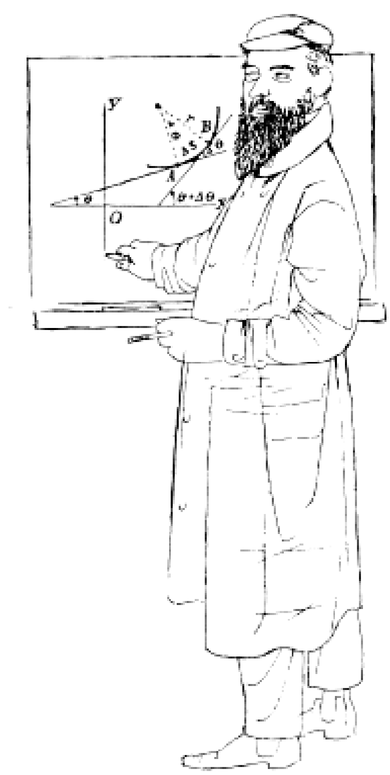
Geometric (Clifford) algebra: a practical tool for efficient geometrical representation

Leo Dorst (`leo@wins.uva.nl`)
Intelligent Autonomous Systems,
Dept. of Computer Science,
University of Amsterdam, The Netherlands

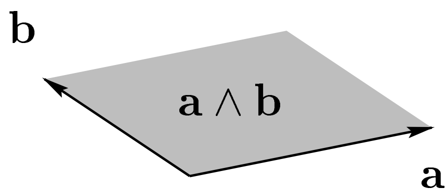
May 24, 1999

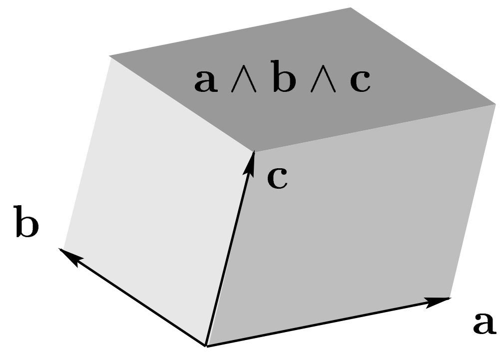


William Kingdon Clifford (1845-1879)

1 What if...

... we could span areas, volumes, etcetera using vectors so that the notation has a *computational* meaning?



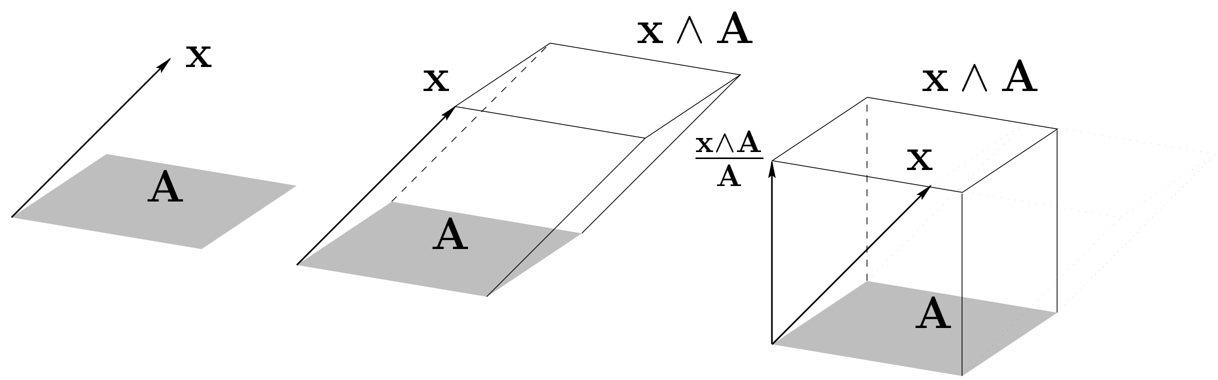


What computation laws should these obey? How should $(\mathbf{a} \wedge \mathbf{b}) \wedge \mathbf{c}$ and $\mathbf{a} \wedge (\mathbf{b} \wedge \mathbf{c})$ and $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$ compare? Could $\mathbf{a} \wedge \mathbf{b}$ and $\mathbf{b} \wedge \mathbf{a}$ represent directed areas of opposite orientation?

What could we do if we had such a 'product' for vectors? Would it lead to new insights in geometry, giving new algorithms?

2 Then we might be able to...

... determine the component $\mathbf{x}_\perp$ of a vector $\mathbf{x}$ perpendicular to a plane $\mathbf{A}$, by just ...

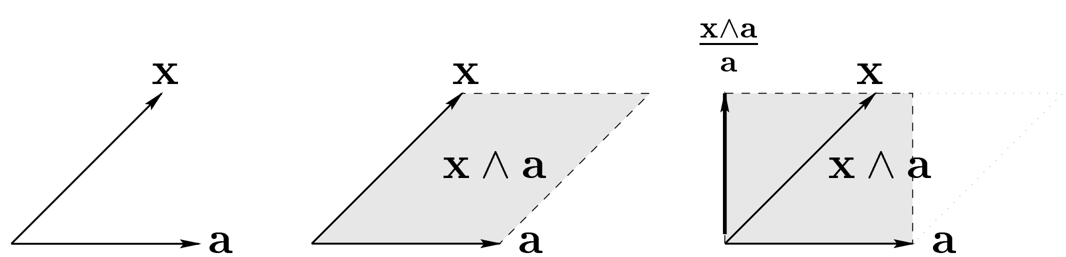


... constructing the volume $\mathbf{x} \wedge \mathbf{A}$ spanned by $\mathbf{x}$ and $\mathbf{A}$, and (after reshaping it, preserving the content), dividing that by $\mathbf{A}$, so that

$$\mathbf{x}_\perp = \frac{\mathbf{x} \wedge \mathbf{A}}{\mathbf{A}}$$

To do so, *we need to be able to construct re-shape-able volumes, and to divide by areas segments.*

A similar construction would also work for perpendicularity to vectors:

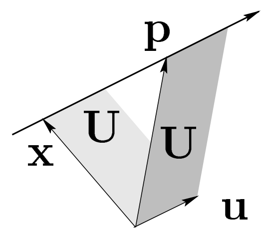


and maybe even in n-dimensional space?

3 Also, we might be able to...

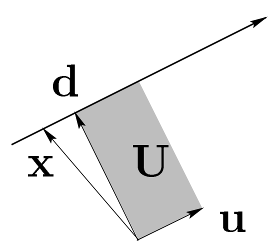
... characterize the points on a line by the fact that they all span the same re-shape-able area $\mathbf{U}$:

$$\mathbf{x} \wedge \mathbf{u} = \mathbf{U}.$$



Thus $\mathbf{u}$ and $\mathbf{U}$ (a direction vector and an area) determine a line. The area element also defines the plane of the line, so this works for lines in n-D; and a similar construction can be used for planes, hyperplanes, etc.

The *perpendicular support vector* $\mathbf{d}$ of the line is then simply:



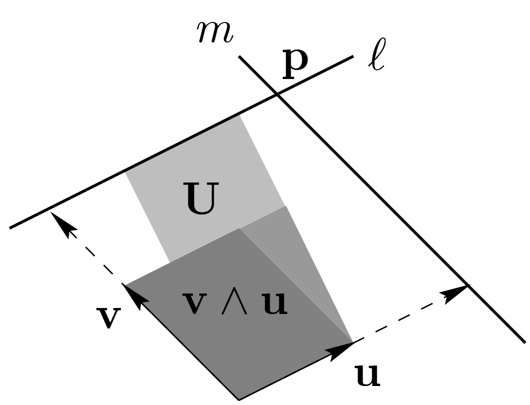
$$\mathbf{d} = \frac{\mathbf{U}}{\mathbf{u}}.$$

4 And, who knows ...?

Another anti-symmetrical geometrical operation between two points (vectors?) x and y is the directed line from x to y. Could we design a 'join'-operator $\dot{\wedge}$ such that $x \mathbin{\dot{\wedge}} y$ is a computable representation of the line through x and y, and $x \mathbin{\dot{\wedge}} y \mathbin{\dot{\wedge}} z$ would represent the plane through x, y and z, etcetera? Feels similar to $\wedge$ (anti-symmetry) – *is there a common algebra, and hence implementation, underneath?*

We also want a 'meet'-operator so that $(\mathbf{a}_1 \mathbin{\dot{\wedge}} \mathbf{a}_2) \vee (\mathbf{b}_1 \mathbin{\dot{\wedge}} \mathbf{b}_2)$ is the intersection of the two lines $\mathbf{a}_1 \mathbin{\dot{\wedge}} \mathbf{a}_2$ and $\mathbf{b}_1 \mathbin{\dot{\wedge}} \mathbf{b}_2$.

