Practical Computing with Geometric Algebra
Converting Basic Geometric Algebra Relations to Computations on Multivector Coordinates

Ahmad Hosny Eid,
Assistant Professor,
Computer Engineering Division,
Faculty of Engineering,
Port-Said University, Egypt

July 2016

Abstract

Geometric Algebra (GA) is a fascinating mathematical language for unifying many tools engineers and scientists use on daily basis. GA can also be used for algebraic manipulation of Euclidean and non-Euclidean geometries in a consistent manner. Many sources can be found to explain GA and its properties, but few relate GA to computing for practical software engineering. This article provides a summary, without proofs, of the fundamental algebraic concepts and operations of GA (sections 1 to 6). After this, the article contains an explanation of how to transform high-level mathematical GA products and algebraic operations into equivalent lower-level computations on multivector coordinates (sections 7 and 8). The aim is to provide a computational basis for implementing compilers that can automatically perform such conversion for the purpose of efficient software implementations of GA-based models and algorithms.

1 The Outer Product: Constructing Blades

Creating a geometric algebra requires a base real vector space with an associated symmetric bilinear form that defines a metric on the space. Additional mathematical structure comes from the outer product operation that extends vectors into blades. Blades algebraically correspond to subspaces in the base vector space. In what follows the additional structure that distinguishes GA from classical linear algebra is summarized; mostly based on [1] and [2]. For more information about vector spaces and classical linear algebra the reader can refer to [3] and [4].

1.1 Subspaces of Real Vector Spaces

A subspace W of a vector space V (denoted by $W \leq V$) is a subset of the vector space that is closed under vector addition and vector multiplication with scalars; thus any subspace must contain the zero vector. The set intersection of two subspaces is always a subspace. The set union of two subspaces is not guaranteed to be a subspace. A similar operation to set union that guarantees a subspace result is called the sum of subspaces defined as:

$$W + U = \{x :: x = w + u;\ w \in W, u \in V\},\ W, V \leq V$$

Having a set of mutually disjoint subspaces $W_1, W_2, \cdots, W_k \leq V$ (i.e. $W_i \cap W_j = \{0\}\ \forall i, j = 1, 2, \cdots k,\ i \neq j$) the subspace sum of W_i is called the direct sum of the disjoint subspaces and is denoted by $W_1 \oplus W_2 \oplus \cdots \oplus W_k$. The dimension of the direct sum of disjoint subspaces is equal to the numerical sum of their respective dimensions. Another important concept is the orthogonal complement of a metric subspace $W \leq V$ defined by $W^{\perp} = \{x :\in V : y \perp x\, \forall y \in W\}$. The orthogonal complement of a subspace $W \leq V$ has the following properties:

$$\begin{aligned} V &= W \oplus W^{\perp} \\ x &\perp y\ \forall x \in W, y \in W^{\perp} \\ (W^{\perp})^{\perp} &= W \end{aligned}$$

1.2 Direct Representation of Subspaces

Having a n-dimensional real vector space $\mathbb{R}^n$ with an ordered set of basis vectors $\langle e_1, e_2, \cdots, e_n \rangle$, the focus is on all subspaces of $\mathbb{R}^n$ of all dimensions k where $0 \le k \le n$. The geometric meaning of any such subspace is a k-dimensional flat (the origin, a line, a plane, etc.) in $\mathbb{R}^n$ that contains the origin. The outer product of an ordered set of k Linearly Independent (LID) vectors $\langle a_1, a_2, \cdots, a_k \rangle$ is used to define algebraic objects, called k-blades in GA, that can be used to represent such subspaces algebraically with four main characteristics for each subspace:

1. The dimensionality of a subspace k.
2. The attitude of the subspace: this is equivalent to the traditional span in classical linear algebra of the set of vectors $\{a_1, a_2, \cdots, a_k\}$.
3. The orientation of the subspace: which is a sign (+1 or -1) associated with the subspace to define the relative orientation or handedness of the basis set.
4. The weight of the subspace: which is a real number associated with the attitude (and it also includes the sign i.e. the orientation\handedness of the subspace).

The simplest subspace is the 0-dimensional subspace spanned by no vectors (i.e. it only contains the zero vector) with a corresponding 0-blade that is simply a scalar $\lambda \in \mathbb{R}$; this subspace will be denoted by $B_0^n = \mathbb{R}$. Any vector $x \in \mathbb{R}^n$ is a 1-blade by definition and it corresponds to a 1-dimensional subspace spanned by that vector alone; the space of 1-blades will be denoted by by $B_1^n = \mathbb{R}^n$. The set of k-blades for any value of $k \in \{0, 1, \cdots, n\}$ is denoted by B_k^n and the set of all blades is denoted by $B^n = \bigcup_{i=0}^{n} B_i^n$.

The outer product is an associative bilinear product used to construct higher-grade blades from lower-grade ones $\wedge : B_r^n \times B_s^n \to B_{r+s}^n$, $r, s, r+s \in \{0, 1, \cdots, n\}$. The basic properties of the outer product of scalars (0-blades) and vectors (1-blades) and general k-blades are as follows:

$$
\begin{aligned}
\alpha \wedge \beta &= \alpha\beta && (1)\\
\alpha \wedge x = x \wedge \alpha &= \alpha x && (2)\\
x \wedge y &= -y \wedge x && (3)\\
X \wedge (Y + Z) &= X \wedge Y + X \wedge Z && (4)\\
A \wedge (B \wedge C) &= (A \wedge B) \wedge C && (5)\\
A \wedge (\alpha B) &= \alpha(A \wedge B) && (6)
\end{aligned}
$$

$$
\begin{aligned}
\alpha, \beta &\in B_0^n;\\
x, y, z &\in B_1^n;\\
X, Y, Z, (Y + Z) &\in B_k^n;\\
A, B, C &\in B^n
\end{aligned}
$$

The anti-symmetry property (3) of the outer product with respect to vectors leads to the important relation:

$$x \wedge x = -x \wedge x = 0 \qquad (7)$$

In addition, (3) is a special case of a more general property:

$$X \wedge Y = (-1)^{rs} Y \wedge X,\ X \in B_r^n, Y \in B_s^n$$

Having a k-dimensional subspace $\overleftrightarrow{A}$ spanned by a set of LID vectors $\langle a_1, a_2, \cdots, a_k \rangle$, the k-blade $A = a_1 \wedge a_2 \wedge \cdots \wedge a_k$ can be used to determine if a vector x belongs to the subspace $\overleftrightarrow{A}$ based on x being a linear combination of the spanning vectors of $\overleftrightarrow{A}$; hence $x \in \overleftrightarrow{A} \Leftrightarrow x \wedge A = 0$. In this case the k-blade A is called a direct representation of the subspace $\overleftrightarrow{A}$ (denoted by $A \propto \overleftrightarrow{A}$). It is obvious that $A \propto \overleftrightarrow{A} \Rightarrow \lambda A \propto \overleftrightarrow{A}\ \forall \lambda \in B_0^n$. The same concept can be generalized to arbitrary subspaces by stating that a subspace $\overleftrightarrow{B} = span\{b_1, b_2, \cdots, b_r\}$ is contained in $\overleftrightarrow{A}$ (i.e. $\overleftrightarrow{B} \le \overleftrightarrow{A}$) if each spanning vector in $\overleftrightarrow{B}$ satisfies $b_i \wedge A = 0$, $i = 1, \cdots, r$. This is not at all equivalent to $b_1 \wedge b_2 \wedge \cdots \wedge b_r \wedge A = 0$ because this last relation holds if any spanning vector b_i is contained in $\overleftrightarrow{A}$. As soon as, say, a 2-blade $A = \lambda a \wedge b$ is constructed from two LID vectors a, b and a scalar λ , all algebraic information regarding the exact values of the two vectors and scalar in the constructed 2-blade are lost. This is apparent from the fact that the two vectors can always be expressed as a linear combination of two arbitrary LID vectors x, y in the same 2-dimensional subspace as a, b and this means that the exact same 2-blade can be constructed using an infinite number of outer products of two vectors:

$$\begin{aligned} a &= a_1 x + a_2 y \\ b &= b_1 x + b_2 y \\ A &= \lambda a \wedge b \\ &= \lambda(a_1 x + a_2 y) \wedge (b_1 x + b_2 y) \\ &= \gamma x \wedge y \ , \gamma = \lambda(a_1 b_2 - a_2 b_1) \end{aligned}$$

This directly means that a 2-blade has no specific shape but rather can be understood as a unit of area freely floating around like the case of free vectors in traditional vector algebra. For any k-blade $a_1 \wedge a_2 \wedge \cdots \wedge a_k$ the number k representing the dimensionality of the represented subspace is called the grade (or step) of the k-blade. The grade operator is thus defined as $grade(a_1 \wedge a_2 \wedge \cdots \wedge a_k) = k$. The grade of the outer product of two blades A, B is the sum of their respective grades: $grade(A \wedge B) = grade(A) + grade(B)$. A blade of grade n is called a pseudo-scalar of the GA because it contains all vectors and blades of the GA. The unit pseudo-scalar is the n-blade $I = e_1 \wedge e_2 \wedge \cdots \wedge e_n$. Another useful operator in this context is the odd operator defined as:

$$\begin{aligned} odd(A) &= \begin{cases} 1 & grade(A)\, is\, odd \\ 2 & grade(A)\, is\, even \end{cases} \\ & \forall A \in B^n \end{aligned}$$

The concept of a k-blade enables the comparison of subspaces of the same attitude (subspaces of the same span). If any two k-blades of the same grade A, B can be written such that $A = \lambda B, \lambda \in \mathbb{R}$ then the real number $\lambda = A/B$ holds the relative weight and orientation of the two blades. To compare two blades of different attitude a metric is needed to enable the operation of rotating one blade to the other as described in section 2.

1.3 The Algebra of k-vectors and the Graded Algebra of Multivectors

k-blades are formed by the outer product of k LID 1-blades (vectors). The resulting structure is a linear combination of new elements called basis k-blades $\langle E_1, E_2, \cdots, E_r \rangle$where $r = \begin{pmatrix} n \\ k \end{pmatrix}$. A basis k-blade E_i can be obtained by an outer product of k different basis vectors $e_{j_1} \wedge e_{j_2} \wedge \cdots \wedge e_{j_k}$, $j_1 < j_2 < \cdots < j_k$. Algebraically, not every linear combination of the basis k-blades results in a k-blade. Hence the vector space spanned by the basis k-blades contains all k-blades in addition to other non blade elements called k-vectors. The algebraic vector space spanned by the basis k-blades is called the k-vector space $\bigwedge_k^n$. k-blades are only just a part of the k-vector space $B_k^n \subseteq \bigwedge_k^n$ and hence every k-blade is a k-vector but not every k-vector is a k-blade. The only values of k where all k-vectors are k-blades are $k = 0, 1, n-1, n$ (i.e. $B_k^n = \bigwedge_k^n$ for these values of k). That means that in general not every linear combination of basis k-blades has a geometric meaning of a weighted oriented subspace of $\mathbb{R}^n$ but it is algebraically useful to define addition between arbitrary k-blades of the same grade to form k-vectors. The outer product of any vector by itself is the zero scalar. The zero scalar is taken to be equivalent to all the zeros of all k-vector spaces where $0 \leq k \leq n$ that is because the scalar zero can be geometrically interpreted as the empty subspace of any dimensionality. The k-vector space is simply a span of the basis k-blades hence any linear operation on k-blades can be easily extended to k-vectors including the bilinear outer product.

Allowing linear combinations of basis k-blades of the same grade produces the space of k-vectors. Now if linear combinations of basis blades of different grades are allowed, the graded space of multivectors is obtained, which is a vector space (a linear space) having 2^n basis blades with grades ranging from 0 to n. This space is called the Grassmann space of multivectors and denoted by $\bigwedge^n$. It is convenient to define the grade extraction operator acting on multivectors $\langle\rangle_k : \bigwedge^n \to \bigwedge_k^n$ that extracts the k-vector part of a multivector. Since the outer product is bilinear it can be generalized to arbitrary multivectors $A, B \in \bigwedge^n$ as :

$$\begin{aligned} A \wedge B &= \sum_{r=0}^{n} \sum_{s=0}^{n} \langle A \rangle_r \wedge \langle B \rangle_s \\ \Rightarrow \langle A \rangle_r \wedge \langle B \rangle_s &= \langle A \wedge B \rangle_{r+s}, r+s \leq n \end{aligned}$$

In addition, the zero scalar is equivalent to the zero multivector and all zero k-vectors in both addition operation and outer product:

$$A + 0 = A,\ A \wedge 0 = 0,\ A \in \bigwedge^n$$

Any general multivector thus contains a mixture of different grade elements. If the multivector only contains k-vectors of even grade it is called an even multivector. If it only contains k-vectors of odd grades it is called an odd multivector. The odd operator on blades can be generalized to multivectors as:

$$\begin{aligned} odd(A) &= \begin{cases} 1 & A\,is\,odd \\ 2 & A\,is\,even \\ undefined & otherwise \end{cases} \\ & \forall A \in \bigwedge^n \end{aligned}$$

1.4 The Reversion and Grade Involution

Two useful operations can be defined on blades. The first is the reversion defined on a k-blade $A = a_1 \wedge a_2 \wedge \cdots \wedge a_k$ as:

$$\widetilde{A} = a_k \wedge a_{k-1} \wedge \cdots \wedge a_1 = (-1)^{k(k-1)/2} A,\ A \in B_k^n$$

This sign change exhibits a $++--++--$ pattern with periodicity 4. The reversion can be generalized to k-vectors as:

$$\begin{aligned} \widetilde{A} &= (-1)^{k(k-1)/2} A,\ A \in \bigwedge_k^n \\ \Rightarrow \widetilde{A} &= \sum_{r=0}^{n} (-1)^{r(r-1)/2} \langle A \rangle_r,\ A \in \bigwedge^n \end{aligned}$$

.For any two multivectors A, B the reversion has the following properties that result in it being called an "anti-involution" because its result when applied twice is the same multivector (an involution) and when applied to the outer product it reverses the order of the arguments:

$$\begin{aligned} (A^{\sim})^{\sim} &= A \\ (A \wedge B)^{\sim} &= B^{\sim} \wedge A^{\sim} \end{aligned}$$

The second operation is the grade involution defined by:

$$\begin{aligned} \hat{A} &= (-1)^{odd(A)} A,\ A \in \bigwedge_k^n \\ \hat{A} &= \sum_{r=0}^{n} (-1)^{r} \langle A \rangle_r,\ A \in \bigwedge^n \end{aligned}$$

This sign change exhibits a $+-+-+-+-$ pattern with periodicity 2. The grade involution has the following properties for any two multivectors A, B:

$$\begin{aligned}(A^{\wedge})^{\wedge} &= A\\ (A\wedge B)^{\wedge} &= A^{\wedge}\wedge B^{\wedge}\end{aligned}$$

Using the two operations a third one called the Clifford conjugate can be defined, which is also an anti-involution as:

$$\begin{aligned}\overline{A} &= (\widetilde{A})^{\wedge}\\ &= (-1)^{k(k+1)/2}A,\ A\in\bigwedge_{k}^{n}\\ \overline{A} &= \sum_{r=0}^{n}(-1)^{r(r+1)/2}\left\langle A\right\rangle_{r},\ A\in\bigwedge^{n}\end{aligned}$$

This sign change exhibits a $+--++--+$ pattern with periodicity 4. The Clifford conjugate has the following properties for any two multivectors A, B:

$$\begin{aligned}\overline{\overline{A}} &= A\\ \overline{(A\wedge B)} &= \overline{B}\wedge\overline{A}\end{aligned}$$

2 Metric Products of Sub-spaces: Comparing Blades

2.1 Symmetric bilinear Forms and Quadratic Forms: The Inner Product

A metric space is just a vector space with a way to compute the norm of an arbitrary vector. This essentially associates each vector in the space with some scalar. If two vectors are associated with the same scalar they are of equal norm. In this context the norm is any general number; even zero and negative numbers are allowed for non-zero vectors. This is one big generalization different from metrics in classical linear algebra that are usually positive definite. The objective here is to enable comparing vectors and subspaces of different attitude in space using numbers.

