria; dimiter.prodanov@iict.bas.bg
[2] Neuroelectronics Research Flanders, IMEC, 3001 Leuven, Belgium

Abstract: Clifford algebras are an active area of mathematical research having numerous applications in mathematical physics and computer graphics, among many others. This paper demonstrates algorithms for the computation of characteristic polynomials, inverses, and minimal polynomials of general multivectors residing in a non-degenerate Clifford algebra of an arbitrary dimension. The characteristic polynomial and inverse computation are achieved by a translation of the classical Faddeev–LeVerrier–Souriau (FVS) algorithm in the language of Clifford algebra. The demonstrated algorithms are implemented in the Clifford package of the open source computer algebra system Maxima. Symbolic and numerical examples residing in different Clifford algebras are presented.

Keywords: multivector; characteristic polynomial; Clifford algebra; computer algebra

MSC: 15A66; 15A67, 20C99

1. Introduction

Clifford algebras provide 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 was based on the insights of Hamilton, Grassmann, and Clifford from the 19th century. After a hiatus lasting many decades, Clifford geometric algebra experienced a renaissance with the advent of contemporary computer algebra systems. There are multiple actively researched applications in computer-aided design (CAD), computer vision and robotics, protein folding, neural networks, modern differential geometry, genetics, and mathematical physics. Readers are directed towards Hitzer et al. for a recent survey on such applications [1].

Clifford algebras are implemented in a variety of general-purpose computer languages and computational platforms. There are many implementations of the major computer algebra systems, such as the package CLIFFORD for Maple [2], the Clifford Multivector Toolbox for Matlab [3], the Clifford package (v 2.5.4) for Maxima [4], and 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/), and Mathematica—the Geometric Algebra Mathematica package by Acus and Dargys (https://github.com/ArturasAcus/GeometricAlgebra).

As its main contribution, this article demonstrates an algorithm for multivector inversion, based on the Faddeev–LeVerrier–Souriau (FVS) algorithm. The algorithm is implemented in the computer algebra system *Maxima* (version 5.37 or greater) using the

check for updates

Academic Editor: Danail Brezov

Received: 13 February 2025
Revised: 20 March 2025
Accepted: 24 March 2025
Published: 27 March 2025

Citation: Prodanov, D. Computation of Minimal Polynomials and Multivector Inverses in Non-Degenerate Clifford Algebras. *Mathematics* **2025**, *13*, 1106. https://doi.org/10.3390/math13071106

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Clifford package [4,5]. The multivector FVS algorithm was presented in a preliminary form at the Computer Graphics International conferee, CGI 2023, Shanghai, 28 August–1 September 2023 [6]. Compared to the initial presentation, the current paper introduces the apparatus of minimal polynomials and the related concept of a multivector rank as central computational tools.

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 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 best of the present author's knowledge, the FVS algorithm has not been used systematically to exhibit multivector inverses.

This paper is organized as follows. Section 2 introduces the notation. Section 4 exhibits a real matrix representation of the algebra. Section 3 discussed the indicial representation of the algebra. Section 5 discusses the multivector inverse and derives the FVS multivector algorithm. Section 6 introduces the notion of rank of a multivector. Section 7 demonstrates the algorithm. Section 10 provides the Conclusion section of the paper.

2. Notation and Preliminaries

2.1. Notation

$C\ell_n$ denotes a Clifford algebra of order n but with an unspecified signature. Clifford multiplication is denoted by a 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 the 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 degree of the polynomial P will be denoted as $\deg P$.

2.2. General Definitions

Definition 1. *Let $p, q \geq 0$ with $n := p + q$ be given integers and let $C\ell_{p,q} := C\ell_{p,q}[e_1, \ldots, e_n]$ denote the non-degenerate real Clifford algebra with signature (p, q) and ordered sequence of generators $e_1, \ldots, e_n$. That is, $C\ell_{p,q}$ is a real associative algebra with unit 1 generated freely by its elements e_i which satisfy the relations*

$$e_i^2 = 1 \;\; (i \leq p), \quad e_i^2 = -1 \;\; (i > p), \quad e_i e_j = -e_j e_i \;\; (i \neq j).$$

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 $E = \{e_1, \ldots, e_n\}$ and their irreducible products.

Definition 2 (Scalar product)**.** *The scalar product of the blades A and B will be denoted by $*$ as*

$$A * B := \langle AB \rangle_0 \tag{1}$$

and extended by linearity to the entire algebra.

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 multi-index, 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.

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

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

holds true.

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

Definition 4. *The term general multivector will denote a multivector for which* $\mathrm{span}[A] = E$*.*

Definition 5 (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 recall 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 a 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,

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

These can be further composed into Clifford conjugation:

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

Readers are also directed to the works of [7–9] for more details. Another less-used automorphism is the Hitzer–Sangwine involution [10]:

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

for the multiindex J, which has no standard notation.

Lastly, the inverse (or Hermitian) automorphism is defined as follows [11]. Suppose that M is represented as the sum of blades $M = \sum_{k=0}^{n} c_k B_k$. Then,

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

by linearity. This also lacks standard notation. The term "Hermitian" originates from the use of complex-valued Clifford algebras.

2.4. Minimal Polynomials

Definition 6 (Minimal polynomial)**.** *The minimal polynomial of a multivector A is the monic polynomial $\mu : C\ell_n \mapsto C\ell_n$ of minimal degree m, such that*

$$\mu(A) = \sum_{k=0}^{m} c_k A^k = 0, \quad c_m = 1 \tag{7}$$

for a given multivector A.

Evaluation of μ in the Clifford algebra or the complex numbers will be assumed depending on the context of discussion. Furthermore, we have the following result:

Proposition 1. *The minimal polynomial μ is unique for a given multivector $A \in C\ell_{p,q}$.*

Proof. The proof is given in [12]. 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. □

3. Indicial or Sparse Representation

Definition 7 (Indicial map)**.** *Define the indicial map ι_e, acting on symbols by concatenation (i.e., of a set), such that*

$$\begin{aligned} \iota_e &: \oplus_n \mathbb{N} \mapsto \mathbf{B} \\ \iota_e &: g \mapsto e_g \end{aligned}$$

where g is set-valued and let the convention $\iota_e : \varnothing \mapsto 1_{Cl}$.