This should be an immediately computable representation, combining well with the previous ideas. For instance, in 2-D, the sketch:



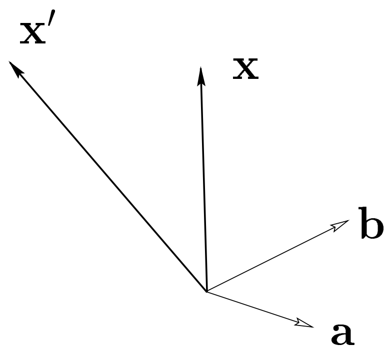
should give the formula/algorithm:

$$\mathbf{p} = \ell \vee m = \frac{\mathbf{U}}{\mathbf{v} \wedge \mathbf{u}}\,\mathbf{v} + \frac{\mathbf{V}}{\mathbf{u} \wedge \mathbf{v}}\,\mathbf{u}.$$

which should also follow from the computation with the $\vee$ operator.

5 And wouldn't it be nice if...

... we could make a new vector $\mathbf{x}'$ that is to a known vector $\mathbf{x}$ as a known vector $\mathbf{b}$ to a known vector $\mathbf{a}$:



and if we could write this in a formula as:

$$\frac{\mathbf{x}'}{\mathbf{x}} = \frac{\mathbf{b}}{\mathbf{a}}$$

which could be solved to give:

$$\mathbf{x}' = \frac{\mathbf{b}}{\mathbf{a}}\mathbf{x}.$$

This would implement *rotation/scaling* for arbitrary vectors, and characterize those operations by the ratio of two known vectors $\mathbf{a}$ and $\mathbf{b}$ (which probably depends only on their plane, angle and relative magnitude).

6 Welcome to the world of Geometric Algebra!

These constructions are all examples of things you can do in *geometric algebra.* (This is actually something mathematicians know as *Clifford algebra*, but extended with suitably defined 'geometric macros and techniques'.)

It makes our geometric intuition almost directly computable (*"a sketch is an algorithm"*). It does so in a *coordinate-free* manner, and in n dimensions.

It has been dormant for about 100 years (from 1878 to 1984); it was recently used to unify geometry in theoretical physics (e.g. by Riesz and Hestenes); and it could now revolutionalize the computational treatment of geometry in the real-world-related computer sciences of vision, robotics and simulation.

This presentation conveys the basics of Clifford algebra, and presents some 'geometric macros' to show that it is indeed a 'geometric algebra'. It is too short to convey the full richness of geometric algebra – but I hope it gets you interested in learning more!

7 Vector spaces V^n 'over scalars' such as $\mathbb{R}$

We start with a vector space over the real numbers $\mathbb{R}$. In V^2, elements of the form:

$$\{\alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 \mid \alpha_1, \alpha_2 \in \mathbb{R}\}$$

The linear mappings on this vector space form a *(linear) algebra*: the product of mappings is again a mapping, etcetera.

But this is a bit indirect. We would like to have an algebra *directly* on vectors, in which the *product of two vectors* is again in the algebra.

Grassmann (1862) and Clifford(1878) found a way to do this – but they needed to include elements other than vectors to get a useful algebra. As we will see, these are precisely the 'areas', 'volumes', etcetera which we desired. Also, scalars will not be subservient to vectors, but full elements of the algebra.

So: we are going to make an algebra with products between scalars, vectors, areas, volumes etcetera. Such an algebra is called a *Grassmann algebra* (or *exterior algebra*) if we do not use a metric; but if we do – and this is much more useful – we get a *Clifford algebra.*

8 The Clifford geometric product

We introduce a *product ab between elements a and b* by the properties:

1. it is linear in each of the arguments
2. it is associative: $(ab)c = a(bc)$
3. it is *not* necessarily commutative: ab may differ from ba
4. demand *closure*, i.e. applicability between all elements
5. for *scalars* set it equal to the usual commutative scalar product in a vector space
6. for any *vector* $\mathbf{a}$, $\mathbf{aa}$ must equal a *scalar* (denoted $Q(\mathbf{a})$)

That is all – this product contains all we need for geometry (plus a notion of differentiation, i.e. limits). Everything we want to do must be expressed in it (augmented by calculus if desired).

You can choose only two things: the dimensionality n and the scalar-valued vector-function Q. And there the only essential choice (up to scaling) turns out to be the sign of Q for a given $\mathbf{a}$.

The set of elements after closure is called a *Clifford algebra*, and we write

$$\mathcal{C}\ell_{p,q}$$

if it has a basis of p positive vectors, and q negative vectors ($p+q = n$).

9 Inner and outer product

Consider vectors $\mathbf{a}$ and $\mathbf{b}$.

- No commutation for $\mathbf{ab}$; consider the symmetric expression:

  $$\mathbf{ab}+\mathbf{ba} = (\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b})-\mathbf{aa}-\mathbf{bb} = Q(\mathbf{a}+\mathbf{b})-Q(\mathbf{a})-Q(\mathbf{b}),$$

  so this is a *scalar product of vectors*. Choose Q such that this corresponds to the classical *inner product*:

  $$\mathbf{a}\cdot\mathbf{b} \equiv \tfrac{1}{2}(\mathbf{ab}+\mathbf{ba}).$$

  Thus the familiar inner product is still usable. (Phew!)

- What is left is the anti-symmetric part of $\mathbf{ab}$. This defines the *outer product*:

  $$\mathbf{a}\wedge\mathbf{b} \equiv \tfrac{1}{2}(\mathbf{ab}-\mathbf{ba}).$$

  This is new; and $\mathbf{a}\wedge\mathbf{b}$ is a new 'geometric object' which we need to understand.

  (It is somewhat strange: in a real Euclidean space with orthonormal basis $\{\mathbf{e}_i\}$ (so that $\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}$) we compute:

  $$\begin{aligned}(\mathbf{e}_i\wedge\mathbf{e}_j)^2 &= (\mathbf{e}_i\mathbf{e}_j-\mathbf{e}_i\cdot\mathbf{e}_j)^2 = (\mathbf{e}_i\mathbf{e}_j)^2 = \mathbf{e}_i\mathbf{e}_j\mathbf{e}_i\mathbf{e}_j = \mathbf{e}_i(\mathbf{e}_j\mathbf{e}_i)\mathbf{e}_j \\ &= \mathbf{e}_i(\mathbf{e}_j\wedge\mathbf{e}_i)\mathbf{e}_j = -\mathbf{e}_i(\mathbf{e}_i\wedge\mathbf{e}_j)\mathbf{e}_j = -\mathbf{e}_i\mathbf{e}_i\mathbf{e}_j\mathbf{e}_j \\ &= -\mathbf{e}_i^2\mathbf{e}_j^2 = -1.\end{aligned}$$

  So in this space, $\mathbf{e}_1\wedge\mathbf{e}_2$ is an object of which the square is negative – therefore it cannot be a real scalar or a vector, it is something new. We call it a *bivector*.)

10 Bivectors in the standard model

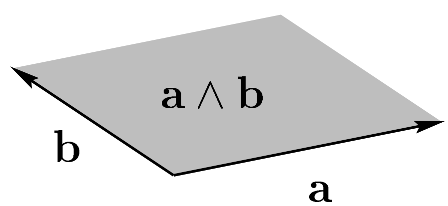
The anti-symmetric outer product $\mathbf{a}\wedge\mathbf{b}$ produces a *bivector*. Bivectors can be used to represent several geometrical objects, depending on our mapping between reality and Clifford algebra.