In GA the definition of a metric is based on the concept of a symmetric bilinear form and the associated concept of a quadratic form. A symmetric bilinear form $\mathbf{B}$ on the real vector space $\mathbb{R}^{\mathrm{n}}$ is a mapping $\mathbf{B}:\mathbb{R}^{\mathrm{n}}\times\mathbb{R}^{\mathrm{n}}\to\mathbb{R}$ that is linear in both arguments (i.e. bilinear) and symmetric $\mathbf{B}(u,v)=\mathbf{B}(v,u)\ \forall u,v\in\mathbb{R}^{\mathrm{n}}$. The bilinear form essentially associates any pair of vectors (regardless of their order) with a real scalar. If the vector space has the basis $\langle e_1, e_2,\cdots,e_n\rangle$ then the so called bilinear form matrix $\mathbf{A_B}=[a_{ij}]\,, a_{ij}=\mathbf{B}(e_i,e_j)$ can be constructed. This matrix is naturally a real symmetric matrix that can be used to compute the bilinear form of any two vectors given their representation on the basis as follows:

$$\begin{aligned}u &= u_1e_1+\cdots+u_ne_n,\\ v &= v_1e_1+\cdots+v_ne_n\\ \Rightarrow\mathbf{B}(u,v) &= \begin{pmatrix}u_1 & \cdots & u_n\end{pmatrix}\mathbf{A_B}\begin{pmatrix}v_1 & \cdots & v_n\end{pmatrix}^T\end{aligned}$$

Using bilinear forms the concept of orthogonality of vectors can be defined as follows: two vectors u, v are called orthogonal if and only if $\mathbf{B}(u,v)=0$. The inner product of two vectors is simply the bilinear form of the vectors $(u\cdot v=\mathbf{B}(u,v))$ and the metric is the inner product of a vector with itself thus justifying the use of the name "inner product matrix" for the bilinear form matrix.

A related concept is the quadratic form that is related to a symmetric bilinear form by: $\mathbf{Q}(u)=\frac{1}{2}\mathbf{B}(u,u),\ \mathbf{B}(u,v)=\mathbf{Q}(u+v)-\mathbf{Q}(u)-\mathbf{Q}(v)\ \forall u,v\in\mathbb{R}^{\mathrm{n}}$. The quadratic form satisfies the relation $Q(\lambda u)=\lambda^2Q(u)\ \forall u\in\mathbb{R}^{\mathrm{n}},\lambda\in\mathbb{R}$. This means that the metric (the squared length of a vector in the special case of Euclidean vector spaces) is simply twice the value of the quadratic form. The value of the quadratic form can be computed from the associated bilinear form matrix as follows:

$$
\begin{aligned}
u &= u_1 e_1 + \cdots + u_n e_n \\
\Rightarrow \mathbf{Q}(u) &= \frac{1}{2} \begin{pmatrix} u_1 & \cdots & u_n \end{pmatrix} \mathbf{A_B} \begin{pmatrix} u_1 & \cdots & u_n \end{pmatrix}^T
\end{aligned}
$$

The inner product matrix is a real symmetric matrix. Any real symmetric matrix $\mathbf{A}$ can be diagonalized by an orthogonal matrix to obtain a diagonal matrix $\mathbf{D} = \mathbf{P}^T \mathbf{A} \mathbf{P}$ where $\mathbf{P}$ is an orthogonal matrix (i.e. $\mathbf{P}^{-1} = \mathbf{P}^T$). The orthogonal matrix $\mathbf{P}$ used in the diagonalization process is actually a "change of basis" matrix that also happens to be orthogonal. The columns of $\mathbf{P}$ are actually the eigen vectors of $\mathbf{A}$. The diagonalization can always be performed such that the numbers on the diagonal (called the eigen values) are either -1, 0, or +1. The number of eigen values that are -1, 0, and 1 are characteristics for a given inner product matrix and define what is called the signature of the bilinear\quadratic form. A bilinear\quadratic form is said to have the signature (p, q, r) if there exists a diagonalization of the inner product matrix having p eigen values with positive value, q eigen values with negative value and r eigen values with zero values. If the inner product matrix is singular (i.e. has no inverse) the bilinear form is called degenerate. If all the eigen values are positive the inner product matrix is positive definite and the space is a Euclidean space (there exists a basis with all basis vectors norms equal to +1). A mixed-signature metric space has non-zero vectors with norm equal to zero. Such vectors are called null vectors and only exist in mixed-signature spaces (spaces having a bilinear form with both +1 and -1 signatures).

2.2 The Scalar Product of Blades: Comparing Same-Grade Blades

In a Euclidean space useful geometric operations on vectors can be defined using the inner product. For example the squared length of a vector $\|x\|^2 = x \cdot x$ and the angle between two vectors $\cos(\theta) = u \cdot v \diagup (\|u\| \, \|v\|)$. Similar geometric operations can be defined for higher grade blades using the scalar product of two k-blades of the same grade k. For example the scalar product of 2-blades can be used to define their areas, volumes for 3-blades, etc. The scalar product is defined follows:

$$
\begin{aligned}
* : B_k^n \times B_k^n &\rightarrow B_0^n \\
\alpha * \beta &= \alpha\beta, \\
\textit{where } \alpha, \beta &\in B_0^n \\
X * Y &= (-1)^{k(k-1)/2} \begin{vmatrix} \mathbf{B}(x_1, y_1) & \mathbf{B}(x_1, y_2) & \cdots & \mathbf{B}(x_1, y_k) \\ \mathbf{B}(x_2, y_1) & \mathbf{B}(x_2, y_2) & \cdots & \mathbf{B}(x_2, y_k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}(x_k, y_1) & \mathbf{B}(x_k, y_2) & \cdots & \mathbf{B}(x_k, y_k) \end{vmatrix} \\
&= \begin{vmatrix} x_1 \cdot y_k & x_1 \cdot y_{k-1} & \cdots & x_1 \cdot y_1 \\ x_2 \cdot y_k & x_2 \cdot y_{k-1} & \cdots & x_2 \cdot y_1 \\ \vdots & \vdots & \ddots & \vdots \\ x_k \cdot y_k & x_k \cdot y_{k-1} & \cdots & x_k \cdot y_1 \end{vmatrix} \\
\textit{where } X &= x_1 \wedge x_2 \wedge \cdots \wedge x_k, \\
Y &= y_1 \wedge y_2 \wedge \cdots \wedge y_k \\
X * Y &= 0 \textit{ otherwise}
\end{aligned}
$$

From the symmetry of the definition the following properties of the scalar product can be deduced:

$$
\begin{aligned}
A * B &= B * A \\
&= \widetilde{A} * \widetilde{B}
\end{aligned}
$$

Using the scalar product the squared norm of a k-blade A as: $\|A\|^2 = A * \widetilde{A}$ can be defined. The angle θ between two k-blades A, B of the same grade k can be defined as:

$$
\cos(\theta) = \frac{A * \widetilde{B}}{\|A\| \, \|B\|}
$$

Reinterpreting a zero cosine within this larger context, it either means that two blades are geometrically perpendicular in the usual sense (i.e. it takes a right turn to align them); or that they are algebraically orthogonal in the sense of being independent, (i.e., not having enough in common thus there is no single rotation with any angle that can make them identical).

2.3 The Contraction Product of Blades: Comparing General Blades

The scalar product has a non-zero value only for blades of the same grade (i,e. it relates subspaces of the same dimension). To compare subspaces of different dimension another method is required that should be universally applicable to all blades and linear in the same time. The left contraction of blades (denoted by $A\rfloor B$ and pronounced as A contracted on B) is one such method where $\rfloor : B^n_r \times B^n_s \rightarrow B^n_{s-r}\ , r, s, s-r \in \{0, 1, \cdots, n\}$. Having $A \propto \overleftrightarrow{A}$, $B \propto \overleftrightarrow{B}$, $A \in B^n_r B \in B^n_s$ the geometric meaning of $A\rfloor B$ is a (s-r)-blade $C \propto \overleftrightarrow{B} \cap (\overleftrightarrow{A})^{\perp}$. If the subspace $\overleftrightarrow{B} \cap (\overleftrightarrow{A})^{\perp}$has a dimension other than $s-r$ the result of $A\rfloor B$ is considered zero to make the left contraction a linear product. The explicit definition of the left contraction is as follows:

$$
\begin{aligned}
\alpha\rfloor\beta &= \alpha\beta \\
\alpha\rfloor A &= \alpha A \\
A\rfloor B &= 0,\ grade(A) > grade(B) \\
a\rfloor b &= \mathbf{B}(a, b) = a \cdot b \\
a\rfloor(B \wedge C) &= (a\rfloor B) \wedge C + (-1)^{grade(B)} B \wedge (a\rfloor C) \\
(A \wedge B)\rfloor C &= A\rfloor(B\rfloor C) \\
\alpha, \beta &\in B^n_0, \\
A, B, C &\in B^n
\end{aligned}
$$

The left contraction is bilinear and distributive over addition (but not associative) thus can be directly extended to k-vectors and thus to multivectors as:

$$
\begin{aligned}
A\rfloor B &= \sum_{r=0}^{n}\sum_{s=0}^{n} \langle A\rangle_{\,r}\rfloor \langle B\rangle_{\,s} \\
\Rightarrow \langle A\rangle_{\,r}\rfloor \langle B\rangle_{\,s} &= \langle A\rfloor B\rangle_{\,s-r}\,, s \geq r
\end{aligned}
$$

The non-associativity of the left contraction product is apparent from comparing the grade of $(A\rfloor B)\rfloor C$ and $A\rfloor(B\rfloor C)$ that are generally not equal. In addition the relation

$$(A \wedge B)\rfloor C = A\rfloor(B\rfloor C) \tag{8}$$

is valid for any three blades A, B, C whereas the following relation of the three blades is only valid in a certain condition:

$$(A\rfloor B)\rfloor C = A \wedge (B\rfloor C),\ A \leq C \tag{9}$$

Equations (8) and (9) are called the duality formulas.

One useful property of the contraction is given by:

$$
\begin{aligned}
x\rfloor(a_1 \wedge a_2 \wedge \cdots \wedge a_k) &= \sum_{i=1}^{k} a_1 \wedge a_2 \wedge \cdots \wedge (x\rfloor a_i) \wedge \cdots \wedge a_k \\
\Rightarrow x\rfloor(a \wedge b) &= (x \cdot a)b - (x \cdot b)a
\end{aligned}
$$

Geometrically when A, B are blades, $A\rfloor B$ is another blade contained in B and perpendicular to A with a norm proportional to the norms of A, B, and the projection of A on B. In addition the following relation between a vector and a blade is important:

$$x\rfloor A = 0 \Leftrightarrow x \perp y,\ \forall y : y \wedge A = 0$$

(i,e. $x\rfloor A = 0$ if and only if x is perpendicular to all vectors contained in A). In addition, the left contraction of same grade blades is identical to the scalar product of the blades:

$$A\rfloor B = A * B,\ \forall A, B \in B^n_k$$

A reversed version of the left contraction can be defined that is called the right contraction product (denoted by $B\lfloor A$ and pronounced as B contracted by A) where $\lfloor : B^n_r \times B^n_s \to B^n_{r-s}\ , r, s, r-s \in \{0, 1, \cdots, n\}$. The right contraction is related to the left contraction by:

$$B\lfloor A = (\widetilde{A}\rfloor \widetilde{B})^{\sim} = (-1)^{a(b+1)} A\rfloor B,\ a = odd(A), b = odd(b)$$

The duality formulas (8) and (9) can be written for the right contraction as:

$$\begin{aligned} C\lfloor(B\wedge A) &= (C\lfloor B)\lfloor A,\ \forall A, B, C \in B^n \\ C\lfloor(B\lfloor A) &= (C\lfloor B)\wedge A\ \forall A, B, C \in B^n, A \le C \end{aligned}$$

2.4 Orthogonality and Duality of Blades

Any blade $A \in B^n_k$ with non-zero norm $\|A\|^2 \neq 0$ (i.e. a non-null blade) can have an inverse blade A^{-1} with respect to the left contraction product (i.e. $A\rfloor A^{-1} = 1$) defined as:

$$A^{-1} = \frac{\widetilde{A}}{\|A\|^2} = \frac{(-1)^{k(k-1)/2}}{A * \widetilde{A}} A,\ k = grade(A)$$

This inverse is not unique with respect to the left contraction but is always present for blades like A. A special case is the inverse of a non-null vector given by $a^{-1} = \dfrac{a}{\|a\|^2}$. Hence for any blade with unit norm like the pseudo-scalar of a Euclidean space the inverse of the blade is its reverse $I^{-1} = I^{\sim}, \|I\| = 1$. For a general space with signature $(p, q, 0)$ the inverse of the pseudo-scalar is given by $I^{-1} = (-1)^q I^{\sim}$. Using the inverse of a blade a very important operation on blades can be defined that is called the dual of a blade $A \in B^n_r$ with respect to a larger blade $X \in B^n_s, s \ge r$ that is a mapping $* : B^n_r \times B^n_s \to B^n_{s-r}$ that acts as follows:

$$A^{*X} = A\rfloor X^{-1},\ \forall A \le X$$

When the larger blade is the space pseudo-scalar I the dual is simply written as $A^* = A\rfloor I^{-1}$. The geometric meaning of the dual A^* is simply a blade orthogonal to the original blade A such that they together complete the space; i.e. if A is the direct representation of the subspace $\overleftrightarrow{A}$ then A^* is the direct representation of $(\overleftrightarrow{A})^{\perp}$:

$$A \propto \overleftrightarrow{A} \Leftrightarrow A^* \propto (\overleftrightarrow{A})^{\perp}$$

Taking the dual for a blades two times results in the same blade with a weight change:

$$\begin{aligned} (A^{*X})^{*X} &= (A\rfloor X^{-1})\rfloor X^{-1} \\ &= A \wedge (X^{-1}\rfloor X^{-1}) \\ &= A \wedge (\frac{(-1)^{s(s-1)/2}}{\|X\|^2} X\rfloor X^{-1}) \\ &= \frac{(-1)^{s(s-1)/2}}{\|X\|^2} A \\ & \quad \forall A \in B^n_r, X \in B^n_s, A \le X \end{aligned}$$

Another related operation in a blade $A \le X$ called the undualization of the blade A with respect to the blade X can be defined as follows:

$$A^{\odot X} = A\rfloor X,\ \forall A \le X$$

Applying the undualization after the dualization (and similarly applying the dualization after the undualization) results in the original blade with no weight change: $(A^{*X})^{\odot X} = A\rfloor X^{-1})\rfloor X = A\wedge(X^{-1}\rfloor X) = A$. Using the duality formulas a duality relation can be found between the contraction products and the outer product for any two blades A, B:

$$
\begin{aligned}
(A \wedge B)^{*X} &= A \rfloor B^{*X} \\
(A \rfloor B)^{*X} &= A \wedge B^{*X} \\
& \forall A, B \leq X
\end{aligned}
$$

The above relations enable the use of another representation of subspaces called the dual representation of a subspace. Having a blade that is a direct representation of a subspace $A \propto \overleftrightarrow{A}$ then $x \in \overleftrightarrow{A} \Leftrightarrow x \wedge A = 0$. Now if the dual of A is defined the same relation between the vector x and the subspace $\overleftrightarrow{A}$ as can be expressed: $x \in \overleftrightarrow{A} \Leftrightarrow x \rfloor A^* = 0$. The blade A^* is thus called the dual representation of $\overleftrightarrow{A}$. This gives more flexibility in the choice of a representation blades for a given subspace. Using the dual of a blade a generalized linear projection operation of a general blade X onto a general blade with larger grade B can be defined as:

$$
\begin{aligned}
\mathbf{P}_B[X] &= (X \rfloor B)^{*B} \\
&= (X \rfloor B) \rfloor B^{-1} \\
&= (X \rfloor B^{-1}) \rfloor B
\end{aligned}
$$

The familiar geometric projection operation is algebraically non-linear because if projecting, say, a line on a plane the result is often another line, but may also be a point when they are perpendicular. Note that the exact weight of B is irrelevant in this definition (rolled out by the inverse of B) and only its attitude is used. Taking the undualization of both sides leads to $X \rfloor B = \mathbf{P}_B[X] \rfloor B$ which leads to defining the contraction geometrically in terms of the projection and the dualization as: the contraction $A \rfloor B$ is a sub-blade of B of grade $grade(B) - grade(A)$ dual with respect to B to the projection of A on B. The contraction is used as the base for defining the projection rather than the other way around because the contraction is linear in both arguments whereas the projection is only linear in the projected blade. Using the outer product and the dual the 3D cross product of vectors $a \times b$ can be generalized to any dimension by noting its geometric meaning and translating it to GA. The cross product of two vectors is simply the orthogonal complement of the homogeneous plane spanned by the two vectors. Thus $a \times b = (a \wedge b)^* = (a \wedge b) \rfloor I^{-1},\ \forall a, b \in \mathbb{R}^3$. This definition is independent of the dimension of the vector space and can be used with any two vectors of any GA with non-null pseudo scalar I.