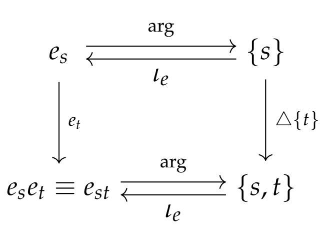
Definition 8. *Define the argument map* arg *acting on symbol compositions as*

$$\arg : \; f(g) \equiv f \circ g \mapsto g$$

Let $\arg f = \varnothing$ *for the atomic symbol f.*

These definitions allow for stating a very general result about Clifford algebra representations.

Theorem 1 (Indicial representation)**.** *For generators $e_s, e_t \in C\ell_{p,q,r}$, such that $s \neq t$, we have the following diagram:*



Proof. The right–left ι action follows from the construction of $C\ell_{p,q,r}$. The left–right argument action is trivial. We observe that $\arg f = \varnothing$. Trivially, $\{s\} \triangle \varnothing = \varnothing \triangle \{s\} = \{s\}$. Let us suppose that $s = t$. We notice that $\{s\} \triangle \{t\} = \{t\} \triangle \{s\} = \varnothing$. Let us suppose that $s \neq t$. We notice that $\{s\} \triangle \{t\} = \{s, t\}$ and $\{t\} \triangle \{s\} = \{t, s\}$. □

This theorem is used for the reduction of products of blades in the Clifford package as shown by the author of [4].

4. Clifford Algebra Real Matrix Representation Map

Definition 9 (Scalar product table)**.** *Define the diagonal scalar product matrix as*

$$\mathbf{G} := \{\sigma_{IJ} = e_I * e_J |\, e_I, e_J \in \mathbf{B}, I \prec J\} \tag{8}$$

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

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. The proof is given in [6] and will not be repeated. □

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

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

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

Proof. The proof is given in [6] but will be repeated for completeness of the discussion. 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.t. $L \prec M \prec L'$. Then, for $L \prec M \prec L'$ and $S \prec T$,

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

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

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

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 a unique index λ'' such that $m_{\lambda\lambda''} e_{st}$. Since λ is fixed, this implies that $L'' = L' \Rightarrow \lambda'' = \lambda'$. Therefore,

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

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}. □

Proposition 2. *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. The proof is given in [6]. □

Proposition 3. *For* $e_s \in \mathbf{E}$*, the matrix* $\mathbf{A}_s = C_s(\mathbf{M})$ *is sparse.*

Proof. The proof is given in [6]. □

Proposition 4. *For generator elements* e_s *and* e_t*,* $\mathbf{E}_s\mathbf{E}_t + \mathbf{E}_t\mathbf{E}_s = \mathbf{0}$*.*

Proof. The proof is given in [6]. Consider the basis elements e_s and e_t. By linearity and homomorphism of the π map (Theorem 2), we have $\pi : e_s e_t + e_t e_s = 0 \mapsto \pi(e_s e_t) + \pi(e_t e_s) = \mathbf{0}$. Therefore, for two vector elements, $\mathbf{E}_s\mathbf{E}_t + \mathbf{E}_t\mathbf{E}_s = \mathbf{0}$. □

Proposition 5. $\mathbf{E_s E_s} = \sigma_s \mathbf{I}$

Proof. The proof is given in [6]. 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*:

$$\sigma_q \sigma_\mu = a^s_{\mu q} a^s_{q\mu} \sigma_s \tag{9}$$

We observe that if $\sigma_\mu = 0$ or $\sigma_q = 0$, the result follows trivially. In this case, we also have that $\sigma_s = 0$. Therefore, let us 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 10 (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.*

In non-degenerated algebras, the coefficient map can be represented by the formula

$$\mathbf{A}_S(M) = M * e_S^{\#}$$

using the inverse automorphism.

Definition 11 (Canonical matrix map)**.** *Define the map* $\pi : \mathbf{B} \mapsto \mathbf{Mat}_{\mathbb{R}}(2^n \times 2^n)$*,* $n = p + q + r$ *as*

$$\pi : e_S \mapsto \mathbf{E}_s := \mathbf{GA}_s \tag{10}$$

where s *is the ordinal of* $e_S \in \mathbf{B}$ *and* $\mathbf{A}_S$ *is computed as in Definition 10.*

The Maxima code implementing the π-map is given in Listing A1.

Proposition 6. *The* π*-map is linear.*

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

Theorem 2 (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 respect 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. The proof is given in [6] but will be repeated for completeness of the discussion. 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 the 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 7. *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 Proposition 3, 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 (Complete Real Matrix Representation)**.** *Define the map $g : \mathbf{A} \mapsto \mathbf{GA}$ as matrix multiplication with $\mathbf{G}$. Then, for a fixed multi-index 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 following diagram:*

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

Proof. The proof is given in [6] but will be repeated for completeness of the discussion. 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 Proposition 4.
3. $\mathbf{E}_s\mathbf{E}_s = \sigma_s\mathbf{I}$ by Proposition 5.

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 pre-image:

$$\mathrm{tr}\mathbf{A} = 2^n \langle A \rangle_0 \tag{11}$$

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

Characteristic and Minimal Polynomials of Multivector

Let us first introduce the notion of a multivector characteristic polynomial and contrast it with the previously introduced definition of a minimal polynomial. The distinction between the characteristic and minimal polynomials will become clear from the subsequent results. The coefficients of these polynomials will be assumed to be real numbers, although this is not strictly necessary, as discussed in [12].

Definition 12 (Characteristic polynomial)**.** *The characteristic polynomial p_A of the multivector A is the pre-image 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

$$\pi^{-1} : P_A(\mathbf{A}) = \mathbf{0} \mapsto p_A(A) = 0 \tag{12}$$

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 also be defined according to its usual meaning—that is, a member of the list of real or complex numbers $\{\lambda\}_i$, such that the equation

$$p_A(A) = \prod_i^{2^n} (A - \lambda_i) = 0 \tag{13}$$

holds true for the multivector A.

Remark 2. *Defined in this way, the characteristic polynomial is related to the real matrix representation, while the minimal polynomial is representation-independent. Therefore, it is expected that a given matrix algorithm can be translated in a one-to-one manner to a Clifford algebra algorithm via the characteristic polynomial.*

Theorem 4. *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 following diagram:

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

□

As an illustration, consider the special case of the algebra pseudoscalar I.