- In the **standard model**, a vector $\mathbf{a}$ represents a *direction* from the origin to a point in our space.

  If we have a space with basis unit vectors $\{\mathbf{e}_1, \mathbf{e}_2\}$ in the $(\mathbf{a}, \mathbf{b})$-plane, then:

$$\begin{aligned}
&\mathbf{a}\wedge\mathbf{b} = \\
&\quad = (\alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2)\wedge(\beta_1\mathbf{e}_1 + \beta_2\mathbf{e}_2) \\
&\quad = \alpha_1\beta_1\mathbf{e}_1\wedge\mathbf{e}_1 + \alpha_2\beta_2\mathbf{e}_2\wedge\mathbf{e}_2 + \alpha_1\beta_2\mathbf{e}_1\wedge\mathbf{e}_2 + \alpha_2\beta_1\mathbf{e}_2\wedge\mathbf{e}_1 \\
&\quad = 0 + 0 + (\alpha_1\beta_2 - \alpha_2\beta_1)\mathbf{e}_1\wedge\mathbf{e}_2.
\end{aligned}$$

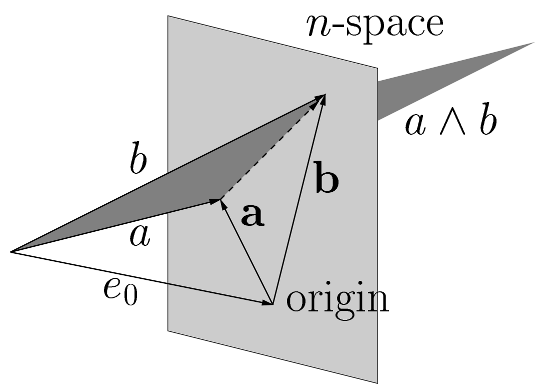
  The factor $(\alpha_1\beta_2 - \alpha_2\beta_1)$ is the scalar value of the directed area between $\mathbf{a}$ and $\mathbf{b}$. Apparently $\mathbf{e}_1\wedge\mathbf{e}_2$ indicates the plane, i.e. the 'two-dimensional direction' in which this scalar measure resides.



  Thus in the standard model, $\mathbf{a}\wedge\mathbf{b}$ is a *directed area segment* in n-dimensional space.

11 Bivectors in the homogeneous model

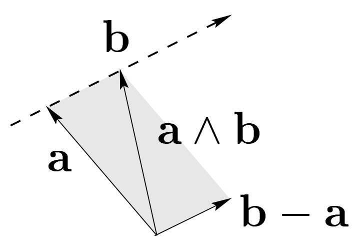
- In the **homogeneous model** of real Euclidean n-space, we consider quantities called *points* as represented as a vector $a = e_0 + \mathbf{a}$, with $\mathbf{a}$ a vector in n-space, and e_0 a vector in an $(n+1)$-dimensional space containing the n-space. Thus an *n-D point* is represented as an *$(n+1)$-D vector* pointing to it.



The outer product between vectors in $(n+1)$-space gives:

$$a \wedge b = (e_0 + \mathbf{a}) \wedge (e_0 + \mathbf{b}) = e_0 \wedge (\mathbf{b} - \mathbf{a}) + (\mathbf{a} \wedge \mathbf{b}).$$

We recognize $(\mathbf{b} - \mathbf{a})$ as the direction vector (or *tangent*) of the line segment from a to b, and $\mathbf{a} \wedge \mathbf{b}$ as its *moment*.



Thus $a \wedge b$, in the homogeneous model ($(n+1)$-space), represents the line segment from the point a to the point b in n-space.

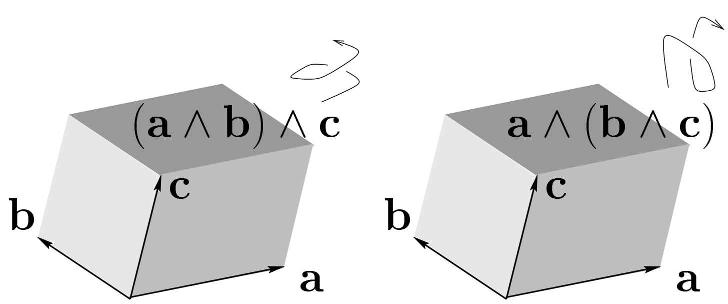
12 Extended outer products

We extend the outer product by linearity and associativity to a totally anti-symmetric product on $\mathcal{C}\ell_{p,q}$:

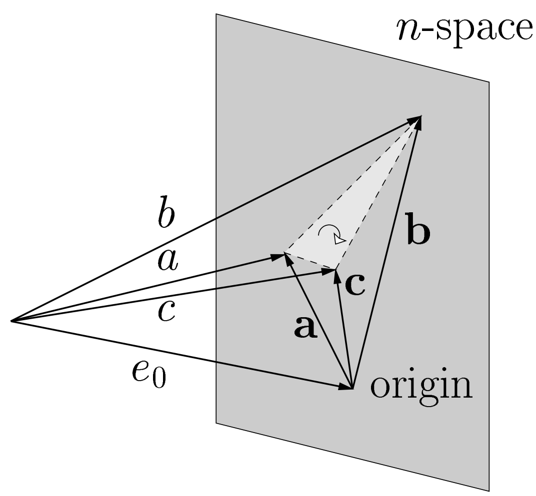
$$\mathbf{a} \wedge (\mathbf{b} \wedge \mathbf{c}) = (\mathbf{a} \wedge \mathbf{b}) \wedge \mathbf{c} \equiv \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}.$$

Its semantics depends again on the model:

- **standard model**: directed hypervolumes



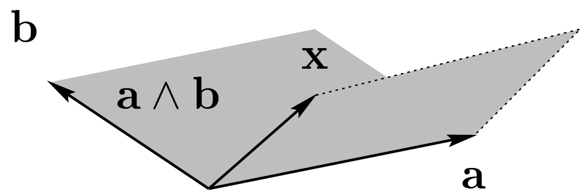
- **homogeneous model**: directed faces of simplexes



$a \wedge b \wedge c$ represents a simplex face; $a \wedge b$ represents a simplex edge; a represents a simplex vertex.

13 Blades are subspaces

If $\mathbf{x} \wedge (\mathbf{a} \wedge \mathbf{b}) = 0$, then $\mathbf{x}$ spans no volume with $(\mathbf{a} \wedge \mathbf{b})$; so $\mathbf{x}$ is in the $(\mathbf{a}, \mathbf{b})$-plane.



Generalize:

$$\mathbf{x} \wedge (\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \cdots \wedge \mathbf{a}_k) = 0 \quad \Longleftrightarrow \quad \mathbf{x} \text{ in } (\mathbf{a}_1, \mathbf{a}_2, \cdots \mathbf{a}_k) - \text{space}$$

Thus elements which can be written as an outer product of vectors represent linear subspaces. We call them *blades*.