As an application on the concepts in this section, a typical need is to express a vector $x \in \mathbb{R}^n$ as a linear combination of general (i.e. not necessarily orthogonal) basis vectors $\langle b_1, b_2, \cdots, b_n \rangle$. First an association of each basis vector b_i with a reciprocal vector is done, defined as $c_i = (-1)^{i-1}(b_1 \wedge b_2 \wedge \cdots \wedge b_{i-1} \wedge b_{i+1} \wedge \cdots \wedge b_n) \rfloor I^{-1},\ i = 1, 2, \cdots, n,\ I = b_1 \wedge b_2 \wedge \cdots \wedge b_n$. The basis $\langle b_1, b_2, \cdots, b_n \rangle$ and $\langle c_1, c_2, \cdots, c_n \rangle$ are easy to be shown mutually orthogonal $b_i \cdot c_j = \delta_i^j,\ \forall i, j = 1, 2, \cdots n$. The geometric meaning of a reciprocal basis vector c_i is the orthogonal complement of the span of all basis vectors except the basis vector b_i. Now to determine the coefficients x_i such that $x = x_1 b_1 + x_2 b_2 + \cdots x_n b_n$ the relation $x_i = x \cdot c_i$ (i.e. $x = \sum_{i=1}^{n} (x \cdot c_i) b_i$) is simply used. If the vector space is Euclidean with orthonormal basis then all basis vectors have a norm of $\|b_i\| = 1$ hence the reciprocal basis vector c_i is the same as the basis vector b_i. Generally, two reciprocal basis vectors are not co-linear $b_i \wedge c_i \neq 0$ however the following relation holds: $\sum_{i=1}^{n} b_i \wedge c_i = 0$.

3 Linear Transformations of Sub-spaces: Changing Blades

3.1 Linear Transforms of Vector Spaces

In linear algebra a linear operator on a real vector space $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a general mapping that acts on vectors to produce other vectors such that: $\mathbf{f}[\alpha x] = \alpha \mathbf{f}[x]$, $\mathbf{f}[x + y] = \mathbf{f}[x] + \mathbf{f}[y]$, $\forall \alpha \in \mathbb{R},\ x, y \in \mathbb{R}^n$. Classically the concept of a linear operators is associated with matrices through the following construction: assuming the vector space has the general basis $\boldsymbol{E} = \langle e_1, e_2, \cdots, e_n \rangle$ any vector $x \in \mathbb{R}^n$ can be expressed as a linear combination of the vectors in $\boldsymbol{E}$ as $x = x_1 e_1 + \cdots + x_n e_n$. The representation of the vector x with respect to the basis $\boldsymbol{E}$ is defined as the 1-column matrix $[x]_{\boldsymbol{E}} = \boldsymbol{Rep}_{\boldsymbol{E}}[x] = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^T$. Having a linear operator defined $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ such that the effect of $\mathbf{f}$ on the basis vectors of $\boldsymbol{E}$ is known and expressed as linear combination of another general basis $\boldsymbol{B} = \langle b_1, b_2, \cdots, b_n \rangle$ (i.e. $\boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}[e_i]]$ is known for all e_i), the matrix of $\mathbf{f}$ acting on $\boldsymbol{E}$ with respect to $\boldsymbol{B}$ is defined as:

$$[\mathbf{f}]_{\boldsymbol{E},\boldsymbol{B}} = \boldsymbol{Rep}_{\boldsymbol{E},\boldsymbol{B}}[\mathbf{f}] = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \end{bmatrix},\ f_i = \boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}[e_i]]$$

Hence the transformation of any vector can be found through the following simple matrix multiplication relation:

$$\begin{aligned}
\boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}[x]] &= \boldsymbol{Rep}_{\boldsymbol{B}}[x_1\mathbf{f}[e_1] + x_2\mathbf{f}[e_2] + \cdots + x_n\mathbf{f}[e_n]] \\
&= x_1\boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}[e_1]] + x_2\boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}[e_2]] + \cdots + x_n\boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}[e_n]] \\
&= \boldsymbol{Rep}_{\boldsymbol{E},\boldsymbol{B}}[\mathbf{f}]\boldsymbol{Rep}_{\boldsymbol{E}}[x] \\
\Rightarrow [\mathbf{f}[x]]_{\boldsymbol{B}} &= [\mathbf{f}]_{\boldsymbol{E},\boldsymbol{B}}\,[x]_{\boldsymbol{E}}
\end{aligned}$$

When the two basis are the same ($\boldsymbol{E} = \boldsymbol{B}$) the relation becomes $[\mathbf{f}]_{\boldsymbol{E}} = \boldsymbol{Rep}_{\boldsymbol{E}}[\mathbf{f}] = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \end{bmatrix},\ f_i = \boldsymbol{Rep}_{\boldsymbol{E}}[\mathbf{f}[e_i]],\ [\mathbf{f}[x]]_{\boldsymbol{E}} = [\mathbf{f}]_{\boldsymbol{E}}\,[x]_{\boldsymbol{E}}\, \forall x \in \mathbb{R}^n$. The unique matrix $[\mathbf{f}]_{\boldsymbol{E},\boldsymbol{B}}$ is called the matrix representation of $\mathbf{f}$ with respect to basis $\boldsymbol{E}$ and $\boldsymbol{B}$. Many properties of the linear transform can then be found by matrix algebra. For example the linear transform is invertible if and only if the matrix is non-singular. The determinant of the linear operator is proportional to the determinant of the matrix. Another example is the adjoint linear transform $\overline{\mathbf{f}}$ related to a linear operator $\mathbf{f}$ defined on a real inner product space $\mathbb{R}^n$ with bilinear form $\mathbf{B}$ is the transform that satisfies $\mathbf{B}(\mathbf{f}[x], y) = \mathbf{B}(x, \overline{\mathbf{f}}[y])\ \forall x \in \mathbb{R}^n$. The matrix representation of $\overline{\mathbf{f}}$ is the transpose of the matrix representation of $\mathbf{f}$: $\left[\overline{\mathbf{f}}\right]_{\boldsymbol{E}} = ([\mathbf{f}]_{\boldsymbol{E}})^T$. In addition, two matrices are called similar if they represent the same linear operator. An invertible linear operator $\mathbf{f}$ that satisfies $\mathbf{B}(\mathbf{f}[x], \mathbf{f}[y]) = \mathbf{B}(x, y)\ \forall x, y \in \mathbb{R}^n$ is called an orthogonal operator. An orthogonal operator $\mathbf{f}$ preserves the inner product between vectors (hence it preserves lengths and angles) and its adjoint is equal to its inverse: $\mathbf{f^{-1}} = \overline{\mathbf{f}}$. The concept of linear operator on vector spaces can be generalized to linear transforms where the domain and co-domain spaces are different; perhaps even having different dimensions resulting in non-square transformation matrices. GA provides a coordinate free alternative for studying and extending linear operators and transformations without the use of matrices.

3.2 Applying Linear Transforms to Blades: Outermorphisms

The concept of a linear operator can be extended to act on whole subspaces by applying $\mathbf{f}$ to the basis vectors of the subspace and reconstructing the transformed subspace afterwards. An alternative approach is possible using the algebraic constructions of GA through extending the linear operator to act on arbitrary blades, by constructing what is called an outermorphism, as follows:

$$\begin{aligned}
\mathbf{f} : B^n &\to B^n \\
\mathbf{f}[\alpha] &= \alpha,\ \forall \alpha \in B^n_0 \\
\mathbf{f}[A + B] &= \mathbf{f}[A] + \mathbf{f}[B],\ \forall A, B \in B^n_k \\
\mathbf{f}[X \wedge Y] &= \mathbf{f}[X] \wedge \mathbf{f}[Y],\ \forall X, Y \in B^n
\end{aligned}$$

An extension of a map of "vectors to vectors" in this manner to the whole of the Grassmann algebra is called extension as a (linear) outermorphism, since the third property shows that a morphism (i.e., a mapping) is obtained that commutes with the outer product. Outermorphisms have nice algebraic properties that are essential to their geometrical usage:

- Blades Remain Blades: Geometrically, oriented subspaces are transformed to oriented subspaces of the same grade: $grade(A) = grade(\mathbf{f}[A]),\ \forall A \in B^n$. This means that the dimensionality of subspaces do not change under a linear transformation.

- Preservation of Factorization. If two blades A, B have a blade C in common then the blades $\mathbf{f}[A], \mathbf{f}[B]$ have $\mathbf{f}[C]$ in common. Hence the meet of subspaces is preserved under the linear transform.

The determinant of a linear operator $\mathbf{f}$ is a fundamental scalar property of $\mathbf{f}$ defined implicitly as: $\mathbf{f}[I] = det(\mathbf{f})I$. It signifies the change in weight between the pseudo-scalar of the space I and its transformed version under $\mathbf{f}$ which is the original definition of determinants in linear algebra. using this definition it is easy to show properties of determinants of linear transforms such as $det(\mathbf{g} \circ \mathbf{f}) = det(\mathbf{g})det(\mathbf{f})$. This is all done without using matrices and coordinates as usually applied in linear algebra. Another important concept in linear algebra is the adjoint of a linear operator $\mathbf{f}$ denoted by $\overline{\mathbf{f}}$. For any linear operator $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ defined on a real vector space $\mathbb{R}^n$ having arbitrary (not necessarily orthogonal) basis $\langle b_1, b_2, \cdots, b_n\rangle$ the adjoint operator is defined using the reciprocal basis $\langle c_1, c_2, \cdots, c_n\rangle$as:

$$\begin{aligned}\overline{\mathbf{f}} : \mathbb{R}^n &\rightarrow \mathbb{R}^n \\ \overline{\mathbf{f}}[x] &= \sum_{i=1}^{n} (x \cdot \mathbf{f}[b_i]) c_i, \ \forall x \in \mathbb{R}^n\end{aligned}$$

Now the outermorphism of the adjoint can be constructed using the outer product. The adjoint outermorphism satisfies the following relations for all blades:

$$\begin{aligned}\mathbf{f}[A] * B &= A * \overline{\mathbf{f}}[B] \ \forall A, B \in B^n \\ \overline{\overline{\mathbf{f}}} &= \mathbf{f} \\ \overline{\mathbf{f^{-1}}} &= \overline{\mathbf{f}}^{-\mathbf{1}}\end{aligned}$$

Applying an outermorphisms to the scalar product is simple since it always produces a scalar:

$$\mathbf{f}[A * B] = A * B$$

For the left contraction product the relation is:

$$\mathbf{f}[A \rfloor B] = \overline{\mathbf{f}}^{-\mathbf{1}}[A] \rfloor \mathbf{f}[B]$$

And in the case that $\mathbf{f}$ is an orthogonal operator the relation becomes simpler (i.e. $\mathbf{f}$ is also an innermorphism in addition to being an outermorphism):

$$\mathbf{f}[A \rfloor B] = \mathbf{f}[A] \rfloor \mathbf{f}[B]$$

When transforming a blade X that directly represents a subspace $\overleftrightarrow{X}$ using a linear operator $\mathbf{f}$ the dual X^*that dualy represents $\overleftrightarrow{X}$ is also transformed under a related operator that shall be called the dual operator of $\mathbf{f}$ and denote it by $\mathbf{f}^*$ that is generally not equal to $\mathbf{f}$. The relation between the two operators must be designed such that the represented subspace remains the same along with its weight: $\mathbf{f}^*[X^*] = (\mathbf{f}[X])^* \forall X \in B^n$. This restriction is to ensure the consistency of the represented subspace and directly results in the following relation between the two linear operators:

$$\mathbf{f}^*[X^*] = det(\mathbf{f})\overline{\mathbf{f}}^{-\mathbf{1}}[X^*]$$

This relation is simplified for orthogonal transforms as:

$$\begin{aligned}\mathbf{f}^*[X^*] &= det(\mathbf{f})\mathbf{f}[X^*] \\ &= \pm\mathbf{f}[X^*]\end{aligned}$$

Since $\mathbf{f}^*[X^*]$ is not generally equal to $\mathbf{f}[X^*]$, it implies that blades that are intended as dual representations of subspaces do not transform in the same way as blades that are intended as direct representations. In a proper representation of geometry, therefore a blade usage interpretation is needed before a linear operator is applied. The last relation enables us to find an expression for the inverse of a linear operator; if it exists:

$$\begin{aligned}\mathbf{f^{-1}}[A] &= \frac{1}{det(\mathbf{f})} \left(\overline{\mathbf{f}}[A^*]\right)^{\odot} \\ &= \frac{1}{det(\mathbf{f})} \overline{\mathbf{f}}[A \rfloor I^{-1}] \rfloor I\end{aligned}$$

Although this expression uses duality it is not a metric expression because the two dualities cancel each other hence any metric can be assumed for computing the outermorphism (preferably the Euclidean metric).

3.3 Matrix Representation of Outermorphisms

Having a linear transform on a real vector space $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with general basis $\boldsymbol{B} = \langle b_1, b_2, \cdots, b_n \rangle$ and reciprocal basis $\boldsymbol{C} = \langle c_1, c_2, \cdots, c_n \rangle$, any vector $x \in \mathbb{R}^n$ has a matrix representation $[x]_{\boldsymbol{B}} = \boldsymbol{Rep}_{\boldsymbol{B}}[x] = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^T$ where the components x_i can be found using the reciprocal basis as $x_i = x \cdot c_i$. In addition, the linear operator has a matrix representation $[\mathbf{f}]_{\boldsymbol{B}} = \boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}] = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \end{bmatrix}$ where the components $f_j = \boldsymbol{Rep}_{\boldsymbol{B}}[\mathbf{f}[b_j]] = \begin{pmatrix} f_{1j} & f_{2j} & \cdots & f_{nj} \end{pmatrix}^T$ are column vectors having the following structure:

$$\begin{aligned} f_{ij} &= \mathbf{f}[b_j] \cdot c_i \\ &= (b_1 \wedge b_2 \wedge \cdots \wedge b_{i-1} \wedge \mathbf{f}[b_j] \wedge b_{i+1} \wedge \cdots \wedge b_n) \rfloor I^{-1} \\ I &= b_1 \wedge b_2 \wedge \cdots \wedge b_n \\ & \forall i, j = 1, 2, \cdots, n \end{aligned}$$

The same construction can be generalized to k-vectors by using the k-vector basis blades $\boldsymbol{B}_{\boldsymbol{k}}^{\boldsymbol{n}} = \left\langle b_1^k, b_2^k, \cdots, b_r^k \right\rangle, r = \begin{pmatrix} n \\ k \end{pmatrix}$. In this case the reciprocal basis k-vectors are defined as $\boldsymbol{C}_{\boldsymbol{k}}^{\boldsymbol{n}} = \left\langle c_1^k, c_2^k, \cdots, c_r^k \right\rangle, r = \begin{pmatrix} n \\ k \end{pmatrix}$ where $c_i^k = (c_{j_1} \wedge c_{j_2} \wedge \cdots \wedge c_{j_k})^{\sim}$ when $b_i^k = b_{j_1} \wedge b_{j_2} \wedge \cdots \wedge b_{j_k}$. The matrix of the outermorphism for the k-vectors is constructed as $[\mathbf{f}]_{\boldsymbol{B}_{\boldsymbol{k}}^{\boldsymbol{n}}} = \boldsymbol{Rep}_{\boldsymbol{B}_{\boldsymbol{k}}^{\boldsymbol{n}}}[\mathbf{f}] = \begin{bmatrix} f_1^k & f_2^k & \cdots & f_r^k \end{bmatrix}$ where the components $f_j^k = \boldsymbol{Rep}_{\boldsymbol{B}_{\boldsymbol{k}}^{\boldsymbol{n}}}[\mathbf{f}[b_j^k]] = \begin{pmatrix} f_{1j}^k & f_{2j}^k & \cdots & f_{nj}^k \end{pmatrix}^T$ are column vectors having the values $f_{ij}^k = \mathbf{f}[b_j^k] * c_i^k$. All these $(n+1)$ matrices can be combined as blocks in a single $2^n \times 2^n$ matrix that acts on the representation of any multivector with respect to the 2^n basis blades of the full GA $\bigwedge^n$.