Proposition 8. *Suppose that A is a blade. Then, it has a quadratic minimal polynomial of the form*

$$\mu(x) = x^2 \pm 1$$

In particular, if $A = I$, then $\mu(x) = x^2 - \sigma_I$.

Proof. Consider $C\ell_{p,q}$, with $p + q = n$. Without loss of generality, consider the pseudoscalar I_n. The square of I_n can be evaluated as

$$I_n^2 = (-1)^q (-1)^{\lceil \frac{n-1}{2} \rceil} = \sigma_I \tag{14}$$

This follows by reduction considering that

$$I_n^2 = e_n^2 (-1)^{n-1} I_{n-1}^2, \quad I_1^2 = 1$$

and there are $\lceil (n-1)/2 \rceil$ odd powers in the product. Therefore, the minimal polynomial is $\mu(x) = x^2 - \sigma_I$. By virtue of the same argument, all blades have quadratic minimal polynomials □

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

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

$$p_A(x) = g(x) \sum_{k=1}^{n=\lfloor N/m \rfloor} a_k \mu^k(x), \quad a_n = 1 \tag{15}$$

where $\deg[p] = N$ *and* $\deg[g] = N - nm$ *and $g(x)$ is monic.*

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.

Consider the matrix representation of A: $\mathbf{A} = \pi(A)$. Suppose that $\lambda \neq 0$ is a root of P_A with multiplicity 1. Then, there exists 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 Theorem 4. 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 Equation (15), we write it first as

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

where r is the reminder term. However, we have already established that $r = 0$. Then, we use a result generated on the condition when one polynomial is a polynomial of another one, as stated in (Proposition 1, [13]); since p_A and μ share the same roots as established above, they do fulfill 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

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

The series terminates for $k > n$. The proof of Equation (16) follows by induction observing that for a differentiable function μ we have that

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

while also $\mu'(\lambda) \neq 0$. □

To optimize the inverse calculation, the following needs to be considered. In 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. This is supported "out of the box" by CASs, such as Maxima.

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

$$p_A(x) = g(x) \sum_{k=0}^{n=\lfloor N/m \rfloor} a_k h^k(x) \tag{17}$$

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,

$$a_k = \frac{1}{k!}\left(\frac{1}{h'(x)}\frac{\partial}{\partial x}\right)^k \frac{p_A(x)}{g(x)}\Bigg|_{x=\lambda} \tag{18}$$

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

From this discussion, it is apparent that, in the general case, the minimal polynomial cannot be determined solely from the characteristic one.

Based on the concept of the minimal polynomial, the determinant of a multivector can be defined as follows.

Definition 13. *Consider a multivector A having a minimal polynomial μ. The determinant is defined as*

$$\det A := \mu(0) \tag{19}$$

This definition ensures the uniqueness of the determinant based on Proposition 1 and its independence of a particular representation. Furthermore, it agrees with the usual definition based on outermorphisms [14]. Indeed, consider the set of linearly independent set of vectors $V = \{v_j\}$ such that $v_i \wedge I_s = 0$ for certain blade I_s. Form the outer product

$$W(V) := \bigwedge_{j=1} v_j$$

Then, $W(V) = \alpha I_s$ for some numerical factor α. Therefore, $W I_s^{\sim} = \alpha$. Furthermore, the map W is linear by the linearity of the outer product and can trivially be extended to outermorphism as

$$W(V) := \bigwedge_{j=1} W(v_j), \quad W(v_j) = v_j$$

Therefore, we can identify $\det W = W I_s^{\sim}$. On the other hand, $W^2 = \alpha^2 I_s^2 = \pm\alpha^2$, so the minimal polynomial is $\mu(x) = x^2 \mp \alpha^2$.

5. Multivector Inverses and the FVS Algorithm

5.1. Low-Dimensional Formulas for the Inverse

The following formulas for the inverse element have been shown to hold [10]: For $n = 1, 2$,

$$M^{-1} = \frac{\overline{M}}{M\overline{M}} \tag{20}$$

For $n = 3$,

$$M^{-1} = \frac{\overline{M}\widehat{M}M^{\sim}}{M\overline{M}\widehat{M}M^{\sim}} \tag{21}$$

For $n = 4$,

$$M^{-1} = \frac{\overline{M}h_{3,4}(M\overline{M})}{M\overline{M}h_{3,4}(M\overline{M})} \tag{22}$$

For $n = 5$,

$$M^{-1} = \frac{\overline{M}\widehat{M}M^{\sim}h_{1,4}(M\overline{M}\widehat{M}M^{\sim})}{M\overline{M}\widehat{M}M^{\sim}h_{1,4}(M\overline{M}\widehat{M}M^{\sim})} \tag{23}$$

Other, albeit equivalent, formulas have been derived by different authors [15–17].

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 during the last step of the FVS algorithm [18]. This algorithm has a direct representation in terms of Clifford multiplications as follows.

Theorem 5 (Reduced-grade FVS algorithm)**.** *Suppose that $A \in C\ell_{p,q}$ is a multivector of span dimension s, such that $A \subseteq \operatorname{span}[e_1, \dots, e_s]$. The Clifford inverse, if it exists, can be computed in $k = 2^{\lceil s/2 \rceil}$ Clifford multiplication steps from the iteration*

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

until the step where $m_k = 0$ so that

$$A^{-1} = -m_{k-1}/c_k. \tag{24}$$

There exists a polynomial A of maximal grade k

$$\chi_A(\lambda) = \lambda^k + c_1\lambda^{k-1} + \dots c_{k-1}\lambda + c_k, \tag{25}$$

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

Proof. The proof is given in [6]. The proof follows from the homomorphism of the π map. We recall the following statement of tge FVS algorithm:

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

where

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

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

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

Furthermore, $\mathrm{tr}[\mathbf{M}_k] = n\langle m_k\rangle_0 = t_k$ by Equation (11). 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 [19] 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 $C\ell_{p,q} \supset C\ell^+_{p,q} \cong C\ell_{q-1,p-1}$. Then, 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 $A = a_1 + e_1a_2 + e_2a_3 + e_{12}a_4$, we compute

$$p_A(v) = \left(a_1^2 - a_2^2 - a_3^2 + a_4^2 - 2a_1v + v^2\right)^2$$

and a similar result holds also for the other signatures of $C\ell_2$ and can be obtained by direct computation using the Clifford package. 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 $A = a_1 + e_1a_2 + e_2a_3 + e_3a_4 + a_5e_{12} + a_6e_{13} + a_7e_{23} + a_8e_{123}$, it is established that p_A factorizes as $p_A(v) = q(v)^2$ for the polynomial

$$\begin{aligned} q(v) = {} & (a_1^2 - a_2^2 - a_3^2 - a_4^2 + a_5^2 + a_6^2 + a_7^2 - a_8^2 + 2i(a_3a_6 - a_4a_5 - a_2a_7 + a_1a_8) - 2(a_1 + ia_8)v + v^2) \\ & (a_1^2 - a_2^2 - a_3^2 - a_4^2 + a_5^2 + a_6^2 + a_7^2 - a_8^2 + 2i(a_4a_5 - a_3a_6 + a_2a_7 - a_1a_8) - 2(a_1 - ia_8)v + v^2). \end{aligned}$$

The above polynomial is factored in $\mathbb{C}$ due to space limitations. Similar results hold also for the other signatures of $C\ell_3$ and can be obtained by direct computation using the Clifford package. 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 count. We compute the restricted multiplication tables $\mathbf{M}(E_s)$ and $\mathbf{G}(E_s)$, respectively, and form the restricted map π_s. Then,

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

Therefore, the FVS algorithm terminates in $k = 2^s$ steps. Observe that $\pi^{-1} : \mathbf{AM}_k \mapsto Am_k$. Therefore, $\mathrm{tr}[\mathbf{AM}_k]$ will map to $2^s A * m_k$ by Equation (11). Now, suppose that $t_k \neq 0$; then, for the last step of the algorithm, we obtain

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

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

Corollary 1. *χ_A is the minimal polynomial of the generic multivector A of span s. The maximal grade of μ is $m = 2^{\lceil s/2 \rceil}$. The algebra signature uniquely determines χ_A.*

Proof. The first statement follows from Theorem 4. The second statement follows from Proposition 1. □

Corollary 2. *The inverse A^{-1} does not exist if* $\det A = 0$.

Proof. The inverse does not exist if $c_k = 0$. By Corollary 1, χ_A is the minimal polynomial and $\chi_A(0) = c_k$. □

One could define the multivector adjunct as follows:

Definition 14. *The adjunct of the generic multivector A is defined from the minimal polynomial as*

$$\mathrm{adj}\, A := -\sum_{k=1}^{m} c_k A^{k-1}$$

where c_k are the coefficients of the minimal polynomial.

Proposition 9. *Suppose that χ_A is minimal. Then, the adjunct of a multivector A can be computed as*

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

Furthermore, $A \,\mathrm{adj}\, A = \det A$.

Proof. The first part of the statement follows Theorem 5 considering that $m_k = 0$ in the last step, which corresponds to the minimal polynomial. For the second part of the statement, observe that

$$A \,\mathrm{adj}\, A = -\sum_{k=1}^{m} c_k A^k = \mu(0) - \mu(A) = \det A$$

□

Remark 3. *To avoid possible confusion, the name "reduced characteristic polynomial" will be kept for the minimal polynomial of the algebra (i.e, of the generic mutlivector of grade n).*

Based on Theorem 5, 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.

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

(sub)-space 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

6. Multivector Rank

The notion of a minimal polynomial allows one to define the rank of a multivector in a related way:

Definition 15 (Rank of a multivector)**.** *The rank of the multivector A, denoted $r(A)$, is the degree of its minimal polynomial* – deg $\mu(A)$*.*

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 achieve this, one could use the following result.

Proposition 10. *The determinant of a multivector A (and hence its inverse if it exists) can be computed in at least $r(A)$ number of steps.*

From the above definition, we can conclude that the rank of a multivector is a measure of its complexity. For instance, the scalars are of rank 1, vectors and blades are of rank 2, etc. One could expect that the multivector rank will also play a role in other algorithms.

Proposition 11. *A non-scalar multivector A is of even rank.*

Proof. Scalars obey $\mu(v) = v - 1$, so we exclude them. Suppose, then, that there exists a multivector A of odd rank $r(A) = 2s + 1$ in $C\ell_n$ for $s > 0$. We proceed by reduction as before. The property should hold for $C\ell_{n-2}$. For $n = 1$, the minimal polynomial is $\mu(v) = v^2 \pm 1$, depending on the signature of the algebra by Proposition 8. Therefore, we have a contradiction. For $n = 2$, we have general quadratic polynomials as per Examples 1–3. Therefore, again we have a contradiction. Hence, generic multivectors are of even rank. □

Denote the inverse of the blade e_J as $e^J \equiv (e_J)^\#$. Then, conveniently, $e_J * e^J = \delta^J_J$, where δ^J_J is the Kronecker symbol.

Proposition 12 (Rank algorithm)**.** *Consider a multivector A having span $E_s = \operatorname{span}[A] = \{e_1, \ldots, e_s\}$ of s dimensions. Define the Krylov exponent sequence as the set*

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

(which can be thought of also as a co-vector of multivectors). Define the trial polynomial

$$c(A) := \sum_{i=0}^{k} c_i \, A^i \equiv \mathbf{C}.\mathbf{K}$$

Populate the simultaneous equation system as $L := \{c(A) * e^J = 0\}, \quad e_J \in P(E_s)$. *Let* $\mathbf{U}$ *be the coefficient matrix of* L *with respect to* $\mathbf{C}$. *If* $k > m$, *then* $\mathrm{rank}(\mathbf{U}) \geq m$. *If* $k = m$, *then*

$$m = 2\lfloor \frac{\mathrm{rank}(\mathbf{U})}{2} \rfloor$$

and the vector $\mathbf{C}$ *spans the coefficients of* $\mu(A)$.