The proper definition of the (anti-symmetric) outer product of vector and a k-blade can be shown to be:

$$\mathbf{a} \wedge \mathbf{A} \equiv \tfrac{1}{2}(\mathbf{a}\mathbf{A} + (-1)^k \mathbf{A}\mathbf{a}),$$

and therefore for the inner product (what is left of $\mathbf{a}\mathbf{A}$):

$$\mathbf{a} \cdot \mathbf{A} \equiv \tfrac{1}{2}(\mathbf{a}\mathbf{A} - (-1)^k \mathbf{A}\mathbf{a}).$$

(For completeness, we need to specify more to complete the definition of the inner product for arbitrary elements, but we will not need that in this talk.) So:

$$\mathbf{x} \in \mathbf{A} \quad \Longleftrightarrow \quad \mathbf{x}\mathbf{A} = -(-1)^k \mathbf{A}\mathbf{x}$$

14 Perpendicularity

We can now split any vector $\mathbf{x}$ into a part $\mathbf{x}_\parallel$ *contained in* $\mathbf{A}$ and a part $\mathbf{x}_\perp$ *perpendicular* to $\mathbf{A}$ (so $\mathbf{x}_\perp \cdot \mathbf{A} = 0$). We desire:

$$\mathbf{x}_\parallel \wedge \mathbf{A} = 0 \ \text{ and } \ \mathbf{x}_\perp \cdot \mathbf{A} = 0.$$

Using this, we find:

$$\mathbf{x}_\perp \mathbf{A} = \mathbf{x}_\perp \cdot \mathbf{A} + \mathbf{x}_\perp \wedge \mathbf{A} = \mathbf{x}_\perp \wedge \mathbf{A} = \mathbf{x}_\perp \wedge \mathbf{A} + \mathbf{x}_\parallel \wedge \mathbf{A} = \mathbf{x} \wedge \mathbf{A}$$

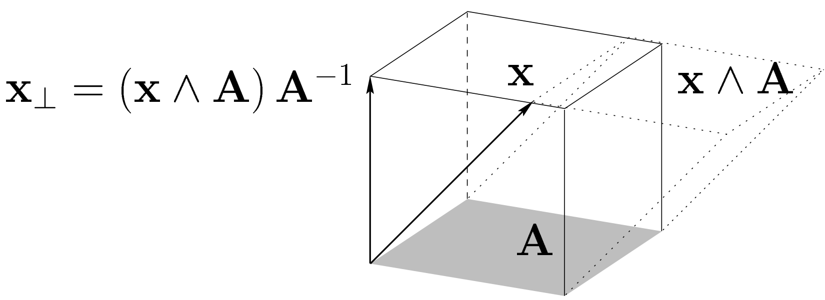
So that:

$$\mathbf{x}_\perp \equiv (\mathbf{x} \wedge \mathbf{A})\mathbf{A}^{-1}$$

and similarly:

$$\mathbf{x}_\parallel \equiv (\mathbf{x} \cdot \mathbf{A})\mathbf{A}^{-1}$$

(with $\mathbf{A}^{-1}$ defined by $\mathbf{A}\mathbf{A}^{-1} = 1$.)

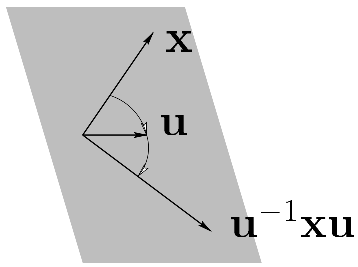


This thus confirms our hope in the introduction, and gives a *projection operator* and a *rejection operator*.

15 Reflection through commutation

Note that for the components of $\mathbf{x}$ relative to a 1-dimensional subspace characterized by a vector $\mathbf{u}$ we have:

$$\mathbf{u}^{-1}\mathbf{x}\mathbf{u} = \mathbf{u}^{-1}(\mathbf{x}_\perp + \mathbf{x}_\|)\mathbf{u} = \mathbf{u}^{-1}\mathbf{u}(-\mathbf{x}_\perp + \mathbf{x}_\|) = \mathbf{x}_\| - \mathbf{x}_\perp.$$



Therefore $\mathbf{u}^{-1}\mathbf{x}\mathbf{u}$ is a *reflection* of $\mathbf{x}$ in $\mathbf{u}$.

The expression $\mathbf{u}^{-1}\mathbf{x}\mathbf{u}$ cleverly uses commutation relative to the geometric product to construct reflection. We'll see that this is an extendable technique!

16 Closure produces Clifford algebra of $\mathbb{R}^2$

Pick an orthonormal basis $(\mathbf{e}_1, \mathbf{e}_2)$ in $\mathbb{R}^2$ such that:

$$\mathbf{e}_1\mathbf{e}_1 = \mathbf{e}_2\mathbf{e}_2 = 1 \quad \text{and} \quad (\mathbf{e}_1 \cdot \mathbf{e}_2) = 0$$

It follows that

$$\mathbf{e}_1 \wedge \mathbf{e}_2 = \mathbf{e}_1\mathbf{e}_2 = -\mathbf{e}_2\mathbf{e}_1.$$

Then evaluate the product $\mathbf{xy}$ of two vectors in $\mathbb{R}^2$:

$$\begin{aligned}
\mathbf{xy} &= \mathbf{x} \cdot \mathbf{y} + \mathbf{x} \wedge \mathbf{y} \\
&= (x_1y_1 + x_2y_2) + (x_1y_2 - x_2y_1)\,\mathbf{e}_1 \wedge \mathbf{e}_2 \\
&= \alpha + \beta\,\mathbf{e}_1 \wedge \mathbf{e}_2
\end{aligned}$$

Do this again (we want closure!):

$$\begin{aligned}
\mathbf{z}(\mathbf{xy}) &= (z_1\mathbf{e}_1 + z_2\mathbf{e}_2)(\alpha + \beta\,\mathbf{e}_1 \wedge \mathbf{e}_2) \\
&= (z_1\mathbf{e}_1 + z_2\mathbf{e}_2)(\alpha + \beta\,\mathbf{e}_1\mathbf{e}_2) \\
&= \alpha z_1\,\mathbf{e}_1 + \beta z_2\,\mathbf{e}_1\mathbf{e}_1\mathbf{e}_2 + \alpha z_2\,\mathbf{e}_2 + \beta z_2\mathbf{e}_2\,\mathbf{e}_1\mathbf{e}_2 \\
&= (\alpha z_1 - \beta z_2)\,\mathbf{e}_1 + (\alpha z_2 + \beta z_1)\,\mathbf{e}_2
\end{aligned}$$

This is a vector. Closure! We have found scalars, vectors and bivectors, and that's it for $\mathbb{R}^2$. (This result is independent of the basis used for $\mathbb{R}^2$.)

17 The full Clifford algebra

So $\mathbb{R}^2$ gives only only terms made up of scalar multiples of:

$$\underbrace{1}_{\text{scalars}}\;,\;\underbrace{\mathbf{e}_1,\;\mathbf{e}_2}_{\text{directions}}\;,\;\underbrace{\mathbf{e}_1\wedge\mathbf{e}_2}_{\text{directed area}}$$

In $\mathbb{R}^3$, we get objects on the 'basis':

$$\underbrace{1}_{\text{scalars}}\;,\underbrace{\mathbf{e}_1,\;\;\mathbf{e}_2,\;\mathbf{e}_3}_{\text{directions}},\;\;\underbrace{\mathbf{e}_1\wedge\mathbf{e}_2,\;\mathbf{e}_2\wedge\mathbf{e}_3,\;\mathbf{e}_3\wedge\mathbf{e}_1}_{\text{directed areas}},\;\;\underbrace{\mathbf{e}_1\wedge\mathbf{e}_2\wedge\mathbf{e}_3}_{\text{directed volume}}$$

For V^n, we denote such a 'basis' by $\wedge V^n$, and each of the m-dimensional subspaces by $\wedge^m V^n$. Note that there are in general 2^n elements, each of the $\wedge^m V^n$ contributing $\frac{n!}{m!(n-m)!}$ independent terms to the basis.

Such a set, endowed with the geometric (Clifford) product between terms, forms a *Clifford algebra* of V^n, which is denoted $\mathcal{G}(V^n)$ or $\mathcal{C}\ell_n$. If we desire to denote an algebra with p vectors with positive square and q with negative square, we write $\mathcal{C}\ell_{p,q}$.

$\mathbf{e}_1\wedge\mathbf{e}_2\wedge\cdots\wedge\mathbf{e}_n$, the n-dimensional directed volume, is often called a *pseudoscalar* of V^n and denoted $\mathbf{I}_n$ or even $\mathbf{I}$.

18 Some important algebraic stuff: inverses

An *inverse* of an element a of the Clifford algebra is defined by $aa^{-1} = 1$ (right inverse) or $a^{-1}a = 1$ (left inverse). Not all elements have inverses; and even if they do, they may be hard to compute.