4 Intersection and Union of Sub-spaces: The Meet and Join of Blades

Given two subspaces $\overleftrightarrow{A}, \overleftrightarrow{B}$ of the overall vector space $\mathbb{R}^n$, the largest subspace common to both of them is called the meet of those subspaces, and as a set is the intersection $\overleftrightarrow{A} \cap \overleftrightarrow{B}$ of those subspaces. The join of the two given subspaces is the smallest super-space common to both of them, and as a set is the sum $\overleftrightarrow{A} + \overleftrightarrow{B} = \left\{ x_1 + x_2 : x_1 \in \overleftrightarrow{A}, x_2 \in \overleftrightarrow{B} \right\}$ of those subspaces. $\overleftrightarrow{A} + \overleftrightarrow{B}$ is not usually a "direct sum" $\overleftrightarrow{A} \oplus \overleftrightarrow{B}$, as the decomposition $x_1 + x_2$ of a nonzero element of the join is uniquely determined only when $A \cap B = 0$. The system of subspaces, with its subset partial ordering and meet and join operations, is an example of the type of algebraic system called a "lattice". Blades needs to be defined that directly represent the meet $M = A \cap B$ and join $J = A \cup B$ of two subspaces $\overleftrightarrow{A}, \overleftrightarrow{B}$ given their direct representation blades A, B.

The meet $M = A \cap B$ of two blades A, B can be defined as any blade that satisfies:

$$\forall x \in \mathbb{R}^n : [x \wedge M = 0] \Leftrightarrow [x \wedge A = 0, x \wedge B = 0]$$

While the join $J = A \cup B$ can be defined as any blade that satisfies

$$\forall x \in \mathbb{R}^n : [x \wedge J = 0] \Leftrightarrow [\exists x_1 \exists x_2 : x = x_1 + x_2, x_1 \wedge A = 0, x_2 \wedge B = 0]$$

The two conditions defining the meet and join are restrictions on the attitudes of the blades M, J but not on their weight. Thus if a blade M satisfies the meet condition then any blade $\lambda M, \lambda \in \mathbb{R}$ will also satisfy the same condition leading to an infinite number of representing blades with common attitude; and the same can be said for the join. Geometrically it is advantageous to define the meet of two blades with respect to their join or to define the join with respect to their meet. This geometric constraint, which is not more than a reasonable convention, leads to any of the following three equivalent algebraic constraints:

$$\begin{aligned} J &= (A \lfloor M^{-1}) \wedge M \wedge (M^{-1} \rfloor B) \\ &= A \wedge (M^{-1} \rfloor B) \\ &= (A \lfloor M^{-1}) \wedge B \\ M &= (B \rfloor J^{-1}) \rfloor A \\ &= B^{*J} \rfloor A \\ &= B \lfloor (J^{-1} \lfloor A) \\ M \rfloor J^{-1} &= (B \rfloor J^{-1}) \wedge (A \rfloor J^{-1}) \Leftrightarrow M^{*J} = B^{*J} \wedge A^{*J} \end{aligned}$$

These algebraic constraints relate any change in the weight of the meet to the weight of the join (and vice versa) such that if any of them is multiplied by a scalar λ , the other must be multiplied by $1/\lambda$.

The two given conditions for the meet and join are obviously metric-independent, but notice that the above constraints use metric GA products. The key is that the actual metric is irrelevant and a Euclidean metric can be assumed for computations without affecting the final results. In these constraints the blades A, B are assumed to be factored such that $A = \acute{A} \wedge M$ and $B = M \wedge \acute{B}$ leading to $J = \acute{A} \wedge M \wedge \acute{B}$, $\acute{A} = A \lfloor M^{-1}$, $\acute{B} = M^{-1} \rfloor B$.

If an invertible outermorphism $\mathbf{f}$ is applied to the meet M and join J of two blades A, B determining their relations to the meet $\bar{M}$ and join $\bar{J}$ of the transformed blades $\bar{A} = f[A], \bar{B} = f[B]$ might be required. using the definition of meet and join, it is possible to proving that if setting $\bar{M} = \mathbf{f}[M], \bar{J} = \mathbf{f}[J]$ then $\bar{M}$ and $\bar{J}$ are "a meet and a join" for $\bar{A}, \bar{B}$ for any meet and join (i.e. regardless of the above geometric constraint convention). Now introducing the geometric constraint on M, J, it is possible to prove that $\bar{M} = \mathbf{f}[M], \bar{J} = \mathbf{f}[J]$ satisfy the same geometric constraint automatically. The following are 3 options for selecting the weight of the 4 blades $J, M, \bar{J}, \bar{M}$:

1. Scaling the meet M by λ the join J is scaled by $1/\lambda$ (according to the geometric constraint) thus $\bar{J}$ will be scaled by $1/\lambda$ (because $\mathbf{f}[J/\lambda] = \mathbf{f}[J]/\lambda$) and $\bar{M}$ is scaled by λ.
2. The norm of the join is required to be of a specific value (say α) so $\mathbf{f}[J]$ is re-scaled such that the join and meet of the transformed blades are now defined as $\bar{J} = \frac{\alpha}{\|\mathbf{f}[J]\|}\mathbf{f}[J]$, $\bar{M} = \frac{\|\mathbf{f}[J]\|}{\alpha}\mathbf{f}[M]$.
3. If J is an eigen blade of $\mathbf{f}$ (i.e. J and $\mathbf{f}[J]$ have the same attitude) then $\mathbf{f}[J] = \lambda J$ hence defining $\bar{J} = J = \mathbf{f}[J]/\lambda$, $\bar{M} = \lambda \mathbf{f}[M]$.

5 The Fundamental Product of Geometric Algebra: The Geometric Product

5.1 Defining the Geometric Product

Having a fixed and known vector $a \in \mathbb{R}^n$, a vector x is to be found given its inner product with a is knows: $a \cdot x = \alpha$ The problem here is that an infinite number of vectors satisfy the given relation thus there can be non-unique inverse for a with respect to the inner product. A similar situation is with the outer product when having the relation $a \wedge x = B$ where a, B are known. Thus there is no inverse with respect to the outer product either. But if both relations are true there can be only one vector x that satisfies both equations. This is the main motivation behind defining a invertible product between vectors called the geometric product as: $ax = a \cdot x + a \wedge x$ that results in a multivector rather than a scalar or bi-vector alone. The geometric product is bilinear, associative, and distributive over addition but not commutative nor anti-commutative. To generalize the geometric product to higher-grade blades, an orthogonal basis is defined for $\mathbb{R}^n$: $\langle e_1, e_2, \cdots, e_n \rangle$ with a symmetric bilinear form $\mathbf{B}$ and on such basis. This leads to $e_i e_i = e_i \cdot e_i + e_i \wedge e_i = \mathbf{B}(e_i, e_i)\, \forall i$ and $e_i e_j = e_i \cdot e_j + e_i \wedge e_j = -e_j \wedge e_i = -e_j e_i \,\forall i \neq j$. Using these two relations a table can be constructed to compute all values of $e_i e_j$ for all $i, j = 1, 2, \cdots, n$ (that will always give a weighted basis blades) and using the linearity, distributivity, and associativity of the geometric product any geometric product of multivectors can be computed easily. With the geometric product any non-null vector $a \in \mathbb{R}^n$ has the unique inverse: $a^{-1} = \dfrac{1}{\|a\|^2} a$ (i.e. a vector in the same direction of a but properly scaled to make $a \cdot a^{-1} = 1$).

The geometric product is related to the inner and outer products of blades through the following relations (that can be extended by linearity to products of vectors with general multivectors):

$$\begin{aligned} a \wedge B &= \frac{1}{2}(aB + \widehat{B}a) \\ B \wedge a &= \frac{1}{2}(Ba + a\widehat{B}) \\ a \rfloor B &= \frac{1}{2}(aB - \widehat{B}a) \\ B \lfloor a &= \frac{1}{2}(Ba - a\widehat{B}) \\ & \forall a \in B_0^n, B \in B^n \end{aligned}$$

Another alternative to computing the outer product and the metric products is to compute the geometric product then extract the appropriate grades from the result using the following relations:

$$\begin{aligned} A \wedge B &= \langle AB \rangle_{r+s} \;\forall A \in B_r^n, B \in B_s^n, r+s \leq n \\ A \rfloor B &= \langle AB \rangle_{s-r} \;\forall A \in B_r^n, B \in B_s^n, 0 \leq s-r \leq n \\ A \lfloor B &= \langle AB \rangle_{r-s} \;\forall A \in B_r^n, B \in B_s^n, 0 \leq r-s \leq n \\ A * B &= \langle AB \rangle_0 \;\forall A \in B_k^n, B \in B_k^n \end{aligned}$$

These relations can be generalized to k-vectors and multivectors using linearity and are the basis of implementing all bilinear products using the geometric product alone. In addition, using the geometric product produces an alternative definition for the projection of blade X onto a general blade with larger grade B as: $\mathbf{P}_B[X] = (X \rfloor B)B^{-1}$.

5.2 Representing Orthogonal Transformations: Computing with Versors

Using the geometric product, a definition for a powerful GA-based representation for orthogonal transformations can be made. This representation, alternative to real matrices, is called a versor. Geometrically, any orthogonal transformation in $\mathbb{R}^n$ is equivalent to a collection of simple reflections on (n-1)-dimensional hyper-planes. Algebraically, a reflection on a hyper-plane dually represented by a non-null vector $a \in \mathbb{R}^n$ is defined as:

$$\mathbf{L}_a[X] = (-1)^x a X a^{-1} \,\forall X \in B^n, x = odd(X)$$

In this expression the actual weight (norm) of a is irrelevant since it is canceled by the inverse in a^{-1}. If the vector a is used as a direct representation for a line (a 1D subspace) then the operation performed here is a reflection in a line rather than a reflection in a hyper-plane as:

$$\acute{\mathbf{L}}_a[X] = a X a^{-1} \,\forall X \in B^n$$

Geometrically, a reflection in a line is not a simple reflection because it is actually a composition of (n-1) reflections on (n-1)-dimensional hyper-planes in $\mathbb{R}^n$. For example a reflection in the line passing through the x-axis in $\mathbb{R}^3$ is actually two reflections: one in the xy-plane followed by one in the xz-plane thus it is actually rotation around the x-axis by 180 degrees. A rotation in $\mathbb{R}^n$ is geometrically an even number of hyper-plane reflections in $\mathbb{R}^n$. An odd number of such reflection is called an anti-rotation. Thus a set of consecutive simple reflections for the blade X on k hyper-planes dually represented by vectors $v_1, v_2, \cdots, v_k$ can be written as:

$$\begin{aligned} \mathbf{V}[X] &= (-1)^{kx} v_k \cdots v_2 v_1 X v_1 v_2 \cdots v_k \\ &= (-1)^{kx} V X V^{-1} \\ & \quad V = v_1 v_2 \cdots v_k \in \bigwedge^n, \; X \in B^n, \; x = odd(X), k = odd(V) \\ \Rightarrow \mathbf{V}[x] &= (-1)^k V x V^{-1} \\ &= \hat{V} x V^{-1} \,\forall x \in \mathbb{R}^n \end{aligned}$$

The multivector $V = v_1 v_2 \cdots v_k \in \bigwedge^n$ is called a versor and is essentially an even or odd multivector created by the geometric product of the non-null vectors. The bilinear product $\mathbf{V}_V[X] = (-1)^{kx} V X V^{-1}$is called a versor product and is actually an outermorphism created from an orthogonal transform on vectors. This versor product can be applied to any even or odd multivector $X \in \bigwedge^n$ (thus to any other versor) using the same formula. This means that not only subspaces can be transformed by versors but also versors (i.e. orthogonal transformations themselves) can be transformed by versors. In addition, the composition of two orthogonal transformations $L_{V_2} \circ L_{V_1}$ represented by versors V_1, V_2 is the geometric product of the two versors as can be clearly seen in the case of vectors $x \in \mathbb{R}^n$:

$$\begin{aligned} L_{V_2} \circ L_{V_1}[x] &= L_{V_2}[L_{V_1}[x]] \\ &= \hat{V}_2(\hat{V}_1 x V_1^{-1}) V_2^{-1} \\ &= (\hat{V}_2 \hat{V}_1) x (V_1^{-1} V_2^{-1}) \\ &= \hat{V} x V^{-1}, \; V = V_2 V_1 \\ & \quad \forall x \in \mathbb{R}^n \end{aligned}$$

This simply means that orthogonal transformations can be concatenated or composed using the geometric product of their representing versors as V_2V_1, and orthogonal transformations can also be nested using the sandwiching versor product of one representing versor to the other as $(-1)^{v_1v_2}V_2V_1V_2^{-1}$. Thus orthogonal transformations are themselves objects to be transformed by other transformations.

Any even versor represents a rotation which is an orthogonal transformation that has a determinant of 1 (i.e. it preserves handedness of the pseudo-scalar I). Any odd versor represents an anti-rotation (or a reflection) which is an orthogonal transformation that has a determinant of -1 (i.e. it changes handedness of the pseudo-scalar I). This result is independent of the used metric. If an orthogonal transform $\mathbf{L}$ is represented by a versor V then the inverse transform $\mathbf{L}^{-1}$is represented by V^{-1}. In addition, the versor product $\mathbf{V}_V[X] = (-1)^{kx}VXV^{-1}$(being both an outermorphism and an innermorphism) preserves all bilinear products of geometric algebra: the outer, metric, and geometric products:

$$\begin{aligned} \mathbf{V}_V[\alpha A] &= \alpha A \\ \mathbf{V}_V[A+B] &= \mathbf{V}_V[A] + \mathbf{V}_V[B] \\ \mathbf{V}_V[A \wedge B] &= \mathbf{V}_V[A] \wedge \mathbf{V}_V[B] \\ \mathbf{V}_V[A * B] &= \mathbf{V}_V[A] * \mathbf{V}_V[B] = A * B \\ \mathbf{V}_V[A \rfloor B] &= \mathbf{V}_V[A] \rfloor \mathbf{V}_V[B] \\ \mathbf{V}_V[A \lfloor B] &= \mathbf{V}_V[A] \lfloor \mathbf{V}_V[B] \\ \mathbf{V}_V[AB] &= \mathbf{V}_V[A]\mathbf{V}_V[B] \\ & \forall A, B \in \bigwedge^n, \alpha \in \bigwedge_0^n \end{aligned}$$

This is a very important result since it really means that any algebraic construction based on the above operations can be transformed directly under an orthogonal transform in a structure-preserving manner. Meaning that transforming the components and then creating the structure is equivalent to creating the structure and then applying the orthogonal transform to the whole geometric structure (may it be an oriented subspace or an orthogonal transform by itself). A direct consequence of this structure preserving property is that the transform of the meet\join of two blades is the meet\join of the transformed blades:

$$\begin{aligned} \mathbf{V}_V[A \cap B] &= \mathbf{V}_V[A] \cap \mathbf{V}_V[B] \\ \mathbf{V}_V[A \cup B] &= \mathbf{V}_V[A] \cup \mathbf{V}_V[B] \\ & \forall A, B \in B^n \end{aligned}$$

5.3 Reflecting and Projecting on Sub-spaces: Blades as Operators

Having a non-null blade that is a direct representation of a subspace $A \propto \overleftrightarrow{A}, \|A\|^2 \neq 0$ and its dual that is a dual representation for the same subspace $A^* = B \propto (\overleftrightarrow{A})^{\perp}$either blades can be used as a reflection operator for any vector $x \in \mathbb{R}^n$ to geometrically perform a reflection of x by the subspace $\overleftrightarrow{A}$ using one of the following operations:

$$\begin{aligned} \mathbf{L}_{\overleftrightarrow{A}}[x] &= (-1)^{a+1}AxA^{-1} \\ &= (-1)^{b}BxB^{-1} \end{aligned}$$

This reflection can be generalized to an outermorphism to act on any k-blade that directly represents another subspace $X \propto \overleftrightarrow{X}$ as:

$$\begin{aligned} \mathbf{L}_{\overleftrightarrow{A}}[X] &= (-1)^{x(a+1)}AXA^{-1} \\ &= (-1)^{xb}BXB^{-1} \end{aligned}$$

where $a = odd(A), b = odd(B), x = odd(X)$. If the blade to be transformed is itself a dual representation for a subspace $X^* = Y \propto (\overleftrightarrow{X})^{\perp}$the relations become:

$$
\begin{aligned}
\mathbf{L}^*_{\overleftrightarrow{A}}[Y] &= (-1)^{(n-1)(y+1)(a+1)} A Y A^{-1} \\
&= (-1)^{(y+1)b} B Y B^{-1}
\end{aligned}
$$

Where $y = odd(Y)$. The result $\mathbf{L}^*_{\overleftrightarrow{A}}[Y]$ in this case is itself a dual representation for the reflected blade. Using these relations the reflection of any direct or dual blade into another direct or dual blade can be computed to obtain a direct or dual blade with correct orientation, weight, and attitude. If only the attitude is required the simple relation $\mathbf{L}_{\overleftrightarrow{A}}[X] = AXA^{-1}$is sufficient. In the same sense of this type of "sandwiching product" a direct blade $A \propto \overleftrightarrow{A}$ can be used to construct a linear projection operator using the contraction:

$$
\mathbf{P}_{\overleftrightarrow{A}}[X] = (-1)^{x(a+1)} A \lfloor (X \rfloor A^{-1})
$$

Since any non-null k-blade A is simply an outer product of LID vectors $a_1 \wedge a_2 \wedge \cdots \wedge a_k$, any orthogonalization technique could be used to find a set of LID orthogonal vectors that create the same blade $A = b_1 \wedge b_2 \wedge \cdots \wedge b_k$, $b_i \cdot b_j = 0 \, \forall i \neq j$. The outer product of any set of LID orthogonal vectors is equal to the geometric product of such set of vectors $A = a_1 \wedge a_2 \wedge \cdots \wedge a_k = b_1 \wedge b_2 \wedge \cdots \wedge b_k = b_1 b_2 \cdots b_k$ thus any non-null k-blade can be expressed as a geometric product of k LID orthogonal vectors; i.e. any non-null blade is actually a versor. This adds to blades more representative power: a blade can (directly or dually) represent a subspace, a projection operator, a dualization operator, or an orthogonal transform (namely, a reflection in a subspace represented by the same blade).