Proof. Suppose that $E_s = \mathrm{span}[A] = \{e_1, \ldots, e_s\}$ is of s dimensions. Consider the list of scalar products

$$c(A) * e^J = \sum_{k=0}^{n} c_k \, A^k * e_J^{\#} = 0, \quad e_J \in P(E_s)$$

enumerated by the multi-index J. They result in a simultaneous system of equations for the components of the coefficient vector $\mathbf{C}$. The notation can be expanded to formulate the equations

$$\sum_{k=0}^{n} \sum_{I} c_k a_I^{(k)} \delta_I^J = \sum_{k=0}^{n} c_k a_J^{(k)} = 0$$

where $a_J^{(k)}$ denotes the components of the exponentiated multivector. In matrix form, the above system can be represented as $\mathbf{CU} = \mathbf{0}$, where $\mathbf{C} := \{c_k\}$ is the coefficient vector, $\mathbf{U} := \{u_{ik} \equiv a_{\#(I)=i}^{(k)}\}$ is the coefficient matrix, and $\#(I)$ denote the enumeration of multi-index J.

If $k < m$, then only the null vector $\mathbf{C} = \mathbf{0}$ will solve the system so we suppose that $k > m$. Observe that if a multivector power does not contain the blade e_J, this results in the trivial identity $0 = 0$ and hence, in a null row in $\mathbf{U}$. Therefore, the above system of equations is underdetermined and $\det \mathbf{U} = 0$ must hold for a non-trivial solution to exist. Therefore, $\mathrm{rank}(\mathbf{U}) \geq m$.

Finally, if $k = m$, we still obtain an underdetermined system but $m = 2\lfloor \mathrm{rank}(\mathbf{U})/2 \rfloor$ by Proposition 11. Then, the null space of $\mathbf{U}$ is one-dimensional and will consist of the unnormalized coefficients of the minimal polynomial. □

The Maxima code implementing the algorithm is presented in Listing A3.

Remark 4. *If the matrix rank is determined by direct computation, the result stated in Proposition 12 may not be practical. On the other hand, the computation of the set* $\mathcal{K}$ *can be parallelized, which can lead to time saving. On the second place, there maybe more economic algorithms for determining the rank of* A.

An Application to Multivector Exponentiation

A particular application of the presented algorithm is the exponentiation of multivectors. The multivector exponent is defined by the infinite series

$$e^{Mt} := \sum_{k=0}^{\infty} \frac{t^k M^k}{k!} \tag{26}$$

where t is a scalar and M a multivector. The Laplace transform action is, therefore,

$$\mathcal{L}[e^{Mt}](s) = (s - M)^{-1} \tag{27}$$

Therefore, the exponent can be calculated as the line integral

$$e^{Mt} = \frac{1}{2\pi i} \int_{Br} e^{st} (s - M)^{-1} ds$$

along the Bromwich contour. The integral can be evaluated from the Residue Theorem as

$$e^{Mt} = \sum_{i=1}^{k} \mathrm{Res}\Big(e^{st}(s-M)^{-1},\, s = s_i\Big), \tag{28}$$

where k is the degree of the minimal polynomial $\mu(s-M)$ and s_i gives its roots. For rational coefficients, in some cases, the denominator can be decomposed into partial fractions. This amounts to finding the factorization of the characteristic polynomial. For clarity of the discussion, let $P_{s-M}(v) = \mu(s-M)$. The roots of the equation $P_{s-M}(v) = 0$ in the s variable should be evaluated at $v = 0$ to yield the poles s_i. This amounts to computing the determinant and finding its roots in the Laplace variable

$$\det(s-M) = 0.$$

In practice, this can be achieved by numerical global root-finding algorithms.

7. Methods and Implementation

Algorithms were implemented in the Maxima code, based on the *Clifford* package in Maxima, which was first utilized in [4]. The present version of the package is 2.5 and is available for download from a Zenodo repository [5]. The function fadlevicg2cp simultaneously computes the inverse (if it exists) and the characteristic polynomial $p_A(v)$ of a multivector A (Appendix A). The function minpoly2 simultaneously computes the rank of the multivector and its minimal polynomial.

Experiments were performed on a Dell® 64-bit Microsoft Windows 10 Enterprise machine with configuration—Intel® Core™ i5-8350U CPU @ 1.70 GHz, 1.90 GHz, and 16 GB 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.

8. 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 [20])

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

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 reduced 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 can conclude that A_q encodes the same Clifford (i.e., quaternion) algebra, which was also recognized in [20] using the full multiplication table of the matrix algebra. The identification using the multiplication table requires 16 matrix multiplications if the maximal matrix representation is used as was the case in [20], while the multivector FVS algorithm requires only 2 multivector multiplications, as shown above.

In three dimensions, the general element is of the form

$$a_1 + e_1\, a_2 + e_2\, a_3 + e_3\, a_4 + a_5\,(e_1 e_2) + a_6\,(e_1 e_3) + a_7\,(e_2 e_3) + a_8\,(e_1 e_2 e_3)$$

Then we have the following maximal real matrix representations:

Example 5 (Algebra of Physical Space $C\ell_{3,0}$)**.**

$$\mathbf{A} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ a_2 & a_1 & a_5 & a_6 & a_3 & a_4 & a_8 & a_7 \\ a_3 & -a_5 & a_1 & a_7 & -a_2 & -a_8 & a_4 & -a_6 \\ a_4 & -a_6 & -a_7 & a_1 & a_8 & -a_2 & -a_3 & a_5 \\ -a_5 & a_3 & -a_2 & -a_8 & a_1 & a_7 & -a_6 & a_4 \\ -a_6 & a_4 & a_8 & -a_2 & -a_7 & a_1 & a_5 & -a_3 \\ -a_7 & -a_8 & a_4 & -a_3 & a_6 & -a_5 & a_1 & a_2 \\ -a_8 & -a_7 & a_6 & -a_5 & a_4 & -a_3 & a_2 & a_1 \end{pmatrix}$$

Example 6 (Clifford algebra $C\ell_{2,1}$).

$$\mathbf{A} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ a_2 & a_1 & a_5 & a_6 & a_3 & a_4 & a_8 & a_7 \\ a_3 & -a_5 & a_1 & a_7 & -a_2 & -a_8 & a_4 & -a_6 \\ -a_4 & a_6 & a_7 & a_1 & -a_8 & -a_2 & -a_3 & a_5 \\ -a_5 & a_3 & -a_2 & -a_8 & a_1 & a_7 & -a_6 & a_4 \\ a_6 & -a_4 & -a_8 & -a_2 & a_7 & a_1 & a_5 & -a_3 \\ a_7 & a_8 & -a_4 & -a_3 & -a_6 & -a_5 & a_1 & a_2 \\ a_8 & a_7 & -a_6 & -a_5 & -a_4 & -a_3 & a_2 & a_1 \end{pmatrix}$$