However, an important class of objects which have an inverse are the *versors*. A *versor* is an element which can be written as a *geometric* product of vectors: $a = \mathbf{a}_1\mathbf{a}_2 \cdots \mathbf{a}_k$. The inverse is:

$$a^{-1} = \frac{\tilde{a}}{\tilde{a}a}$$

Here $\tilde{a}$, the *reverse* of a, is defined as

$$\tilde{a} \equiv \mathbf{a}_k \cdots \mathbf{a}_2\mathbf{a}_1 = (-1)^{\frac{1}{2}k(k-1)}a.$$

Proof: $\tilde{a}a = \mathbf{a}_k \cdots \mathbf{a}_2\mathbf{a}_1\mathbf{a}_1\mathbf{a}_2 \cdots \mathbf{a}_k = \mathbf{a}_k \cdots \mathbf{a}_2 Q(\mathbf{a}_1)\mathbf{a}_2 \cdots \mathbf{a}_k = \cdots = Q(\mathbf{a}_1)Q(\mathbf{a}_2)\cdots Q(\mathbf{a}_k)$ which is a scalar, commuting with everything, so $(\tilde{a}a)a^{-1} = \tilde{a}(aa^{-1}) = \tilde{a}$ gives the result. □

For a vector $\mathbf{v}$, we get $\mathbf{v}^{-1} = \frac{\mathbf{v}}{Q(\mathbf{v})} = \frac{\mathbf{v}}{|\mathbf{v}|^2}$; for a unit vector $\mathbf{e}$ we get $\mathbf{e}^{-1} = \mathbf{e}$.

For a product of two vectors, which we may express as $\alpha + \beta\,\mathbf{e}_1 \wedge \mathbf{e}_2$, we get:

$$(\alpha + \beta\,\mathbf{e}_1 \wedge \mathbf{e}_2)^{-1} = \frac{(\alpha - \beta\,\mathbf{e}_1 \wedge \mathbf{e}_2)}{\alpha^2 + \beta^2}$$

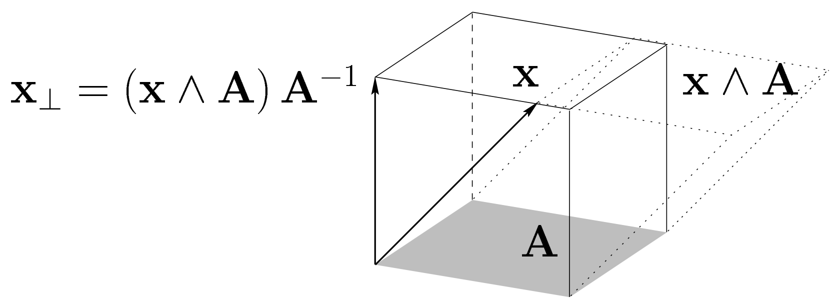
19 Division

Division in the algebra is (obviously!) done through multiplication by the inverse, if it exists. So, for instance:

$$a\,x = b \iff x = a^{-1}b. \tag{1}$$

Division is non-commutative: $a^{-1}b \neq ba^{-1}$, so do *not* use the notation $\frac{b}{a}$, except for scalars!

Now the earlier constructions like $(\mathbf{x}\wedge\mathbf{A})\mathbf{A}^{-1}$ make sense (we only missed the non-commutative essentials that make it work).



20 Duality

The highest order subspace in a space is the *pseudoscalar*:

$$\mathbf{I} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \cdots \wedge \mathbf{e}_n.$$

Dividing by the pseudoscalar gives the *dual* of an element:

$$\mathbf{A}^* \equiv \mathbf{A}\mathbf{I}^{-1}$$

There is provable *duality between inner and outer product* (i.e. between 'spanning' and 'orthogonal complement'):

$$(\mathbf{x} \cdot \mathbf{A})\mathbf{I}^{-1} = \mathbf{x} \wedge (\mathbf{A}\mathbf{I}^{-1}) \quad \text{and} \quad (\mathbf{x} \wedge \mathbf{A})\mathbf{I}^{-1} = \mathbf{x} \cdot (\mathbf{A}\mathbf{I}^{-1}).$$

Therefore we do *not* need normal vectors anymore to denote a hyperplane. In 3-D just use bivectors, it is equivalent:

$$0 = \mathbf{x} \wedge \mathbf{i} = \left(\mathbf{x} \cdot (\mathbf{i}\mathbf{I}^{-1})\right)\mathbf{I} = (\mathbf{x} \cdot \mathbf{n})\mathbf{I}$$

where we define $\mathbf{n} \equiv \mathbf{i}\mathbf{I}^{-1}$.

Using $\mathbf{i}$ is better than using $\mathbf{n}$:

21 Subspace representations: avoid the normal vector!

subspace containment is affine invariant, and it was always a bad custom to use perpendicularity (non-affine) to encode it! Outer products transform very simply under a linear transformation $f : V^n \to V^n$:

$$f(\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \cdots \wedge \mathbf{a}_k) = f(\mathbf{a}_1) \wedge f(\mathbf{a}_2) \wedge \cdots \wedge f(\mathbf{a}_k).$$

Very important: linear transformations preserve the outer product, therefore the outer product is very important in characterizing linear algebra! They are much more specific than determinants and minors (which are merely the magnitudes of outer products). We call the extension of f to blades an *outermorphism*.

Linear transformations do *not* preserve the *inner* product: $f(a \cdot b) = \overline{f}^{-1}(a) \cdot f(b)$ (where $\overline{f}$ is the *adjoint* of f; see elsewhere.) Therefore, encode subspaces by *blades* rather than normal vectors!

22 The meet operation

The outer product ‘spans’ spaces, it is like taking a union of direction vectors. It is sometimes called the *join* operation.

There is also an *intersection* of spaces – but it depends on the smallest common subspace $\mathbf{I}$. It is the *meet* operation, defined through:

$$a \vee_{\mathbf{I}} b = (a\mathbf{I}^{-1}) \cdot b$$

or, easier to remember, with duality in $\mathbf{I}$:

$$(a \vee b)^* = a^* \wedge b^*.$$

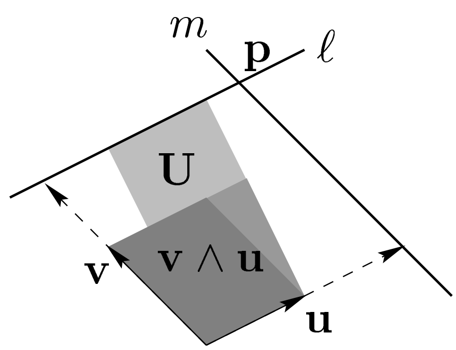
It is a quantitative intersection, giving an intersection set and an intersection strength (useful for numerically stable computations).

23 The semantics of the meet operation

Its meaning depends on the model:

- **standard model**:
  The meet is the *sine of the smallest angle* between subspaces, a familiar distance measure from numerical analysis denoting the 'parallelism' of spaces.

- **homogeneous model**:
  The meet indeed gives the intersection point of spaces:

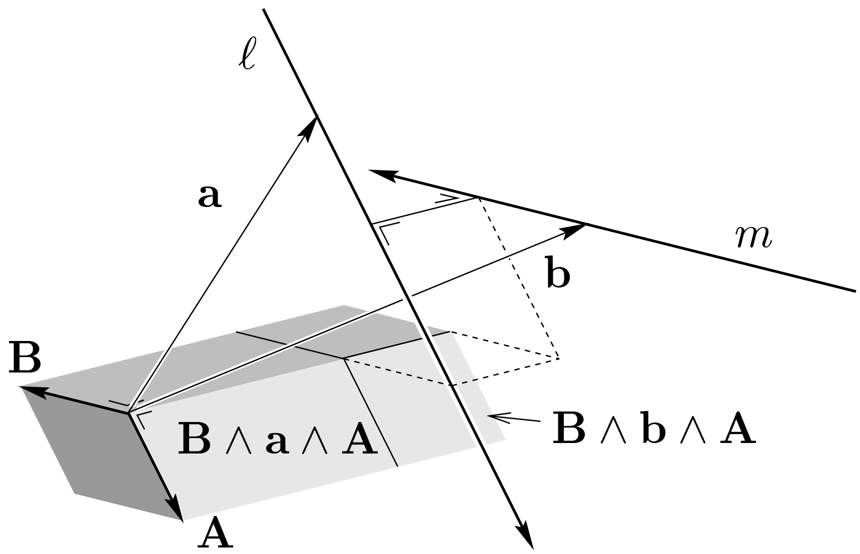


$$p = \ell \vee m = (\mathbf{v} \wedge \mathbf{u})^* \left(e_0 + \mathbf{U}(\mathbf{u} \wedge \mathbf{v})^{-1}\,\mathbf{v} + \mathbf{V}(\mathbf{v} \wedge \mathbf{u})^{-1}\,\mathbf{u}\right),$$

which is basically what we expected on slide 4, weighted by a factor giving the numerical significance of the meet.