6 Rotating Sub-spaces: Computing with Rotors

6.1 A Rotor as the Ratio of Two Unit Vectors: Subsuming Complex Numbers

Any rotation can be geometrically represented by an even number of simple reflections. A rotor R is an even versor normalized such that $R\widetilde{R} = 1 \Leftrightarrow R^{-1} = \widetilde{R}$ this leads to defining a rotation transformation as:

$$
\mathbf{R}_R[X] = RX\widetilde{R} \; \forall X \in \bigwedge^n
$$

The simplest of rotors is the ratio of two unit vectors:

$$
\begin{aligned}
R &= ab^{-1}, a, b \in \mathbb{R}^n, \|a\|^2 = 1, \|b\|^2 = 1 \\
&= a \cdot b + a \wedge b \\
&= \cos(\phi/2) - I \, \sin(\phi/2)
\end{aligned}
$$

Where $\phi/2$ is the angle from a to b and I is the unit pseudo-scalar of the $a \wedge b$ plane. The rotor R in this case can be used to rotate any multivector by an angle ϕ where $0 \leq \phi \leq 2\pi$. The quantity $I\phi$ is called the bi-vector angle and it contains all information relevant to the rotation. An interesting fact relates complex numbers to rotors in a plane: complex numbers $x + iy$ are isomorphic to ratios of 2D vectors to a fixed vector in the 2D plane (usually the unit vector for the real axis). If a vector $a = \alpha e_1 + \beta e_2$ is defined on the Euclidean orthonormal basis $\langle e_1, e_2 \rangle$ then the quantity $c = ae_1^{-1} = \alpha + \beta(e_2 \wedge e_1) = \alpha + \beta(e_2 e_1)$ represents a complex number $\alpha + i\beta$ that geometrically corresponds to a with the usual complex number multiplication taken as the geometric product. Thus i can be interpreted geometrically as the unit bi-vector $e_2 e_1$ and indeed $i^2 = (e_2 e_1)^2 = -1$. The sum and product of two "complex numbers" $A = \alpha_1 + \alpha_2(e_2 e_1)$, $B = \beta_1 + \beta_2(e_2 e_1)$ are equivalent to the usual complex number addition and multiplication:

$$
\begin{aligned}
A + B &= \alpha_1 + \alpha_2(e_2 e_1) + \beta_1 + \beta_2(e_2 e_1) \\
&= (\alpha_1 + \beta_1) + (\alpha_2 + \beta_2)(e_2 e_1) \\
AB &= [\alpha_1 + \alpha_2(e_2 e_1)][\beta_1 + \beta_2(e_2 e_1)] \\
&= (\alpha_1 \beta_1 - \alpha_2 \beta_2) + (\alpha_1 \beta_2 + \alpha_2 \beta_1)(e_2 e_1)
\end{aligned}
$$

This gives a real geometric interpretation for complex numbers and can be used to extend and explain all complex number based constructions using GA.

6.2 Rotors in 3D: Subsuming Unit Quaternions

In $\mathbb{R}^3$ there is an algebraic method for representing rotations of vectors called quaternions. A quaternion $Q = q_0 + q = q_0 + q_1 i + q_2 j + q_3 k$ is a scalar q_0 plus a "vector" $q = q_1 i + q_2 j + q_3 k$ defined on a basis $\langle i, j, k\rangle$ having the defining multiplication rules: $i^2 = j^2 = k^2 = ijk = -1$. A unit quaternion satisfies $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The "vector" part of a unit quaternion is traditionally interpreted geometrically as a kind of rotation axis expressed in a 3D "imaginary" basis $\langle i, j, k\rangle$ with elements that square to -1. As with complex numbers there is an isomorphism between rotors in $\bigwedge^3$ with Euclidean orthonormal basis $\langle e_1, e_2, e_3\rangle$ and unit quaternions defined as:

$$\begin{aligned}
i &= Ie_1 = e_2 e_3 \\
j &= Ie_2 = e_3 e_1 \\
k &= Ie_3 = e_1 e_2 \\
& \quad , I = e_1 e_2 e_3
\end{aligned}$$

Thus the basis i, j, k are not basis vectors but basis bi-vectors for rotations and the component q is actually a bi-vector representing the rotation plane of the quaternion Q that can now be expressed as:

$$\begin{aligned}
Q &= q_0 + Iq \\
&= q_0 + I(q_1 e_1 + q_2 e_2 + q_3 e_3) \\
&= q_0 + q_1 e_2 e_3 + q_2 e_3 e_1 + q_3 e_1 e_2
\end{aligned}$$

In this case the quaternion multiplication of two quaternions $Q = q_0 + Iq$ and $P = p_0 + Ip$ used to compose rotations is:

$$QP = [q_0 p_0 - \langle qp\rangle_0] + I\left[p_0 q + q_0 p + (q \wedge p)I^{-1}\right]$$

Quaternions are thus just special cases of GA rotors in $\mathbb{R}^3$.

6.3 The Exponential Representation of Rotors

In a vector space with a Euclidean metric any geometric product of an even number of unit vectors is a rotor. In other metrics this is not the case unless the rotor R satisfies $R\widetilde{R} = 1$. Even in this case there exists a distinction between rotors that are "continuously connected to the identity" and those that are not. This property implies that some rotors can be performed gradually in small amounts (such as rotations), but that in the more general metrics there are also rotors that are like reflections and generate a discontinuous motion. Only the former are candidates for the proper orthogonal transformations that is hoped to be represented by proper rotors. There is an alternative method for constructing proper rotors other than multiplying unit vectors. That method is based on an exponential of a bi-vector angle $I\phi$ that directly computes the rotor from the bi-vector angle (without the need for the more basic reflection vectors). Using the traditional power series the exponential of a bi-vector angle $I\phi$ can be defined where $I^2 = -1$ as:

$$\begin{aligned}
e^{I\phi} &= 1 + \frac{1}{1!}(I\phi) + \frac{1}{2!}(I\phi)^2 + \cdots + \frac{1}{n!}(I\phi)^n + \cdots \\
&= \left[1 - \frac{1}{2!}\phi^2 + \frac{1}{4!}\phi^4 - \cdots\right] + I\left[\frac{1}{1!}\phi - \frac{1}{3!}\phi^3 + \frac{1}{5!}\phi^5 - \cdots\right] \\
&= \cos(\phi) + I\sin(\phi) \\
\Rightarrow R_{I\phi} &= \cos(\phi/2) - I\sin(\phi/2) = e^{-I\phi/2}
\end{aligned}$$

This exponential representation is isomorphic to the traditional exponential representation of unit complex numbers in 2D.

The same Taylor series expansion can be applied to any 2-blade A in a general metric space to create a "pure" rotor R_A (called pure because it is a product of two vectors i.e. a scalar plus a bi-vector with no other even grades of more than 2). The result of the series can be summarized according to the value of A^2 as follows:

$$
\begin{aligned}
exp(A) &= \cos(\alpha) + A\frac{\sin(\alpha)}{\alpha} \\
&= \cos(\alpha) + U\sin(\alpha),\ A^2 = -\alpha^2, \alpha \in \mathbb{R} \\
exp(A) &= 1 + A,\ A^2 = 0 \\
exp(A) &= \cosh(\alpha) + A\frac{\sinh(\alpha)}{\alpha} \\
&= \cosh(\alpha) + U\sinh(\alpha),\ A^2 = \alpha^2, \alpha \in \mathbb{R}
\end{aligned}
$$

In GA the geometric product of rotors is a rotor but the geometric product of exponentials of 2-blades are not guaranteed to be an exponential of a bi-vector because generally $e^A e^B \neq e^{A+B}$, $A, B \in B_2^n$. This means that not every rotor can be represented as an exponential of a bi-vector. It is proven however that only in spaces with Euclidean signatures$(n,0,0), (0,n,0)$ and Minkowski signatures $(n,1,0), (1,n,0)$ any exponential of a bi-vector is a rotor and any rotor can be expressed as an exponential of a bi-vector. Moreover only in these spaces a bi-vector B can be written as the sum of commuting (i.e. orthogonal) 2-blades $B = B_1 + B_2 + \cdots + B_k$, $B_i B_j = B_j B_i$. This allows the decomposition of the rotor that is the exponential of the bi-vector B into the geometric product of pure rotors:

$$
\begin{aligned}
R_B &= e^{B/2} \\
&= e^{(B_1+B_2+\cdots+B_k)/2} \\
&= e^{B_1/2} e^{B_2/2} \cdots e^{B_k/2} \\
&= R_{B_1} R_{B_2} \cdots R_{B_k}
\end{aligned}
$$

Thus any rotor can be decomposed to pure rotors in these spaces.

6.4 Logarithms of Rotors

Using a bi-vector B a rotor R can be created by the exponentiation $R = e^{B/2}$. The reverse process of finding the bi-vector B given tho rotor R (that is by definition a kind of logarithm $\log R$) is also possible put not generally easy. If such logarithm could be found, it can be used to create interpolations of rotations that finds smaller rotations with step N as: $R^{1/N} = \exp(\log(R)/N)$. The result $R^{1/N}$ is a rotor that performs the rotation from X to $RX\widetilde{R}$ as N smaller rotations. If the bi-vector B is a 2-blade, its exponential expansion involves standard trigonometric or hyperbolic functions, and its principal logarithm can be found using the inverse functions atan or atanh. However, the general rotor is the exponent of a bi-vector, not a 2-blade. Since a bi-vector does not usually square to a scalar, there are now no simple expansions of the exponential, and many mixed terms result. Often it's desirable to get back to the basic trigonometric or hyperbolic functions (to get geometrically significant parameters like bi-vector angles, translation vectors, and scalings). In this case a factorization of the total expression is needed. That would effectively split the bi-vector into mutually commuting 2-blades with sensible geometric meaning, and would make the logarithm extractible in closed form. Unfortunately this factorization is hard to do in general.

7 Frames of Basis Vectors

The second part of this article explains how to express all the GA operations in the first part using simple manipulations of real coordinates of multivectors. This explanation is an extension of the treatments in [1, 5, 6]. This topic is fundamental for any practical software implementation for GA computations.

7.1 Components of a GA Frame

A GA frame $\mathcal{F}(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}})$ is the container that is used to define all basic computations of a geometric algebra $\bigwedge^n$ in terms of the more basic scalar coordinates often used to write a program on a computer. A frame has can be completely defined using two components:

1. An ordered set of n basis vectors that determine the dimensionality of the frame's base vector space: $\boldsymbol{F}_1^n = \langle f_0, f_1, \cdots, f_{n-1} \rangle$.

2. A symmetric real bilinear form $\mathbf{B} : \boldsymbol{F}_1^n \times \boldsymbol{F}_1^n \to \mathbb{R}$, $\mathbf{B}(f_i, f_j) = \mathbf{B}(f_j, f_i) = f_i \cdot f_j$ to determine the inner product of basis vectors usually given by the bilinear form matrix $\mathbf{A}_{\mathcal{F}} = [f_i \cdot f_j]$; also called the Inner Product Matrix (IPM) of the frame.

From the above two components three other components can be automatically constructed (as will be explained shortly) to serve important purposes for GA computations within the frame:

1. The ordered set of 2^n basis blades of all grades $\boldsymbol{F}^n = \langle F_0, F_1, \cdots, F_{2^n-1} \rangle$. This set is automatically determined by the set of basis vectors $\boldsymbol{F}_1^n$ as will be explained latter. This component is completely independent of the metric represented by $\mathbf{A}_{\mathcal{F}}$.

2. The geometric product of basis blades $G_{\mathcal{F}} : \boldsymbol{F}^n \times \boldsymbol{F}^n \to \bigwedge^n$ that defines the geometric product of basis blades as a multivector expressed on the same basis blades $G_{\mathcal{F}}(F_i, F_j) = F_i F_j = \sum_{k=0}^{2^n-1} m_k F_k$, $m_k \in \mathbb{R}$. This operation is automatically determined by the set of basis vectors and the bilinear form as will be explained later.

3. If the bilinear form is not orthogonal (i.e. $\mathbf{A}_{\mathcal{F}}$ is not diagonal), an orthogonal base frame $\mathcal{E}(\boldsymbol{E}_1^n, \mathbf{A}_{\mathcal{E}})$ of the same dimension is needed and an orthogonal change-of-basis matrix $\mathbf{C}^{-1} = \mathbf{C}^T$ that can be used to express the basis vectors of $\mathcal{F}$ as linear combinations of the basis vectors of $\mathcal{E}$. This component is required for the computation of the geometric product of basis blades $G_{\mathcal{F}}$.