Example 7 (Clifford algebra $C\ell_{1,2}$).

$$\mathbf{A} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ a_2 & a_1 & a_5 & a_6 & a_3 & a_4 & a_8 & a_7 \\ -a_3 & a_5 & a_1 & -a_7 & -a_2 & a_8 & a_4 & -a_6 \\ -a_4 & a_6 & a_7 & a_1 & -a_8 & -a_2 & -a_3 & a_5 \\ a_5 & -a_3 & -a_2 & a_8 & a_1 & -a_7 & -a_6 & a_4 \\ a_6 & -a_4 & -a_8 & -a_2 & a_7 & a_1 & a_5 & -a_3 \\ -a_7 & -a_8 & -a_4 & a_3 & -a_6 & a_5 & a_1 & a_2 \\ -a_8 & -a_7 & -a_6 & a_5 & -a_4 & a_3 & a_2 & a_1 \end{pmatrix}$$

Example 8 (Clifford algebra $C\ell_{0,3}$).

$$\mathbf{A} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ -a_2 & a_1 & -a_5 & -a_6 & a_3 & a_4 & -a_8 & a_7 \\ -a_3 & a_5 & a_1 & -a_7 & -a_2 & a_8 & a_4 & -a_6 \\ -a_4 & a_6 & a_7 & a_1 & -a_8 & -a_2 & -a_3 & a_5 \\ -a_5 & -a_3 & a_2 & -a_8 & a_1 & -a_7 & a_6 & a_4 \\ -a_6 & -a_4 & a_8 & a_2 & a_7 & a_1 & -a_5 & -a_3 \\ -a_7 & -a_8 & -a_4 & a_3 & -a_6 & a_5 & a_1 & a_2 \\ a_8 & -a_7 & a_6 & -a_5 & -a_4 & a_3 & -a_2 & a_1 \end{pmatrix}$$

Example 9. *Consider $C\ell_{3,0}$. Let*

$$A = a_1 + e_1 a_2 + e_2 a_3 + e_3 a_4 + a_5 e_{12} + a_6 e_{13} + a_7 e_{23} + a_8 e_{123}$$

Then the application of the reduced 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

$$V=\begin{pmatrix}e_1\\e_2\\e_3\end{pmatrix}\begin{pmatrix}-a_1^2a_2+a_2^3+a_2a_3^2+a_2a_4^2-a_2a_5^2-a_2a_6^2+2a_4a_5a_7-2a_3a_6a_7+a_2a_7^2-2a_1a_7a_8+a_2a_8^2\\-a_1^2a_3+a_2^2a_3+a_3^3+a_3a_4^2-a_3a_5^2-2a_4a_5a_6+a_3a_6^2-2a_2a_6a_7-a_3a_7^2+2a_1a_6a_8+a_3a_8^2\\-a_1^2a_4+a_2^2a_4+a_3^2a_4+a_4^3+a_4a_5^2-2a_3a_5a_6-a_4a_6^2+2a_2a_5a_7-a_4a_7^2-2a_1a_5a_8+a_4a_8^2\end{pmatrix} \tag{29}$$

the bi-vector part is given by

$$BV=\begin{pmatrix}e_{12}\\e_{23}\\e_{13}\end{pmatrix}\begin{pmatrix}-a_1^2a_5+a_2^2a_5+a_3^2a_5-a_4^2a_5-a_5^3+2a_3a_4a_6-a_5a_6^2-2a_2a_4a_7-a_5a_7^2+2a_1a_4a_8+a_5a_8^2\\2a_3a_4a_5-a_1^2a_6+a_2^2a_6-a_3^2a_6+a_4^2a_6-a_5^2a_6-a_6^3+2a_2a_3a_7-a_6a_7^2-2a_1a_3a_8+a_6a_8^2\\-2a_2a_4a_5+2a_2a_3a_6-a_1^2a_7-a_2^2a_7+a_3^2a_7+a_4^2a_7-a_5^2a_7-a_6^2a_7-a_7^3+2a_1a_2a_8+a_7a_8^2\end{pmatrix} \tag{30}$$

and the pseudoscalar part by

$$Q=I\Big(2a_1a_4a_5-2a_1a_3a_6+2a_1a_2a_7-a_1^2a_8-a_2^2a_8-a_3^2a_8-a_4^2a_8+a_5^2a_8+a_6^2a_8+a_7^2a_8-a_8^3\Big)$$

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.

Example 10. *Consider $C\ell_{3,0}$ and the multivector $A=-4/3e_1+2/3I$. Then the Laplace transform is*

$$C=(s+4/3e_1-2/3I)^{-1}$$

The rank of the multivector C is 4 as computed from the matrix

$$U=\begin{pmatrix}1 & s & \frac{1}{3}(4+3s^2) & 4s+s^3 & \frac{1}{81}(-112+648s^2+81s^4)\\0 & \frac{4}{3} & \frac{8}{3}s & \frac{1}{27}(16+108s^2) & \frac{1}{27}(64s+144s^3)\\0 & 0 & -\frac{16}{9} & -\frac{16}{3}s & -\frac{1}{27}(128+288s^2)\\0 & -\frac{2}{3} & -\frac{4}{3}s & -\frac{1}{27}(88+54s^2) & -\frac{1}{27}(352s+72s^3)\end{pmatrix}$$

The minimal polynomial is

$$81\mu(v)=(400-216s^2+81s^4+(432s-324s^3)v+(-216+486s^2)v^2-324sv^3+81v^4)$$

Then C can be explicitly computed in four steps to yield

$$C=\frac{3\big(80e_1+40I-36s-48se_{23}-36s^2e_1+18s^2I+27s^3\big)}{(20-24s+9s^2)(20+24s+9s^2)}$$

The partial fraction decomposition is

$$C=-\frac{1}{2}\frac{(12-12e_1-6I+6e_{23}+(-9+9e_1)s)}{20-24s+9s^2}+\frac{1}{2}\frac{(12+12e_1+6I+6e_{23}+(9+9e_1)s)}{20+24s+9s^2} \tag{31}$$

The poles are located at

$$s_{1,2}=-\frac{2}{3}(2\pm i),\quad s_{3,4}=\frac{2}{3}(2\pm i)$$

Therefore, the exponent is

$$e^{At}=\frac{1}{2}\Big((1+e_1)\cos\Big(\frac{2t}{3}\Big)+(I+e_{23})\sin\Big(\frac{2t}{3}\Big)\Big)e^{-\frac{4t}{3}}+\frac{1}{2}\Big((1-e_1)\cos\Big(\frac{2t}{3}\Big)+(I-e_{23})\sin\Big(\frac{2t}{3}\Big)\Big)e^{\frac{4t}{3}} \tag{32}$$

9. Numerical Experiments

Note that the trivial last steps will be omitted. To demonstrate the utility of the FVS algorithm, we will now follow some high-dimensional numerical examples. Examples for higher-dimensional algebras are not particularly instructive as they result in very long expressions. These can nevertheless be useful for hard-coding formulas in particular niche applications.