24 The meet in the homogeneous model (continued)

If the objects *meet in a scalar*, this is precisely the Euclidean distance between the objects:



$$\delta = \ell \vee m = (\mathbf{B} \wedge \mathbf{a} \wedge \mathbf{A} - \mathbf{B} \wedge \mathbf{a} \wedge \mathbf{A})\mathbf{I}^{-1}.$$

A coordinate-free pictorial algorithm!

Again: just *one* operation in Clifford algebra – but with *many* meanings in geometrical worlds.

25 Modeling geometries by Clifford algebra

We have seen how a *bivector* from the appropriate Clifford algebra could be used to represent different objects in an application; the algebraic essence which it captures is that it is anti-symmetric. Both areas and lines are, so it can represent them.

Similarly, we saw how a single operation such as the *meet* implements many different operations – they turned out to be algebraically identical, in the proper model.

We claim that *any* two-term geometric object which is anti-symmetric in its arguments can be represented as $a \wedge b$ in some properly chosen Clifford algebra; and that *any* 'incidence relationship' is a meet in disguise. And we claim that similar truths hold for other geometric objects and operators.

If this is true, then *all we need to implement and study* are those Clifford algebras, and the 'interfaces' to them. This is the hope and essence of *geometric algebra*.

26 So far, so good

Sofar, then, Clifford algebra has given us a richer set of tools to do geometry:

- subspaces as basic elements, replacing normal vectors etc.,
- a full algebra: we can divide, span, project, intersect, etc.
- by selecting the proper Clifford algebra we can represent different geometric objects and get their algebraic (and therefore algorithmic) properties for free.

This is already quite something. But there is more, and it concerns ease and computational efficiency in the representation of *transformations* using *versors*. It is a somewhat advanced subject.

27 Rotations and scalings in Euclidean space

Consider a vector $\mathbf{u}$ in E^n, relative to some 'standard' unit vector $\mathbf{e}$ in E^n. We can see $\mathbf{u}$ as '*produced from* $\mathbf{e}$ *by a Clifford product with an unknown object* a' from the Clifford algebra of E^n. So set:

$$\mathbf{u} = \mathbf{e}\, a$$

We get (using $\mathbf{e}^{-1} = \mathbf{e}$ since $\mathbf{e}$ is a unit vector):

$$a = \mathbf{e}^{-1}\mathbf{u} = \mathbf{e}\mathbf{u},$$

so the unknown object a is the Clifford product of two vectors.

Develop $\mathbf{u}$ on an orthonormal basis in the $(\mathbf{e}, \mathbf{u})$-plane. Take as basis: $\{\mathbf{e}_1 = \mathbf{e}\,,\ \mathbf{e}_2\}$, so $\mathbf{u} = u_1\,\mathbf{e}_1 + u_2\,\mathbf{e}_2$ (with $u_1^2 + u_2^2 = Q(\mathbf{u}) \equiv |\mathbf{u}|^2$, so we can set $u_1 = |\mathbf{u}|\cos\phi$ and $u_2 = |\mathbf{u}|\sin\phi$ – just a parametrization.) Then:

$$\begin{aligned} a &= \mathbf{e}_1(u_1\,\mathbf{e}_1 + u_2\,\mathbf{e}_2) = u_1 + u_2\,\mathbf{e}_1 \wedge \mathbf{e}_2 \\ &= |\mathbf{u}|\,(\cos\phi + \sin\phi\,\mathbf{e}_1 \wedge \mathbf{e}_2) = |\mathbf{u}|\,e^{(\mathbf{e}_1 \wedge \mathbf{e}_2)\,\phi} \end{aligned}$$

The last step above can be seen as a notational shorthand which follows naturally by remembering that $(\mathbf{e}_1 \wedge \mathbf{e}_2)^2 = -1$, so symbolically (setting $\mathbf{i} \equiv \mathbf{e}_1 \wedge \mathbf{e}_2$ for convenience):

$$\begin{aligned} e^{\mathbf{i}\phi} &= 1 + (\mathbf{i}\phi) + \frac{(\mathbf{i}\phi)^2}{2!} + \frac{(\mathbf{i}\phi)^3}{3!} + \frac{(\mathbf{i}\phi)^4}{4!} + \cdots \\ &= 1 + \mathbf{i}\phi - \frac{\phi^2}{2!} - \frac{\mathbf{i}\phi^3}{3!} + \frac{\phi^4}{4!} + \cdots = \cos\phi + \mathbf{i}\sin\phi. \end{aligned}$$

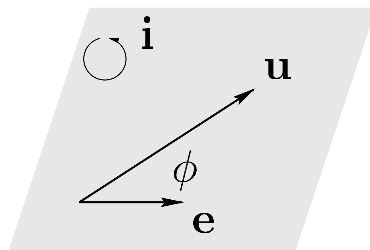
(Looks familiar? It is just real geometric algebra, so you must be wrong: no complex numbers anywhere... See slide 36.)

28 Angles

So any desired vector $\mathbf{u}$ can be made from a unit vector $\mathbf{e}$ as:

$$\mathbf{u} = \mathbf{e}\,|\mathbf{u}|\,e^{\mathbf{i}\phi}$$

with $\mathbf{i}$ a unit bivector spanning the $(\mathbf{e}, \mathbf{u})$-plane, ϕ the angle between $\mathbf{e}$ and $\mathbf{u}$ in that plane. This works in n-dimensional space!



We may as well take $\mathbf{i}\phi$ as *the definition of the angle* between $\mathbf{e}$ and $\mathbf{u}$. This angle is constructive: we can use it to make $\mathbf{u}$ from $\mathbf{e}$ by its exponential.

Bivectors are like angles. The exponent of a bivector is an object of the Clifford algebra with a scalar and a bivector part. It acts like a *rotation operation* under the Clifford product. It is called a *spinor*.

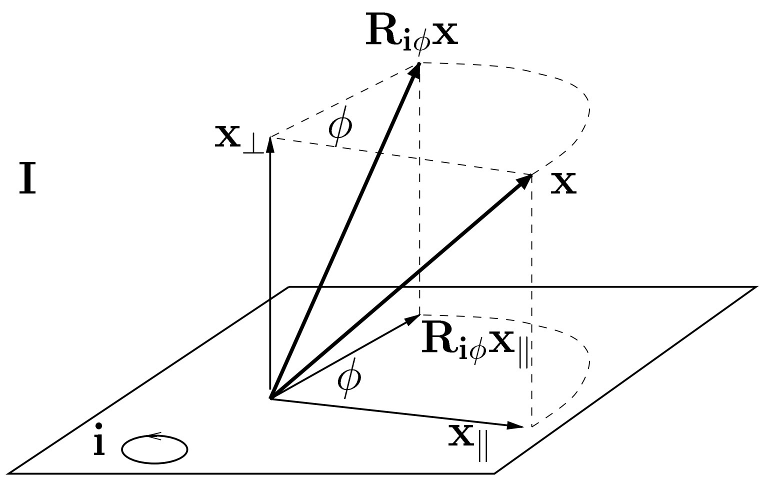
An arbitrary vector is determined by a standard unit vector $\mathbf{e}$, a scalar norm $|\mathbf{u}|$, and a bivector angle $\mathbf{i}\phi$. These are precise polar coordinates, in n-D specifying the plane they work in.