Using these five components any multivector X can be represented by a list of real coefficients $[x_i]$ where $X = \sum_{k=0}^{2^n-1} x_k F_k$, $x_k \in \mathbb{R}$ and the geometric product of two multivectors X, Y can be easily computed as:

$$XY = \sum_{r=0}^{2^n-1} \sum_{s=0}^{2^n-1} x_r y_s G_{\mathcal{F}}(F_r, F_s)$$

According to the form of the IPM $\mathbf{A}_{\mathcal{F}}$ a frame $\mathcal{F}$ can be of any of the following types listed from more special to more general:

Frame Type	IPM Form	Inner Product of Basis Vectors
Euclidean	Identity matrix	$f_i \cdot f_i = 1,\ f_i \cdot f_j = 0\, \forall i \neq j$
Orthonormal	Diagonal with ± 1 entries	$f_i \cdot f_i = \pm 1,\ f_i \cdot f_j = 0\, \forall i \neq j$
Orthogonal	Diagonal	$f_i \cdot f_i = d_i,\ f_i \cdot f_j = 0\, \forall i \neq j$
Non-orthogonal	Symmetric non-diagonal	$f_i \cdot f_j = f_j \cdot f_i = b_{ij}$

7.2 Representing Basis Blades using Basis Vectors

In order to define the basis blades $\boldsymbol{F}^n = \langle F_0, F_1, \cdots, F_{2^n-1} \rangle$ for a frame of any type, a canonical representation is defined based on the basis vectors $\boldsymbol{F}_1^n = \langle f_0, f_1, \cdots, f_{n-1} \rangle$. First the "subset selection" operator $\prod_{\oplus}(S, i)$ is introduced that applies an associative binary operator $\oplus$ with the identity element $I_{\oplus}$ to a subset of an ordered set of elements $S = \langle s_0, s_1, \cdots, s_{k-1} \rangle$ selected according to the integer index i as follows:

$$\prod_{\oplus}(S, i) = \begin{cases} I_{\oplus} & , i = 0 \\ s_m & , i = 2^m, m \in \{0, 1, \cdots, k-1\} \\ s_{i_1} \oplus s_{i_2} \oplus \cdots \oplus s_{i_r} & , \begin{array}{c} i = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_r}, \\ i_i < i_2 < \cdots < i_r \end{array} \end{cases}$$

This operator basically expresses the integer i as a binary number and selects an ordered subset S_i from S such that the binary representation of the index i_k of any element s_{i_k} in S_i has only one bit set to 1 at a position i_k that is present in the binary number corresponding to i . Using this operator, the basis blades are defined using basis vectors as:

$$\begin{aligned} F_k &= \prod_{\wedge}(\boldsymbol{F}_1^n, k) \\ &= \begin{cases} 1 & , k = 0 \\ f_m & , k = 2^m, m \in \{0, 1, \cdots, n-1\} \\ f_{i_1} \wedge f_{i_2} \wedge \cdots \wedge f_{i_r} & , \begin{array}{c} k = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_r}, \\ i_i < i_2 < \cdots < i_r \end{array} \end{cases} \end{aligned}$$

After defining the basis blades, any multivector $M = \sum_{i=0}^{2^n-1} m_i F_i$ can be represented using a column vector of real coefficients $[M]_{\boldsymbol{F}^n} = \begin{pmatrix} m_0 & m_1 & \cdots & m_{2^n-1} \end{pmatrix}^T$. This is called the additive representation of a multivector. The subset selection operator $\prod_{\wedge}(\boldsymbol{F}_1^n, k)$ creates a bijective correspondence between basis blades and n-bit binary numbers as in the following example on a 5D vector space with basis vectors $\boldsymbol{F}_1^5 = \langle f_0, f_1, f_2, f_3, f_4 \rangle$:

$$(13)_{10} = (01101)_2 \Longleftrightarrow \begin{array}{|c|c|c|c|c|} \hline 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline 0 & 1 & 1 & 0 & 1 \\ \hline f_4 & f_3 & f_2 & f_1 & f_0 \\ \hline & f_3 & f_2 & & f_0 \\ \hline \end{array} \Longleftrightarrow f_0 \wedge f_2 \wedge f_3 = F_{13}$$

Thus the index $k \in \{0, 1, \cdots, 2^n - 1\}$ of the basis blade F_k expressed as an n-bit binary number completely defines the structure of the basis blade F_k . This n-bit binary pattern will be called the ID of the basis blade ID_{F_k}. Thus any multivector can be stored in computer memory as an array (or perhaps for efficiency reasons as a dictionary or hash table) of 2^n scalars representing the coefficients of the basis blades with respect to the given frame. A pair of (ID, scalar) is called a term and represents a weighted basis blade thus the multivector is represented as a sum of terms with different IDs ranging from 0 to $2^n - 1$. In addition some useful metric-independent properties of a basis blade F_k can be computed directly from the ID ID_{F_k} like the grade g equal to the number of 1's in the bit pattern ID_{F_k}. Consequently the signs associated with the reversal $\widetilde{F_k}$, grade involution $\widehat{F_k}$, and Clifford conjugate $\overline{F_k}$ can be computed, which are all completely grade-dependent signs. Another important property is the order of the basis blade among its k-vector basis blades of the same grade g (called its k-vector index) $index(F_k)$. Thus a universal lookup table like the following one could be constructed to store all these metric independent information for any frame of dimension n. Using this table, the useful "inverse index" operator could be defined that retrieves a basis blade ID from its grade g and index i:

$$ID_g^n(i) = ID_{F_k} \Leftrightarrow grade(F_k) = g, \, and \, index(F_k) = i$$

The inverse index operator is useful when defining outermorphisms as will be described later. In addition, the subset selection operator could be used with the inverse index operator to describe the ordered set of basis k-vectors of the same grade $g \in \{0, 1, \cdots, n\}$ as:

$$\begin{aligned} \boldsymbol{F}_g^n &= \left\langle F_{k_0}, F_{k_1}, \cdots, F_{k_{r-1}} \right\rangle, \\ k_i &= ID_g^n(i) \; \forall i \in \{0, 1, \cdots, r\}, \; r = \begin{pmatrix} n \\ g \end{pmatrix} \end{aligned}$$

r	F_r	ID_{F_r}	$grade(F_r)$	$index(F_r)$
0	1	0000	0	0
1	f_0	0001	1	0
2	f_1	0010	1	1
3	$f_0 f_1$	0011	2	0
4	f_2	0100	1	2
5	$f_0 f_2$	0101	2	1
6	$f_1 f_2$	0110	2	2
7	$f_0 f_1 f_2$	0111	3	0
8	f_3	1000	1	3
9	$f_0 f_3$	1001	2	3
10	$f_1 f_3$	1010	2	4
11	$f_0 f_1 f_3$	1011	3	1
12	$f_2 f_3$	1100	2	5
13	$f_0 f_2 f_3$	1101	3	2
14	$f_1 f_2 f_3$	1110	3	3
15	$f_0 f_1 f_2 f_3$	1111	4	0

$\Longleftrightarrow$

r	F_r	ID_{F_r}	$grade(F_r)$	$index(F_r)$
0	1	0000	0	0
1	f_0	0001	1	0
2	f_1	0010	1	1
4	f_2	0100	1	2
8	f_3	1000	1	3
3	$f_0 f_1$	0011	2	0
5	$f_0 f_2$	0101	2	1
6	$f_1 f_2$	0110	2	2
9	$f_0 f_3$	1001	2	3
10	$f_1 f_3$	1010	2	4
12	$f_2 f_3$	1100	2	5
7	$f_0 f_1 f_2$	0111	3	0
11	$f_0 f_1 f_3$	1011	3	1
13	$f_0 f_2 f_3$	1101	3	2
14	$f_1 f_2 f_3$	1110	3	3
15	$f_0 f_1 f_2 f_3$	1111	4	0

7.3 Representing Euclidean Frames

A Euclidean frame $\mathcal{F}\left(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}}\right)$ has an diagonal IPM $\mathbf{A}_{\mathcal{F}}$ equal to the identity matrix with basis vectors satisfying:

$$\begin{aligned}
f_i \cdot f_i &= 1 \\
\Leftrightarrow f_i^2 &= 1, \\
f_i \cdot f_j &= 0 \\
\Leftrightarrow f_i f_j &= f_i \wedge f_j \\
&= -f_j \wedge f_i \\
&= -f_j f_i \ \forall i \neq j
\end{aligned}$$

For such frame it is straight forward to compute the geometric product of any two basis blades $G_{\mathcal{F}}(F_r, F_s)$ as it is always a signed basis blade in the form $G_{\mathcal{F}}(F_r, F_s) = F_r F_s = \pm F_k$ and what remains is only to find the value of k and the sign associated with the resulting basis blade F_k. The following is an example for the multiplication process performed algebraically, then its equivalent using the ID representation for the geometric product of basis blades $F_{13}F_{19} = (f_0 \wedge f_2 \wedge f_3)(f_0 \wedge f_1 \wedge f_4) = (f_0 f_2 f_3)(f_0 f_1 f_4)$ defined on a Euclidean frame of dimension 5. Algebraically the result is initially set to be $F_k = F_{13} = f_0 f_2 f_3$, then each basis vector in F_{19} is taken to perform the geometric product with F_k using the associativity and anti-symmetry properties as follows:

$$\begin{aligned}
F_k &\leftarrow F_k f_0 \\
&= (f_0 f_2 f_3) f_0 \\
&= (f_0 f_2)(f_3 f_0) \\
&= -(f_0 f_2)(f_0 f_3) \\
&= -(f_0)(f_2 f_0)(f_3) \\
&= (f_0)(f_0 f_2)(f_3) \\
&= (f_0 f_0)(f_2 f_3) \\
&= (1)(f_2 f_3) \\
&= f_2 f_3
\end{aligned}$$

$$\begin{aligned}
F_k &\leftarrow F_k f_1 \\
&= (f_2 f_3) f_1 \\
&= (f_2)(f_3 f_1) \\
&= -(f_2)(f_1 f_3) \\
&= -(f_2 f_1)(f_3) \\
&= (f_1 f_2)(f_3) \\
&= f_1 f_2 f_3
\end{aligned}$$

$$\begin{aligned}
F_k &\leftarrow F_k f_4 \\
&= (f_1 f_2 f_3) f_4 \\
&= f_1 f_2 f_3 f_4 \\
&= F_{30}
\end{aligned}$$

Using the corresponding IDs:

$$\begin{aligned}
ID_{F_{13}} \, XOR \, ID_{F_{19}} &= (01101)_2 \, XOR \, (10011)_2 \\
&= (11110)_2 \\
&= ID_{F_{30}}
\end{aligned}$$

This is not a coincidence because if the same basis vector f_i is present or absent in both basis blades it will always be absent in the final basis blade due to the property $f_i^2 = 1$ thus the ID of the final basis blade can be found directly by the bit-wise XOR operation of the IDs of the multiplied basis blades. The associated sign can be computed using the algorithm implied by the following table where $ID_1 \leftarrow ID_{F_{13}}, ID_2 \leftarrow ID_{F_{19}}$. The final basis blade will have the ID of $ID_{F_k} \leftarrow ID_1$ and the sign as indicated in the last row.:

	i	j	ID_2	ID_1	Condition	Action	Sign
1	0	-	10011	01101	Is bit i in $ID_2 = 1$? Yes	Enter decreasing loop $for\, j = 4\, to\, i+1$	$+1$
2	0	4	10011	01101	Is bit j in $ID_1 = 1$? No	Do nothing to sign, decrease j	$+1$
3	0	3	10011	01101	Is bit j in $ID_1 = 1$? Yes	Change sign, decrease j	-1
4	0	2	10011	01101	Is bit j in $ID_1 = 1$? Yes	Change sign, decrease j	$+1$
5	0	1	10011	01101	Is bit j in $ID_1 = 1$? No	Do nothing to sign, exit loop on j	$+1$
6	0	-	10011	01101	Is bit i in $ID_1 = 1$? Yes	Set bit i in $ID_1 \leftarrow 0$, increase i	$+1$
7	1	-	10011	01100	Is bit i in $ID_2 = 1$? Yes	Enter decreasing loop $for\, j = 4\, to\, i+1$	$+1$
8	1	4	10011	01100	Is bit j in $ID_1 = 1$? No	Do nothing to sign, decrease j	$+1$
9	1	3	10011	01100	Is bit j in $ID_1 = 1$? Yes	Change sign, decrease j	-1
10	1	2	10011	01100	Is bit j in $ID_1 = 1$? Yes	Change sign, exit loop on j	$+1$
11	1	-	10011	01100	Is bit i in $ID_1 = 1$? No	Set bit i in $ID_1 \leftarrow 1$, increase i	$+1$
12	2	-	10011	01110	Is bit i in $ID_2 = 1$? No	Do nothing, increase i	$+1$
13	3	-	10011	01110	Is bit i in $ID_2 = 1$? No	Do nothing, increase i	$+1$
14	4	-	10011	01110	Is bit i in $ID_2 = 1$? Yes	Do not enter decreasing loop since $4 < i+1$	$+1$
15	4	-	10011	01110	Is bit i in $ID_1 = 1$? No	Set bit i in $ID_1 \leftarrow 1$, exit loop on i	$+1$
16	-	-	-	11110	-	Return $Sign$ and ID_1	$+1$

The implied algorithm goes as follows: assuming $F_r = f_{r_1} f_{r_2} \cdots f_{r_p}$ and $F_s = f_{s_1} f_{s_2} \cdots f_{s_q}$ the geometric product $F_r F_s = \pm F_k$ is required, which is itself a signed basis blade. The following algorithm is performed:

1. Initialize a sign variable $Sign \leftarrow +1$ and ID variables $ID_1 \leftarrow ID_{F_r}, ID_2 \leftarrow ID_{F_s}$.
2. For increasing $i = 0\, to\, n-1$ do steps 3-6
3. If bit i in $ID_2 = 1$ do steps 4-6
4. For decreasing $j = n-1\, to\, i+1$ do step 5
5. If bit j in $ID_1 = 1$ Then Set $Sign \leftarrow -Sign$
6. If bit i in $ID_1 = 1$ Then Set it to 0 Else Set it to 1
7. Return result in $Sign$ and $ID_{F_k} \leftarrow ID_1$

Using this algorithm, a Euclidean geometric product table can be finally constructed with $2^n - 1$ rows and $2^n - 1$ columns where each cell at location (i, j) (i.e. row i and column j) contains the sign $Sign_E(F_i, F_j)$ and ID $ID_{F_k} = ID_{F_i}\, XOR\, ID_{F_j}$ of the basis blade resulting from the geometric product $F_i F_j$. Although this table is specific to Euclidean metric of dimension n it is fundamental in the computations of all bilinear product computations for any type of frame of the same dimension. An important property for $Sign_E(F_i, F_j)$ when applied to the same basis blade $F_k = f_{k_1} \wedge f_{k_2} \wedge \cdots \wedge f_{k_g}$ (i.e when $i = j = k$) can be deduced from the following:

$$
\begin{aligned}
F_k^2 &= F_k F_k \\
&= \left(f_{k_1} \wedge f_{k_2} \wedge \cdots \wedge f_{k_g}\right)\left(f_{k_1} \wedge f_{k_2} \wedge \cdots \wedge f_{k_g}\right) \\
&= (-1)^{g(g-1)/2}\left(f_{k_1} \wedge f_{k_2} \wedge \cdots \wedge f_{k_g}\right)\left(f_{k_g} \wedge f_{k_{g-1}} \wedge \cdots \wedge f_{k_1}\right) \\
&= (-1)^{g(g-1)/2}\left(f_{k_1} f_{k_2} \cdots f_{k_g}\right)\left(f_{k_g} f_{k_{g-1}} \cdots f_{k_1}\right) \\
&= (-1)^{g(g-1)/2} \\
\Rightarrow Sign_E(F_k, F_k) &= (-1)^{g(g-1)/2},\ g = grade\left(F_k\right)
\end{aligned}
$$

7.4 Representing Orthogonal Frames

An orthogonal frame $\mathcal{F}\left(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}}\right)$ has a diagonal IPM $\mathbf{A}_{\mathcal{F}}$ with basis vectors satisfying:

$$
\begin{aligned}
f_i \cdot f_i &= d_i \\
\Leftrightarrow f_i^2 &= d_i, \\
f_i \cdot f_j &= 0 \\
\Leftrightarrow f_i f_j &= f_i \wedge f_j \\
&= -f_j \wedge f_i \\
&= -f_j f_i\ \forall i \neq j
\end{aligned}
$$

Actually Euclidean frames and orthonormal frames are special cases of orthogonal frames. The only difference between a Euclidean frame and an orthogonal frame is that the square of a basis vector can be any real number d_i (including negative numbers and zero!). The same algorithm applied for a Euclidean frame can thus be used to deduce a geometric product for such frame with a single change to step 5 to become: "If bit i in $ID_1 = 1$ Then Set it to 0 and Set $Sign \leftarrow d_i * Sign$ Else Set it to 1". There is another alternative, however, in this case by using the geometric product for a Euclidean frame $\mathcal{E}\left(\boldsymbol{E}_1^n, \mathbf{A}_{\mathcal{E}}\right)$ with the same dimension having basis blades $\boldsymbol{E}^n = \langle E_0, E_1, \cdots, E_{2^n-1}\rangle$. If $E_r E_s = Sign_E(r,s) E_k$ then $F_r F_s = Sign_E(r,s)\lambda_k F_k$ where $\lambda_k = \prod\left(\langle d_0, d_1, \cdots, d_{n-1}\rangle, ID_{F_k}\right)$ is the multiplication of all d_i having a corresponding 1-bit in the bit pattern $ID_{F_k} = ID_{E_k}$. The scalar value λ_k is called the signature of the basis blade F_k and is defined as the squared norm of F_k as can be seen from:

$$
\begin{aligned}
F_k F_k &= F_k^2 \\
&= Sign_E(k,k)\lambda_k F_0 \\
&= (-1)^{g(g-1)/2}\lambda_k,\ g = grade\left(E_k\right) = grade\left(F_k\right) \\
\Rightarrow \lambda_k &= (-1)^{g(g-1)/2} F_k^2 \\
&= F_k \widetilde{F_k} \\
&= \left\|F_k\right\|^2
\end{aligned}
$$

This leads to a save in memory be simply storing n scalar values $\lambda_k = \left\|F_k\right\|^2, k \in \{0, 1, \cdots, 2^n - 1\}$ for the orthogonal frame, then the geometric product lookup table is used for the corresponding Euclidean frame $\mathcal{E}$ to compute $F_r F_s$ as:

$$
\begin{aligned}
F_r F_s &= \lambda_{r,s} F_k, \\
ID_{F_k} &= ID_{F_r}\, XOR\, ID_{F_s}, \\
\lambda_{r,s} &= Sign_{\mathcal{E}}(r,s)\left\|F_k\right\|^2
\end{aligned}
$$

Initially, say, 14 tables for Euclidean frames of dimension 2 to 16 could thus be created and use them to compute the geometric product of any orthogonal frame with any basis vectors signatures for the same dimensions.