Example 11. *In $C\ell_{2,2}$, let $A = 1 + e_1 + e_{134} - 2e_{23}$. Let $B = e_{134}$, $C = e_{123}$*

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

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 12. *Let us compute a rational example in $C\ell_{2,5}$. To avoid cumbersome expressions, let $A = 1 - 2e_{15} + 5e_{134} = 1 - 2B + 5C$, where $B := e_{15}$ and $C := e_{134}$.*

Then span$[A] = \{e_1, e_3, e_4, e_5\}$ *and for the maximal representation we have $k = 2^4 = 16$ steps:*

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

Therefore,

$$A^{-1} = (1 - 5C + 2B)/22$$

and $\chi_A(v) = (22 - 2v + v^2)^8$. The 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:

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

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

Example 13. *This example was presented in [21]. Consider the algebra $C\ell_{5,0}$ and define 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 $D = e_{145}$. The calculation proceeds as*

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

The inverse is

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

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

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

which is of rank 8.

Example 14. *This example was presented in [19]. In $C\ell_{4,0}$, define $A := 1 + e_1 + 3e_{23} - e_{24}$. Then the inverse computes in $n = 4$ steps:*

$$
\begin{aligned}
t_1 &= -4, & m_1 &= 1 + e_1 + 3e_{23} - e_{24};\\
t_2 &= 24, & m_2 &= -12 - 2e_1 + 6e_{123} - 2e_{124} - 6e_{23} + 2e_{24};\\
t_3 &= -40, & m_3 &= 30 - 10e_1 - 6e_{123} + 2e_{124} + 36e_{23} - 12e_{24};\\
t_4 &= 140, & m_4 &= 0,
\end{aligned}
$$

resulting in

$$
A^{-1} = \frac{1}{70}(5 + 5e_1 + 3e_{123} - e_{124} - 18e_{23} + 6e_{24})
$$

The characteristic polynomial is

$$
\chi_A(v) = 140 - 40v + 24v^2 - 4v^3 + v^4 = (10 + v^2)(14 - 4v + v^2)
$$

which is also minimal. Therefore, the rank of the multivector is 4.

The inverse can be computed alternatively in the following way. Let

$$
B := A\widehat{A} = -10 + 6e_{23} - 2e_{24}, \quad B^{\sim} = -10 - 6e_{23} + 2e_{24}, \quad \Delta = BB^{\sim} = 140
$$

Therefore, the inverse is given by the formula

$$A^{-1} = \frac{\widehat{A}(A\widehat{A})^{\sim}}{\Delta}$$

in accordance with the above result. This is an alternative to Equation (22).

Example 15. *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 limitations. 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

$$\chi_A(v) = (5 - 4v + 6v^2 - 4v^3 + v^4)^2 = (1 + v^2)^4(5 - 4v + v^2)^4 \tag{33}$$

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

Example 16. *We use the same data $A = 1 - e_2 + I$ in $C\ell_{5,2}$ to compute the rank according to Proposition 12. The reduced (with zero rows removed) rank matrix is*

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

which is of rank 4. Direct computation of the inverse results in

$$\begin{aligned} t_1 &= -4, \quad m_1 = 1 - e_2 + I; \\ t_2 &= 6, \quad m_2 = -3 + 2e_2 - 2I + 2C; \\ t_3 &= -4, \quad m_3 = 3 + e_2 + 3I - 2\,C \end{aligned}$$

where, as before, $C = e_{134567}$ and $A^{-1} = (1 - e_2 + 2C - 3I)/5$. 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 simple formula*

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

Example 17. *We use $A := 1 - e_2 + e_3 + e_{13456}$ residing in $C\ell_{5,2}$ to compute the rank according to Proposition 12. The span is a six-dimensional vector space—*$\mathrm{span}[A] = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. *The reduced (with zero rows removed) rank matrix is*

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

which is of rank 5.

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

$$A^{-1} = \frac{1}{2}(e_3 + e_{12456} - e_{13456} - e_{1456}) \tag{34}$$

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.

10. Discussion

Computation of inverses of multivectors has drawn continuous attention in the literature as the problem has only been gradually solved [10,21–23]. In order to compute the inverse of a multivector, previous contributions used series of automorphisms of special types discussed in Section 2.3. This allows one to write basis-free formulas with increasing complexity.

From Table 1, it can be concluded 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 proving the FVS algorithm entirely in the language of Clifford algebra in line with [19]. Another possible application is deriving formulas for exponents of multivectors.

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 the sorting of a joined list of algebra generators. On the other hand, the FVS algorithm does not ensure optimality of the computation but nevertheless 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, also its determinant without any resort to a matrix representation. This could be used, for example, for the computation of a multivector resolvent or some other analytical functions.

One of the main applications of the present algorithms could be envisioned 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.

Funding: The present work was funded by the European Union's Horizon Europe program under grant agreement VIBraTE, 101086815.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The author declares that the data supporting the findings of this study are available within the paper and the Zenodo data repository. The studied examples can be downloaded from the Zenodo repository and it includes the file climatrep.mac, which implements different instances of the FVS algorithm [24].

Conflicts of Interest: The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

FVS Faddeev–LeVerrier–Souriau

Appendix A. Program Code

The Clifford package can be downloaded from a Zenodo repository [5].