(Factoid: for $\phi = \pi/2$ we get $e^{\mathbf{i}\phi} = \mathbf{i}$. Turning $\mathbf{e}$ twice over $\pi/2$ produces $-\mathbf{e}$, in a 'U-turn'; so $\mathbf{eii} = -\mathbf{e}$, and therefore $\mathbf{i}^2 = -1$, as we know.)

as:

29 Rotations in space ($\mathbb{R}^3$)

Attempt to denote a rotation of a vector $\mathbf{x}$ over the angle $\mathbf{i}\phi$ (note how this denotes the plane of rotation!). What we want is:



Essential is 'lying in the $\mathbf{i}$-plane', so (anti-)commutativity. Split $\mathbf{x} = \mathbf{x}_\parallel + \mathbf{x}_\perp$ and form (*a great trick!*):

$$\begin{aligned}
e^{-\mathbf{i}\phi}\,\mathbf{x}\,e^{\mathbf{i}\phi} &= (\cos\phi - \mathbf{i}\sin\phi)(\mathbf{x}_\perp + \mathbf{x}_\parallel)(\cos\phi + \mathbf{i}\sin\phi) \\
&= \mathbf{x}_\perp(\cos^2\phi + \sin^2\phi) + \mathbf{x}_\parallel(\cos\phi + \mathbf{i}\sin\phi)^2 \\
&= \mathbf{x}_\perp + \mathbf{x}_\parallel(\cos 2\phi + \mathbf{i}\sin 2\phi) \\
&= \mathbf{x}_\perp + \mathbf{x}_\parallel e^{2\mathbf{i}\phi}
\end{aligned}$$

So $\mathbf{x}_\perp$ is unchanged, and $\mathbf{x}_\parallel$ rotates over an angle $2\mathbf{i}\phi$. Therefore:

$$\mathbf{R}_{\mathbf{i}\phi}\,\mathbf{x} = e^{-\mathbf{i}\phi/2}\,\mathbf{x}\,e^{\mathbf{i}\phi/2}$$

Looks complicated – but just commutation, really.

30 Concatenation of rotations

A rotation of a vector is fully characterized by the spinor $e^{\mathbf{i}\phi/2}$:

$$\mathbf{R}_{\mathbf{i}\phi}\,\mathbf{x} = e^{-\mathbf{i}\phi/2}\,\mathbf{x}\,e^{\mathbf{i}\phi/2}$$

We can consider the spinor $e^{\mathbf{i}\phi/2}$ as 'representing the rotation', independent of whether we want to use it on a vector or not.

Multiplication of rotations on vectors, first over $\mathbf{i}\phi$, then $\mathbf{j}\psi$:

$$\begin{aligned}\mathbf{R}_{\mathbf{j}\psi}\left(\mathbf{R}_{\mathbf{i}\phi}\,\mathbf{x}\right) &= e^{-\mathbf{j}\psi/2}\left(e^{-\mathbf{i}\phi/2}\,\mathbf{x}\,e^{\mathbf{i}\phi/2}\right)e^{\mathbf{j}\psi/2} \\ &= \left(e^{\mathbf{i}\phi/2}e^{\mathbf{j}\psi/2}\right)^{-1}\mathbf{x}\left(e^{\mathbf{i}\phi/2}e^{\mathbf{j}\psi/2}\right)\end{aligned}$$

so characterized by the spinor:

$$e^{\mathbf{i}\phi/2}\;e^{\mathbf{j}\psi/2}$$

Warning: this is *not* in general equal to $e^{(\mathbf{i}\phi/2+\mathbf{j}\psi/2)}$ – although it *is* when $\mathbf{i}$ and $\mathbf{j}$ commute.

(Footnote: in linear algebra, a rotation matrix contains not only the rotation, but also the consequences of wanting to make it act in a location representation: so it also depends on the coordinate system. A spinor does not! In spinors, no need for an eigenvector-analysis to see what it actually does.)

31 Axis of rotation; duality

We have now characterized rotations by their angles, which are *bivectors*. The rotation *axes* are simply *dual* to these.

Remember: *Duality is very simple in geometric algebra: divide by the (unit) pseudoscalar* $\mathbf{I}$ *of the algebra.*

In E^3, a *vector* can be used to denote a rotation axis (in E^n, *not*!). In E^3, the bivector $(\mathbf{e}_2 \wedge \mathbf{e}_3)\phi$ gives the rotation axis $\mathbf{e}_1\phi$, since:

$$(\mathbf{e}_2 \wedge \mathbf{e}_3)\phi\mathbf{I}^{-1} = (\mathbf{e}_2\mathbf{e}_3)\phi(\mathbf{e}_1\mathbf{e}_2\mathbf{e}_3)^{-1} = \mathbf{e}_2\mathbf{e}_3\mathbf{e}_3\mathbf{e}_2\mathbf{e}_1\phi = \mathbf{e}_1\phi$$

etcetera and vice versa:

$$(\mathbf{e}_2 \wedge \mathbf{e}_3)\phi = \mathbf{e}_1\phi\mathbf{I} = \mathbf{I}\mathbf{e}_1\phi.$$

In Euclidean 3-space, we can therefore represent the rotation over ϕ around the axis $\mathbf{v}$ by the spinor:

$$e^{\mathbf{I}\mathbf{v}\phi/2}.$$

which is a very direct way of specifying a rotation, in just the way we always wanted to: *axis-and-angle* in an oriented 3-dimensional space $\mathbf{I}$ (making the convention 'left/right-handed positive' explicit in the formula!). And we can compute with it directly (see slide 33).

32 Sense of rotation

There is also a *sense of rotation*: minus a spinor

$$-e^{\mathbf{i}\phi/2} = e^{-\mathbf{i}\pi}e^{\mathbf{i}\phi/2} = e^{-\mathbf{i}(2\pi-\phi)/2}$$

is an *opposite* rotation over the complementary angle $(2\pi - \phi)\mathbf{i}$.

The minus leaves result on a vector unchanged:

$$(-\sigma)^{-1}\mathbf{x}(-\sigma) = \sigma^{-1}\mathbf{x}\sigma.$$

Since the rotation matrices of linear algebra are only based on the *final* result on vectors, a rotation matrix can *not* indicate the sense of turning! Spinors are superior, in this sense `:-)`.

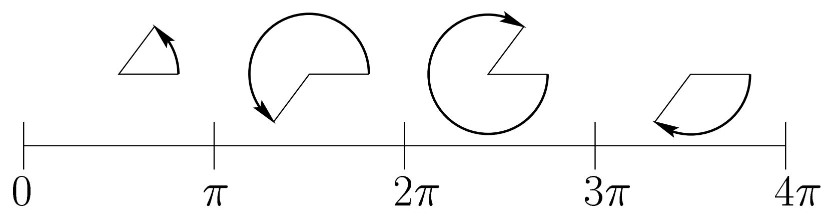
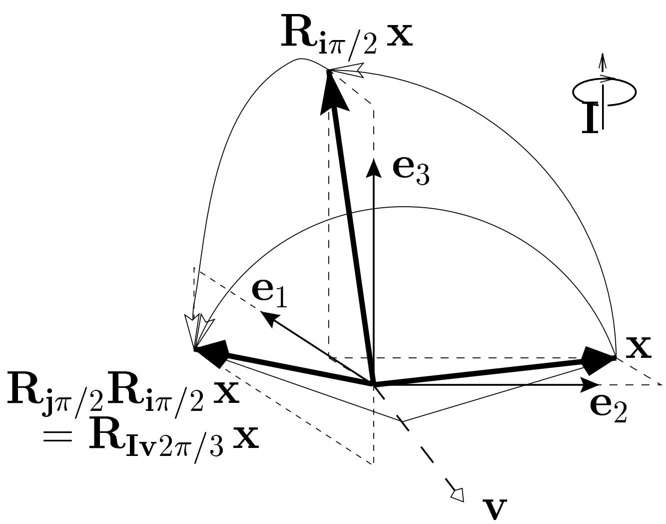


Illustration: as a function of ϕ, the spinor $e^{\mathbf{i}\phi/2}$ encodes different rotations in the $\mathbf{i}$-plane.