7.5 Representing Derived Frames

Having a general frame $\mathcal{E}\left(\boldsymbol{E}_1^n, \mathbf{A}_{\mathcal{E}}\right)$ with basis vectors $\boldsymbol{E}_1^n = \langle e_1, e_2, \cdots, e_n\rangle$, inner product matrix $\mathbf{A}_{\mathcal{E}}$, and basis blades $\boldsymbol{E}^n = \langle E_0, E_1, \cdots, E_{2^n-1}\rangle$ an invertible change-of-basis matrix $\mathbf{C} = [c_{ij}]$ can be applied to any vector x

represented on the basis $\boldsymbol{E}_1^n$ by the column vector $[x]_{\boldsymbol{E}_1^n} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^T$ to obtain its representation $[x]_{\boldsymbol{F}_1^n} = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \end{pmatrix}^T$ on a new basis for the same space $\boldsymbol{F}_1^n = \langle f_1, f_2, \cdots, f_n \rangle$ (that shall be called the derived basis). Let $\mathbf{E} = \begin{pmatrix} e_1 & e_2 & \cdots & e_n \end{pmatrix}^T$ and $\mathbf{F} = \begin{pmatrix} f_1 & f_2 & \cdots & f_n \end{pmatrix}^T$ be column vectors containing symbols for the basis vectors of $\boldsymbol{E}_1^n$ and $\boldsymbol{F}_1^n$ such that $\mathbf{F} = \mathbf{CE} \Leftrightarrow f_i = \sum_{j=1}^n c_{ij} e_j \, \forall i \in \{1, 2, \cdots, n\}$. Then:

$$
\begin{aligned}
x &= [x]_{\boldsymbol{E}_1^n}^T \mathbf{E} \\
&= [x]_{\boldsymbol{F}_1^n}^T \mathbf{F} \\
&= [x]_{\boldsymbol{F}_1^n}^T \mathbf{CE} \\
\Rightarrow [x]_{\boldsymbol{E}_1^n}^T &= [x]_{\boldsymbol{F}_1^n}^T \mathbf{C} \\
\Rightarrow [x]_{\boldsymbol{E}_1^n} &= \mathbf{C}^T [x]_{\boldsymbol{F}_1^n} \\
\Rightarrow [x]_{\boldsymbol{F}_1^n} &= \mathbf{P} [x]_{\boldsymbol{E}_1^n}, \, \mathbf{P} = \left(\mathbf{C}^T\right)^{-1}
\end{aligned}
$$

Only in the case that $\mathbf{C}$ is orthogonal (i.e. $\mathbf{C}^{-1} = \mathbf{C}^T$) then $\mathbf{P} = \mathbf{C}$. The elements of the inner product matrix $A_{\mathcal{F}} = [f_i \cdot f_j]$ can be easily calculated as follows:

$$
\begin{aligned}
f_i \cdot f_j &= \left(\sum_{r=1}^n c_{ir} e_r \right) \cdot \left(\sum_{s=1}^n c_{js} e_s \right) \\
&= \sum_{s=1}^n \sum_{r=1}^n c_{ir} c_{js} (e_r \cdot e_s) \\
&= \sum_{s=1}^n \left(\sum_{r=1}^n c_{ir} (e_r \cdot e_s) \right) c_{sj}^T \\
\Rightarrow \mathbf{A}_{\mathcal{F}} &= \mathbf{C} \mathbf{A}_{\mathcal{E}} \mathbf{C}^T
\end{aligned}
$$

Using $\boldsymbol{F}_1^n$ and $\mathbf{A}_{\mathcal{F}}$ what is called a derived frame $\mathcal{F}(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}})$ can be constructed relative to the given base frame $\mathcal{E}(\boldsymbol{E}_1^n, \mathbf{A}_{\mathcal{E}})$ by means of the change-of-basis matrix $\mathbf{C}$. In the case when $\mathbf{A}_{\mathcal{F}}$ is diagonal, the geometric product of two multivectors represented in the derived frame can be computed using the method in the last subsection.

When $\mathbf{A}_{\mathcal{F}}$ is not diagonal but the base frame $\mathcal{E}$ is orthogonal and the transformation matrix is also orthogonal $\mathbf{C}^{-1} = \mathbf{C}^T \Leftrightarrow \mathbf{P} = \left(\mathbf{C}^T\right)^{-1} = \mathbf{C}$, the geometric product on the derived basis $\boldsymbol{F}^n$ can be computed by extending $\mathbf{C}^T$ and $\mathbf{C}$ as orthogonal outermorphisms $\overline{\mathbf{C}}^T$ and $\overline{\mathbf{C}}$ to be applied to multivectors for transforming back and forth between the base frame and the derived frame. Thus the geometric product of two multivectors X, Y can be computed as:

$$
XY = \overline{\mathbf{C}} \left[\overline{\mathbf{C}}^T [X] \, \overline{\mathbf{C}}^T [Y] \right], \, \overline{\mathbf{C}}^T = \overline{\mathbf{C}}^{-1}
$$

When $\mathbf{A}_F$ is not diagonal and one or both of the orthogonality conditions on the base frame $\mathcal{E}$ and change-of-basis matrix $\mathbf{C}$ do not hold, another method for computing the geometric product is needed, which is explained in the following subsection.

7.6 Representing Non-Orthogonal Frames

For a non-orthogonal frame $\mathcal{F}(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}})$ the geometric product of any two basis blades is not guaranteed to be a term (i.e. a weighted basis blade) but is generally a multivector (i.e. the sum of terms of different basis blades). Each cell in the geometric product table will then be a full multivector that may contain up to 2^n terms. This is a lot to store in memory for a single frame (2^{3n} terms many of which might be zeros). Another alternative is to use a diagonalization technique on the IPM $\mathbf{A}_{\mathcal{F}}$ to express the non-orthogonal frame as a derived frame for a base orthogonal frame $\mathcal{E}(\boldsymbol{E}_1^n, \mathbf{A}_{\mathcal{E}})$ with basis vectors $\boldsymbol{E}_1^n = \langle e_1, e_2, \cdots, e_n \rangle$. This is done by finding the IPM $\mathbf{A}_{\mathcal{E}}$ of the base orthogonal frame and the change-of-basis matrix $\mathbf{C} = [c_{ij}]$ that expresses the basis vectors of the derived non-orthogonal frame f_i as a linear combinations of the orthogonal basis vectors $f_i = \sum_{j=1}^n c_{ij} e_j \, \forall i \in \{1, 2, \cdots, n\}$ as stated in the previous sub-section.

Noting that the IPM $\mathbf{A}_{\mathcal{F}}$ is a symmetric square matrix, it is easy to find the real eigen values λ_i and LID eigen vectors V_i of $\mathbf{A}_{\mathcal{F}}$ that satisfy $A_{\mathcal{F}}V_i = \lambda_i V_i,\ i \in \{0, 1, \cdots, n\}$. Now if an orthonormalization technique is used on the vectors V_i (for example the Gram-Schmidt or Householder techniques) to obtain the orthonormal eigen vectors C_i and create the orthogonal square matrix $\mathbf{C} = \left[\begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array}\right]$ as a concatenation of the column vectors C_i, then the expression $\mathbf{A}_{\mathcal{E}} = \mathbf{C}^T\mathbf{A}_{\mathcal{F}}\mathbf{C}$ is actually a diagonal matrix containing the eigen values on its diagonal thus it can be considered the IPM of a base orthogonal frame to this derived frame as described in the previous section. Now it is simple to extend $\mathbf{C}^T$ and $\mathbf{C}$ as orthogonal outermorphisms $\overline{\mathbf{C}}^T$ and $\overline{\mathbf{C}}$ to be applied to multivectors for transforming back and forth between the base orthogonal frame $\mathcal{E}$ and the non-orthogonal derived frame $\mathcal{F}$. Thus the geometric product of two multivectors X, Y can be computed as:

$$XY = \overline{\mathbf{C}}\left[\overline{\mathbf{C}}^T[X]\,\overline{\mathbf{C}}^T[Y]\right], \overline{\mathbf{C}}^T = \overline{\mathbf{C}}^{-1}$$

This means that for a non-orthogonal frame it is necessary to construct and store the outermorphisms $\overline{\mathbf{C}}^T$ and $\overline{\mathbf{C}}$ to use them for performing the geometric product and other bilinear products as will be explained later..

7.7 Representing Reciprocal Frames

Having a general base frame $\mathcal{E}\left(\boldsymbol{E}_1^n, \mathbf{A}_{\mathcal{E}}\right)$with basis vectors $\boldsymbol{E}_1^n = \langle e_1, e_2, \cdots, e_n\rangle$, inner product matrix $\mathbf{A}_{\mathcal{E}}$, and basis blades $\boldsymbol{E}^n = \langle E_0, E_1, \cdots, E_{2^n-1}\rangle$, it is possible to create a special type of derived frame called the reciprocal frame $\mathcal{F}\left(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}}\right)$ having basis vectors $\boldsymbol{F}_1^n = \langle f_1, f_2, \cdots, f_n\rangle$ using the relation:

$$\begin{aligned}
f_i &= (-1)^{i-1}\left(e_1 \wedge e_2 \wedge \cdots \wedge e_{i-1} \wedge e_{i+1} \wedge \cdots \wedge e_n\right)\rfloor I^{-1} \qquad (10)\\
I &= e_1 \wedge e_2 \wedge \cdots \wedge e_n,\\
I^{-1} &= \frac{(-1)^{n(n-1)/2}}{I * \widetilde{I}} I\\
\Rightarrow f_i \cdot e_i &= 1\ \forall i \in \{1, 2, \cdots, n\},\\
f_i \cdot e_j &= 0\ \forall i, j \in \{1, 2, \cdots, n\}, i \neq j
\end{aligned}$$

If the base frame $\mathcal{E}$ is orthogonal the derived reciprocal frame $\mathcal{F}$ is also orthogonal and the above relation reduces to the simple form:

$$\begin{aligned}
f_i &= \frac{1}{e_i \cdot e_i} e_i\ \forall i \in \{0, 1, \cdots, n-1\}\\
\Leftrightarrow \mathbf{A}_{\mathcal{F}} &= \mathbf{A}_{\mathcal{E}}^{-1}
\end{aligned}$$

For a non-orthogonal base frame $\mathcal{E}$ the reciprocal frame $\mathcal{F}$ is also non-orthogonal, thus equation (10) can be used to compute each f_i as a linear combination of $\boldsymbol{E}_1^n$. Then the change of basis matrix $\mathbf{C}$ can be easily derived, and the process is continued as usual with the non-orthogonal derived frame $\mathcal{F}$ by finding another orthogonal base frame and orthogonal outermorphism as described in the previous subsection.

7.8 Performing bilinear Products

This subsection describes how to implement the following bilinear products of geometric algebra on multivectors X, Y within a frame of any type:

- The outer product $X \wedge Y$
- The scalar product $X * Y$
- The left contraction product $X\rfloor Y$
- The right contraction product $X\lfloor Y$
- The fat-dot product $X \bullet Y$
- The Hestenes dot product $X \bullet_H Y$

- The commutator product $X \otimes Y$
- The anti-commutator product $X \odot Y$

Starting with orthogonal frames, any bilinear product $\star$ of two multivectors $X, Y \in \bigwedge^n$ performed on their representations $[X]_{\boldsymbol{F}^n} = [x_i]_{n\times 1}, [Y]_{\boldsymbol{F}^n} = [y_i]_{n\times 1}$ in an orthogonal frame $\mathcal{F}(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}})$ with basis blades $\boldsymbol{F}^n = \langle F_0, F_1, \cdots, F_{2^n-1}\rangle$ can be implemented as:

$$X \star Y = \sum_{r=0}^{2^n-1}\sum_{s=0}^{2^n-1} x_r y_s (F_r \star F_s)$$

Now the goal is to find the value of $F_r \star F_s$ for all $r, s \in \{0, 1, \cdots, 2^n - 1\}$. Due to the properties of the geometric product on orthogonal frames and the definitions of the bilinear products, the bilinear product of any two basis blades $F_r \star F_s$ is either a zero or a single term $\lambda_{r,s}^{\star} F_k$ but never more than a single term. Actually when the value of $F_r \star F_s = \lambda_{r,s}^{\star} F_k \neq 0$ the term is equal to the geometric product of the two basis blades $\lambda_{r,s}^{\star} F_k = G_{\mathcal{F}}(F_r, F_s) = F_r F_s$. The task is now to answer the following for each $F_r \star F_s$:

1. Based on the general definition of the bilinear product alone (i.e. with no regard for the actual metric), when is $F_r \star F_s$ guaranteed to equal zero?
2. If $F_r \star F_s$ is not guaranteed to equal zero (by the general definition of the bilinear product alone), what is the value of the geometric product $G_{\mathcal{F}}(F_r, F_s) = F_r F_s$?

The second question is already answered in the previous sections for orthogonal frames. What remains is the answer to the first question.

Starting with the outer product that is metric independent, a Euclidean metric with the same dimension can be assumed with no loss of generality. Noting that in a Euclidean frame the outer product of two basis blades with no basis vectors in common is completely equivalent to the geometric product of the two basis blades. Thus the outer product of two basis blades is itself a signed basis blade with a grade equal to the sum of their grades $F_r \wedge F_s = \lambda_{r,s}^{\wedge} F_k$, $\lambda_{r,s}^{\wedge} = \pm 1$, $grade(F_k) = grade(F_r) + grade(F_s)$. In addition the outer product of two basis blades with any common basis vector is always zero: $ID_{F_r}\, AND\, ID_{F_s} \neq 0 \Rightarrow F_r \wedge F_s = 0$. Other than that the process of performing the outer product in any metric (orthogonal or not) is identical to the process of performing the geometric product on a Euclidean frame of the same dimension. Thus the outer product of basis blades $F_r \wedge F_s$ can be formulated using the geometric product of the two corresponding basis blades of the Euclidean frame with same dimension $\mathcal{E}(\boldsymbol{E}_1^n, \mathbf{I}_n)$ as follows:

$$F_r \wedge F_s = \begin{cases} 0 & ID_{F_r}\, AND\, ID_{F_s} \neq 0 \\ G_{\mathcal{E}}(E_r, E_s) & otherwise \end{cases}$$

Next comes the scalar product that has the property $F_r * F_s = \langle F_r F_s\rangle_0 = \langle \lambda_{r,s} F_k\rangle_0 = \lambda_{r,s}\langle F_k\rangle_0$ where $ID_{F_k} = ID_{F_r}\, XOR\, ID_{F_s}$ and $\lambda_{r,s} = Sign_{\mathcal{E}}(r,s)\,\|F_k\|^2$. Since the only basis blade with grade zero is F_0 (i.e. $F_r * F_s = \lambda_{r,s} \Leftrightarrow ID_{F_r}\, XOR\, ID_{F_s} = ID_{F_0} = 0$), the scalar product of two basis blades can be computed using the following relation:

$$F_r * F_s = \begin{cases} 0 & ID_{F_r}\, XOR\, ID_{F_s} \neq 0 \\ F_r F_s & otherwise \end{cases}$$

For the left contraction $F_r \rfloor F_s = \langle F_r F_s\rangle_{b-a} = \langle \lambda_{r,s} F_k\rangle_{b-a} = \lambda_{r,s}\langle F_k\rangle_{b-a}$, $a = grade(F_r)$, $b = grade(F_s)$. Using an argument similar to the scalar product if there is a basis blade in F_r that is not present in F_s the number of ones in the pattern $ID_{F_k} = ID_{F_r}\, XOR\, ID_{F_s}$ will be more than $grade(F_s) - grade(F_r)$ leading to a zero left contraction. Thus the left contraction product of two basis blades can be computed using the following relation:

$$F_r \rfloor F_s = \begin{cases} 0 & ID_{F_r}\, AND\, NOT\, ID_{F_s} \neq 0 \\ F_r F_s & otherwise \end{cases}$$