Listing A1. Matrix representation code.

```
/*  computes blade representation */
climatrep(vv):=block([n, q, AA, lst: elements(all), G, EE],
    local(AA, G, EE),
    n:length(lst),
    /* multiplication table of the algebra */
    AA:genmatrix( lambda([i,j], dotsimpc(lst[i] . lst[j] ) ), n),
    /* signature of the algebra */
    G:diag(AA),
    EE:matrixmap(lambda([q],  scalarpart(ratcoeff (vv, q))), AA),
    G.EE
);
```

```
/*  computes expression representation */
elem2mat1(expr, [ulst] ):=block(
    if emptyp(ulst) then ulst:false
    else ulst:true,
    if mapatom(expr) then return(climatrep1(expr)),
    if ulst=true then
        maplist(climatrep1, expr)
    else
        map(climatrep1, expr)
);
```

Listing A2. 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]
);
```

Listing A3. Minimal polynomial computation.

```
minpoly1(A, v, [m]):=block([Ls:[], L1, u, rnk, U, Un, E, W, Wt, %rnum:0, c,
    cp, listarith:true],
    local(c),  declare(c, scalar),
    if emptyp(m) then m:mgr(A)
    else m:first(m),
    Ls: mpowers(A,m),
    E:elementsr(A),
    E: ei:maplist(lambda([u], dotsimpc(u.u).u ), E),
    L1:  makelist(c[i],i, length(Ls)),
    Ls:Ls.L1,

    U:makelist(Ls @E[i] =0,i, length( E)),
    Un:sublist(U, lambda([u], u# (0=0))),
    W: coefmatrix(Un,L1),
    rnk:fix(rank(W)/2)*2 ,

    Wt: linsolve(Un, makelist(L1[i], i, rnk+1)),
    Wt: subst(makelist(L1[i]=0, i, rnk+2, length(Ls)) , Wt),
    Wt: map(rhs, Wt),
    Wt:Wt/last(Wt),      Wt: ratsimp(Wt),
```

```
        cp: makelist(v^^i,i, 0, rnk),
        [rnk,  Wt.cp]
);
```

References

1. Hitzer, E.; Kamarianakis, M.; Papagiannakis, G.; Vasik, P. Survey of new applications of geometric algebra. *Math. Methods Appl. Sci.* **2024**, *47*, 11368–11384. [CrossRef]
2. Ablamowicz, R.; Fauser, B. Mathematics of CLIFFORD—A Maple package for Clifford and Grassmann algebras. *Adv. Appl. Clifford Algebr.* **2002**, *15*, 157–181.
3. Sangwine, S.J.; Hitzer, E. Clifford Multivector Toolbox (for MATLAB). *Adv. Appl. Clifford Algebr.* **2016**, *27*, 539–558. [CrossRef]
4. Prodanov, D.; Toth, V.T. Sparse Representations of Clifford and Tensor Algebras in Maxima. *Adv. Appl. Clifford Algebras* **2016**, *27*, 539–558. [CrossRef]
5. Prodanov, D. Clifford Maxima Package v 2.5.4. Available online: https://zenodo.org/records/8205828 (accessed on 15 January 2025).
6. Prodanov, D. Algorithmic Computation of Multivector Inverses and Characteristic Polynomials in Non-degenerate Clifford Algebras. In *Advances in Computer Graphics*; Springer Nature: Cham, Switzerland, 2023; pp. 379–390. [CrossRef]
7. Hestenes, D.; Sobczyk, G. *Clifford Algebra to Geometric Calculus*; Springer: Berlin/Heidelberg, Germany, 1984; p. 336.
8. Ablamowicz, R.; Parra, J.; Lounesto, P. *Clifford Algebras with Numeric and Symbolic Computations*; Birkhäuser: Basel, Switzerland, 1996; p. 340.
9. Renaud, P. *Clifford Algebra Lecture Notes on Applications in Physics*; HAL Open Science: Lyon, France, 2020; p. 253.
10. Hitzer, E.; Sangwine, S. Multivector and multivector matrix inverses in real Clifford algebras. *Appl. Math. Comput.* **2017**, *311*, 375–389. [CrossRef]
11. Shirokov, D.S. Clifford algebras and their applications to Lie groups and spinors. In Proceedings of the Nineteenth International Conference on Geometry, Integrability and Quantization, Varna, Bulgaria, 2–7 June 2017; Volume 19, pp. 11–53. [CrossRef]
12. Garibaldi, S. The Characteristic Polynomial and Determinant Are Not Ad Hoc Constructions. *Am. Math. Mon.* **2004**, *111*, 761. [CrossRef]
13. Rickards, J. When Is a Polynomial a Composition of Other Polynomials? *Am. Math. Mon.* **2011**, *118*, 358. [CrossRef]
14. MacDonald, A. *Linear and Geometric Algebra*; CreateSpace Independent Publishing Platform: Scotts Valley, CA, USA, 2010; p. 223.
15. Lundholm, D. Geometric (Clifford) Algebra and Its Applications. Master's Thesis, KTH, Stockholm, Sweden, 2006. [CrossRef]
16. Shirokov, D. Concepts of trace, determinant and inverse of Clifford algebra elements. *Prog. Anal.* **2011**, *1*, 187–194.
17. Dadbeh, P. Inverse and Determinant in 0 to 5 Dimensional Clifford Algebra. *arXiv* **2011**, arXiv:1104.0067. [CrossRef]
18. Faddeev, D.K.; Sominskij, I.S. *Sbornik Zadatch po Vyshej Algebre*; Nauka: Moscow/Leningrad, Russia, 1949.
19. Acus, A.; Dargys, A. The characteristic polynomial in calculation of exponential and elementary functions in Clifford algebras. *Math. Methods Appl. Sci.* **2022**. [CrossRef]
20. Petoukhov, S.V. Symmetries of the genetic code, hypercomplex numbers and genetic matrices with internal complementarities. *Symmetry Cult. Sci.* **2012**, *23*, 225–448.
21. Acus, A.; Dargys, A. The Inverse of a Multivector: Beyond the Threshold $p + q = 5$. *Adv. Appl. Clifford Algebr.* **2018**, *28*, 65. [CrossRef]
22. 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 Algebr.* **2019**, *29*, 29. [CrossRef]
23. Shirokov, D.S. On computing the determinant, other characteristic polynomial coefficients, and inverse in Clifford algebras of arbitrary dimension. *Comp. Appl. Math.* **2021**, *40*, 173. [CrossRef]
24. Prodanov, D. Examples for CGI2023. Available online: https://zenodo.org/records/10327066 (accessed on 15 January 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.