The same for the right contraction $F_r \lfloor F_s = \langle F_r F_s\rangle_{a-b} = \langle \lambda_{r,s} F_k\rangle_{a-b} = \lambda_{r,s}\langle F_k\rangle_{a-b}$, $a = grade(F_r)$, $b = grade(F_s)$ that can be computed using the relation:

$$F_r \lfloor F_s = \begin{cases} 0 & ID_{F_s}\, AND\, NOT\, ID_{F_r} \neq 0 \\ F_r F_s & otherwise \end{cases}$$

The fat-dot product of two basis blades has the property $F_r \bullet F_s = F_r \rfloor F_s + F_r \lfloor F_s - F_r * F_s$ which leads to one of three cases:

1. $a = b \Rightarrow F_r \bullet F_s = F_r * F_s + F_r * F_s - F_r * F_s = F_r * F_s$
2. $a < b \Rightarrow F_r \bullet F_s = F_r \rfloor F_s$
3. $a > b \Rightarrow F_r \bullet F_s = F_r \lfloor F_s$

This leads to the computation relation:

$$F_r \bullet F_s = \begin{cases} 0 & a = b,\, (ID_{F_r}\, XOR\, ID_{F_s}) \neq 0 \\ 0 & a < b,\, (ID_{F_r}\, AND\, NOT\, ID_{F_s}) \neq 0 \\ 0 & a > b,\, (ID_{F_s}\, AND\, NOT\, ID_{F_r}) \neq 0 \\ F_r F_s & otherwise \end{cases}$$

The Hestenes inner product $F_r \bullet_H F_s$ is equal to the fat-dot product unless one of the multivectors is a scalar in which case the result is zero. This leads to the computation relation:

$$F_r \bullet_H F_s = \begin{cases} 0 & ab > 0,\, a = b,\, (ID_{F_r}\, XOR\, ID_{F_s}) \neq 0 \\ 0 & ab > 0,\, a < b,\, (ID_{F_r}\, AND\, NOT\, ID_{F_s}) \neq 0 \\ 0 & ab > 0,\, a > b,\, (ID_{F_s}\, AND\, NOT\, ID_{F_r}) \neq 0 \\ 0 & ab = 0 \\ F_r F_s & otherwise \end{cases}$$

The commutator product of basis blades satisfies $F_r \otimes F_s = \frac{1}{2}(F_r F_s - F_s F_r) = \frac{1}{2}(\lambda_{r,s} - \lambda_{s,r}) F_k$ thus the computation relation can be deduced using the sign of the geometric product of the two corresponding basis blades of the Euclidean frame with same dimension as follows::

$$F_r \otimes F_s = \begin{cases} 0 & Sign_{\mathcal{E}}(E_r, E_s) = Sign_{\mathcal{E}}(E_s, E_r) \\ F_r F_s & otherwise \end{cases}$$

The anti-commutator product of basis blades similarly satisfies $F_r \odot F_s = \frac{1}{2}(F_r F_s + F_s F_r) = \frac{1}{2}(\lambda_{r,s} + \lambda_{s,r}) F_k$ thus:

$$F_r \odot F_s = \begin{cases} 0 & Sign_{\mathcal{E}}(E_r, E_s) = -Sign_{\mathcal{E}}(E_s, E_r) \\ F_r F_s & otherwise \end{cases}$$

For a non-orthogonal frame $\mathcal{F}$ a bilinear product of two basis blades is not guaranteed to produce a weighted basis blade (except for the outer product) thus all the above computational relations become invalid. The alternative computation method comes from the fact that for any bilinear product $\star$ if an orthogonal outermorphism $\overline{\mathbf{T}}^T = \overline{\mathbf{T}}^{-1}$ is used, the transform of the product of two multivectors X, Y equals the product of the transformed multivectors:

$$\begin{aligned} \overline{\mathbf{T}}[X \star Y] &= \overline{\mathbf{T}}[X] \star \overline{\mathbf{T}}[Y],\, \overline{\mathbf{T}}^T = \overline{\mathbf{T}}^{-1} \\ \Leftrightarrow X \star Y &= \overline{\mathbf{T}}^T \left[\overline{\mathbf{T}}[X] \star \overline{\mathbf{T}}[Y]\right] \end{aligned}$$

Thus it is possible to use the outer morphism $\overline{\mathbf{T}} = \overline{\mathbf{C}}^T$ between the non-orthogonal frame $\mathcal{F}$ and its base orthogonal frame $\mathcal{E}$ to perform the bilinear product just as with the geometric product before:

$$X \star Y = \overline{\mathbf{C}} \left[\overline{\mathbf{C}}^T [X] \star \overline{\mathbf{C}}^T [Y]\right]$$

8 Linear Transforms of Multivectors

8.1 Components of a Linear Transform on Multivectors

The fact that a geometric algebra is itself a linear space on 2^n basis blades enables the use of matrices as general linear transformations on multivectors where the input multivector is expressed as a column vector containing the 2^n coefficients of the basis blades of the selected domain frame. A linear transform $\mathbf{T} : \bigwedge^n \to \bigwedge^m$ on multivectors can thus be defined using the following three components:

1. The n-dimensional GA frame of the domain of the transform $\mathcal{E}\left(\boldsymbol{E}_1^n, \mathbf{A}_{\mathcal{E}}\right)$.
2. The m-dimensional GA frame of the co-domain of the transform $\mathcal{F}\left(\boldsymbol{F}_1^m, \mathbf{A}_{\mathcal{F}}\right)$.
3. The $2^m \times 2^n$ real transformation matrix $\mathbf{M_T}$ that can be used to apply the transform using matrix multiplication$Y = \mathbf{T}\left[X\right] \Leftrightarrow \left[Y\right]_{F^m} = \mathbf{M_T}\left[X\right]_{E^n}$.

All linear operations in geometric algebra can be expressed as general linear transformations on multivector. The real problem with this representation is the large storage space required for the of $\mathbf{M_T}$. Fortunately many such operations result in a very sparse $\mathbf{M_T}$ thus may be implemented using sparse arrays of various forms. In addition many important linear transforms are actually outermorphisms (i.e. they preserve the grade of transformed blades). The linear transform matrix of an outermorphism can be represented as a set of smaller block matrices as will be described in the following section. The use of a matrix to represent the linear transform is an easy method to implement many computations on linear transformations:

- The identity transform can be represented by the identity matrix $\mathbf{I}$.
- The adjoint transform $\mathbf{T}^T$ can be represented by matrix transpose $\left(\mathbf{M_T}\right)^T$.
- The inverse transform $\mathbf{T}^{-1}$ can be represented by matrix inverse $\left(\mathbf{M_T}\right)^{-1}$.
- The addition of two transforms $\mathbf{T}_1 + \mathbf{T}_2$ can be represented by addition of matrices $\mathbf{M}_{\mathbf{T}_1} + \mathbf{M}_{\mathbf{T}_2}$.
- The scaling of a linear transform $\alpha\mathbf{T}$ can be represented by matrix multiplication with scalar $\alpha\mathbf{M}_T$.
- The composition of two transforms $\mathbf{T}_1 \circ \mathbf{T}_2$ can be represented by matrix multiplication $\mathbf{M}_{\mathbf{T}_1}\mathbf{M}_{\mathbf{T}_2}$.
- and so forth...

8.2 Representing Outermorphisms

An outer morphism $\overline{\mathbf{f}} : \bigwedge^n \to \bigwedge^m$ is an extension of a linear transformation of vectors (represented by a matrix $\mathbf{M_f} = \begin{bmatrix} m_0 & m_1 & \cdots & m_{n-1} \end{bmatrix}_{m \times n}$ containing the column vectors $m_1, m_2, \cdots, m_n$) to transform whole blades and is naturally linear and grade preserving:

$$
\begin{aligned}
\overline{\mathbf{f}} : \overset{n}{\bigwedge} &\to \overset{m}{\bigwedge} \\
\overline{\mathbf{f}}[\alpha] &= \alpha,\ \forall\alpha \in \overset{n}{\underset{0}{\bigwedge}} \\
\overline{\mathbf{f}}[A+B] &= \overline{\mathbf{f}}[A] + \overline{\mathbf{f}}[B],\ \forall A, B \in \overset{n}{\bigwedge} \\
\overline{\mathbf{f}}[X \wedge Y] &= \overline{\mathbf{f}}[X] \wedge \overline{\mathbf{f}}[Y],\ \forall X, Y \in B^n
\end{aligned}
$$

This means that when applying an outermorphism to a multivector A the transformation can be decomposed as follows:

$$
\begin{aligned}
\overline{\mathbf{f}}[A] &= \overline{\mathbf{f}}\left[\langle A\rangle_0 + \langle A\rangle_1 + \cdots + \langle A\rangle_n\right] \\
&= \overline{\mathbf{f}}\left[\langle A\rangle_0\right] + \overline{\mathbf{f}}\left[\langle A\rangle_1\right] + \cdots + \overline{\mathbf{f}}\left[\langle A\rangle_n\right] \\
&= \mathbf{f}_0\left[\langle A\rangle_0\right] + \mathbf{f}_1\left[\langle A\rangle_1\right] + \cdots + \mathbf{f}_n\left[\langle A\rangle_n\right]
\end{aligned}
$$

Each $\mathbf{f}_i : \bigwedge_i^n \to \bigwedge_i^m$ is a linear transform on the subspace of k-vectors of grade i having an associated transformation matrix $\mathbf{M}_i = \begin{bmatrix} m_0^i & m_1^i & \cdots & m_{r_i}^i \end{bmatrix}$, $r_i = \begin{pmatrix} n \\ i \end{pmatrix}$ that can be directly constructed from $\mathbf{M_f}$. Let $\boldsymbol{E}^n = \langle E_0, E_1, \cdots, E_{2^n-1} \rangle$ be the basis blades of the domain of $\overline{\mathbf{f}}$, $\mathbf{F} = \begin{pmatrix} f_1 & f_2 & \cdots & f_n \end{pmatrix}^T$ be the column vector containing the basis vector symbols of the co-domain, and let $v_i = m_i^T \mathbf{F} \Leftrightarrow [v_i]_{\boldsymbol{F}_1^m} = m_i$ and define the ordered set of vectors $V = \langle v_0, v_1, \cdots, v_{n-1} \rangle$. Then the columns of the matrices $\mathbf{M}_i$ can be defined as follows:

$$\begin{aligned} m_0^0 &= [1]_{1\times 1} \\ m_j^1 &= m_j \\ m_j^i &= \left[\prod_{\wedge} (V, ID_i^n(j)) \right]_{\boldsymbol{F}_i^m} \end{aligned}$$

The last relation deserves some explanation. First noting that the matrix $\mathbf{M}_0$ is used to transform the scalar part of a multivector. Because of the definition of the outermorphism: $\mathbf{f}[\langle A \rangle_0] = \langle A \rangle_0$ this matrix must be a 1×1 matrix containing a single scalar equal to 1. Next the matrix $\mathbf{M}_1$ is used to transform vectors which is the same use of the matrix $\mathbf{M_f}$. Continuing with higher grades, column j of the k-vector transformation matrix $\mathbf{M}_i$, $i > 1$ can be obtained by the following steps:

1. Find the ID $k = ID_i^n(j)$ of the basis blade E_k whose grade is i and index is j: $ID_i^n(j) = ID_{E_k}$, $grade(E_k) = i$, $index(E_k) = j$.

2. Use the ID you found to apply the outer product to a subset of vectors in V: $X_k = v_{k_1} \wedge v_{k_2} \wedge \cdots \wedge v_{k_r}$ where $r = \begin{pmatrix} n \\ i \end{pmatrix}$, $k = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_r}, k_1 < k_2 < \cdots < k_r$

3. Find the representation of X_k with respect to the grade i basis k-vectors $\boldsymbol{F}_i^m$: $[X_k]_{\boldsymbol{F}_i^m}$.

If the outer product is applied (as described shortly) using the representation matrices m_i directly, finding the k-vectors v_i or their representation with respect to the k-vector basis $\boldsymbol{F}_i^m$ is not needed. In addition, m_j^i, $i > 1$ can be computed using the outer product of the two blades represented by m_r^{i-1} and m_s^1 where $ID_i^n(j) = ID_{i-1}^n(r)\, OR\, ID_{i-1}^n(s)$, $ID_{i-1}^n(r)\, AND\, ID_1^n(s) = 0$ (i.e. the bit pattern associated with the basis blade represented by m_j^i is decomposed to two lower-grade bit patterns with no common basis vectors). Thus the columns can be constructed gradually starting from the columns of the original transformation matrix $\mathbf{M_f}$. In addition the determinant of the base linear operator on vectors $det(\mathbf{f}) = |\mathbf{M_f}|$ is defined as $det(\mathbf{f}) = \frac{\mathbf{f}[I]}{I}$ where $I = f_0 \wedge f_1 \wedge \cdots \wedge f_{n-1}$ is the pseudo-scalar of the co-domain frame $\mathcal{F}(\boldsymbol{F}_1^n, \mathbf{A}_{\mathcal{F}})$ for outermorphisms on spaces with the same domain and co-domain. This is simply the value of the single element in the 1×1 matrix m_n^n.

Using this method of constructing outermorphisms, many operations $\overline{\mathbf{g}} = P\left\{\overline{\mathbf{f}}\right\}$ on outermorphisms that produce outermorphisms can be done using one of two methods:

1. The operation is performed on the base matrix $\mathbf{M_f}$ (i.e. the base linear transform on vectors) to obtain a new matrix $\mathbf{S_g} = P\{\mathbf{M_f}\}$ that represents a linear transform of vectors $\mathbf{g}$ then extend $\mathbf{g}$ as an outermorphism $\overline{\mathbf{g}}$.

2. The operation is performed on every k-vector transformation matrix $\mathbf{M}_i$ to obtain a new set of matrices $\mathbf{S}_i = P\{\mathbf{M}_i\}$ that construct the new outermorphism.

The following operations can be implemented using any of the two methods:

- The adjoint of an outermorphism.
- The inverse on an outermorphism.
- The composition of two outermorphisms.

The following operations can only be implemented by the first method:

- The scaling of an outermorphism (equivalent to multiplication of the vector transform matrix by a scalar).
- The addition of two outermorphisms.

The reason for the last two exceptions that if an outermorphism is treated as an ordinary linear transform on multivectors the traditional addition of two matrices $\mathbf{S} = [s_{ij}], \mathbf{T} = [t_{ij}]$ representing the linear transform matrices of the two outermorphisms will never itself be an outermorphism because $s_{11} + t_{11} = 1 + 1 = 2$ thus leading to a linear transform on multivectors that changes the scale of scalars which is definitely not an outermorphism. The same argument holds when multiplying an outermorphism by a real scalar α, , in which case the k-vector matrix $\mathbf{M}_i$ must be multiplied by α^i to obtain the correct result; rather than multiplication by α. This means that there is a structural distinction between performing operations on a general linear transform of multivectors and performing the same operations on outermorphisms to obtain other outermorphisms due to the extra algebraic structure provided by the concept of an outermorphism as an extension for linear transformations on vectors so care must be taken when applying such operations about what is actually meant.

Finally, all GA operations defined using bilinear GA products and outermorphisms can be automatically converted to simpler, but much involved, computations on frames; including linear projections, meet and join of blades, versor products, and reflections and rotations of blades.

References

[1] L. Dorst, D. Fontijne, and S. Mann, *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*, ser. The Morgan Kaufmann Series in Computer Graphics. Elsevier Science, 2010. [Online]. Available: https://books.google.com.eg/books?id=Sg_uR57ZktoC

[2] K. Kanatani, *Understanding Geometric Algebra: Hamilton, Grassmann, and Clifford for Computer Vision and Graphics*. CRC Press, 2015. [Online]. Available: https://books.google.com.eg/books?id=LZS9BwAAQBAJ

[3] S. Axler, *Linear Algebra Done Right*, ser. Undergraduate Texts in Mathematics. Springer International Publishing, 2014. [Online]. Available: https://books.google.com.eg/books?id=5qYxBQAAQBAJ

[4] D. Poole, *Linear Algebra: A Modern Introduction*. Cengage Learning, 2014. [Online]. Available: https://books.google.com.eg/books?id=V-UbCgAAQBAJ

[5] M. D. Zaharia and L. Dorst, "The interface specification and implementation internals of a program module for geometric algebra," JOURNAL OF LOGIC AND ALGEBRAIC PROGRAMMING, Tech. Rep., 2002.

[6] D. Fontijne, *Efficient Implementation of Geometric Algebra*. Universiteit van Amsterdam [Host], 2007. [Online]. Available: https://books.google.com.eg/books?id=oAzdMgEACAAJ