33 Example of rotations



Rotation over $\pi/2$ around $\mathbf{e}_1$ followed by rotation over $\pi/2$ around $\mathbf{e}_2$. What is the total rotation?

$$\begin{aligned}
e^{\mathbf{I}\mathbf{e}_1\pi/4}\; e^{\mathbf{I}\mathbf{e}_2\pi/4} &= \frac{1}{\sqrt{2}}(1+\mathbf{I}\mathbf{e}_1)\,\frac{1}{\sqrt{2}}(1+\mathbf{I}\mathbf{e}_2) \\
&= \tfrac{1}{2}\left(1+\mathbf{I}(\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_1\mathbf{I}\mathbf{e}_2)\right) \\
&= \tfrac{1}{2}\left(1+\mathbf{I}(\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_1\mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_2)\right) \\
&= \tfrac{1}{2}\left(1+\mathbf{I}(\mathbf{e}_1+\mathbf{e}_2-\mathbf{e}_3)\right) \\
&= \tfrac{1}{2}+\tfrac{1}{2}\sqrt{3}\,\mathbf{I}\left(\frac{\mathbf{e}_1+\mathbf{e}_2-\mathbf{e}_3}{\sqrt{3}}\right) \\
&= e^{\mathbf{I}\mathbf{v}\pi/3}
\end{aligned}$$

with $\mathbf{v}=\frac{\mathbf{e}_1+\mathbf{e}_2-\mathbf{e}_3}{\sqrt{3}}$. This represents a rotation over $\mathbf{v}$ over $2\pi/3$. A lot more work with rotation matrices!

34 Rotation as reflections

A rotation of a vector $\mathbf{x}$ is $\sigma^{-1}\mathbf{x}\sigma$, with σ a spinor $e^{\mathbf{i}\phi/2}$.

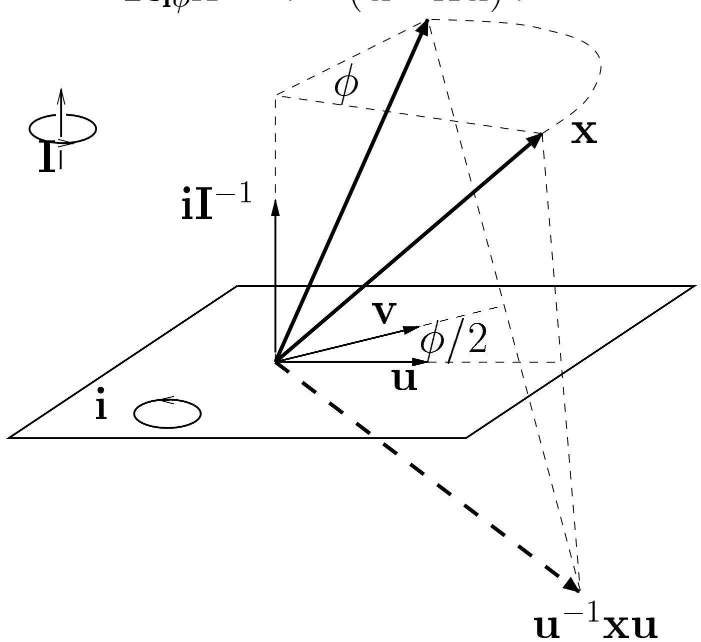
The unit spinor $e^{\mathbf{i}\phi/2}$ is the Clifford product of two vectors $\mathbf{u}$ and $\mathbf{v}$ making an angle $\phi/2$ in the $\mathbf{i}$-plane. So:

$$e^{-\mathbf{i}\phi/2}\,\mathbf{x}\,e^{\mathbf{i}\phi/2} = (\mathbf{uv})^{-1}\mathbf{x}(\mathbf{uv}) = \mathbf{v}^{-1}(\mathbf{u}^{-1}\mathbf{x}\mathbf{u})\mathbf{v}.$$

This is the concatenation of two mappings of the form $\mathbf{x} \mapsto \mathbf{u}^{-1}\mathbf{x}\mathbf{u}$, which is a *reflection* of $\mathbf{x}$ in $\mathbf{u}$ (see slide 14).

So *a rotation can be viewed as two reflections*:

$$\mathbf{R}_{\mathbf{i}\phi}\mathbf{x} = \mathbf{v}^{-1}(\mathbf{u}^{-1}\mathbf{x}\mathbf{u})\mathbf{v}$$



(Footnote: this representation is independent of the choice of the particular $\mathbf{u}$ and $\mathbf{v}$ in $\mathbf{i}$ – as long as they have an angle of $\mathbf{i}\phi/2$ between them.)

35 Linear transformations and versors

Any extension (as outermorphism) of a linear transformation (linear map from V^n to V^n) $\underline{F}$ in $\mathcal{C}\ell_{p,q}$ can be represented by a versor F through the *versor equation*:

$$\underline{F}\mathbf{x} = F\mathbf{x}\widehat{F}^{-1},$$

where *the versor F is the geometric product of vectors.* We have seen a reflection (F is a vector) and a rotation (F is a geometric product of two vectors, the exponent of a bivector). There is more, of course.

The beauty is that this applies to any blade; if the rotation of a vector $\mathbf{x}$ is:

$$\underline{R}\mathbf{x} = R\mathbf{x}R^{-1}.$$

then the roration of a bivector $\mathbf{A}$ is:

$$\underline{R}\mathbf{A} = R\mathbf{A}R^{-1},$$

etcetera. *In geometric algebra, operators can be represented independently of the 'objects' they operate on.* In the matrix representation of linear algebra, vectors are implicit in the representation – no wonder bivectors are not common there!

Combining versors with choosing the proper model gives an enormous scope for description. For instance, I found that *wave propagation* in E^n (which equals collision detection) can be represented as a versor product on tangent blades in $\mathcal{C}\ell_{n+1,1}$, a Minkowski space of two dimensions higher. This space enables versor representation of wave propagation viewed as direction-dependent translation, through a rotation around infinity of tangent blades of wave fronts.

36 Intermezzo: complex numbers and quaternions

In E^2 (with orthonormal basis), we have seen that the product $\mathbf{eu}$ of a unit vector $\mathbf{e} = \mathbf{e}_1$ and a vector $\mathbf{u} = u_1\,\mathbf{e}_1 + u_2\,\mathbf{e}_2$ is a spinor:

$$u_1 + u_2\,\mathbf{e}_1 \wedge \mathbf{e}_2$$

where $(\mathbf{e}_1 \wedge \mathbf{e}_2)^2 = -1$. Algebraically, spinors in E^2 are thus isomorphic to the complex numbers when we set $\mathbf{e}_1 \wedge \mathbf{e}_2 = i$.

Similar in E^3 (with orthonormal basis). Introduce $\mathbf{i}_1 \equiv \mathbf{e}_2 \wedge \mathbf{e}_3$ and $\mathbf{i}_2$, $\mathbf{i}_3$ cyclically. So spinors in E^3 are quantities of the form:

$$q_0\,1 + q_1\,\mathbf{i}_1 + q_2\,\mathbf{i}_2 + q_3\,\mathbf{i}_3$$

with $q_i \in \mathbb{R}$ and the $\mathbf{i}_i$ satisfying: $(\mathbf{i}_1\mathbf{i}_2\mathbf{i}_3)^2 = 1$, $\mathbf{i}_1^2 = \mathbf{i}_2^2 = \mathbf{i}_3^2 = -1$, $\mathbf{i}_1 = \mathbf{i}_3\mathbf{i}_2$ and cyclic. Thus isomorphic to *quaternions*, with their 'new' product actually just the geometric product.

This absorbs the 'tricks' of quaternions and complex numbers in a consistent geometrical framework. In any problem, we can now hope to invent our own tailored techniques!

The geometrical view of finding these structures is general, and extendible to $\mathbb{R}^n$ (the algebraic way was not). Each directed plane naturally has its own 'complex number' (its pseudoscalar), and the product relationships are simply due to the Clifford product.

So you can forget about them now, at least as anything special worth remembering...

37 High hopes

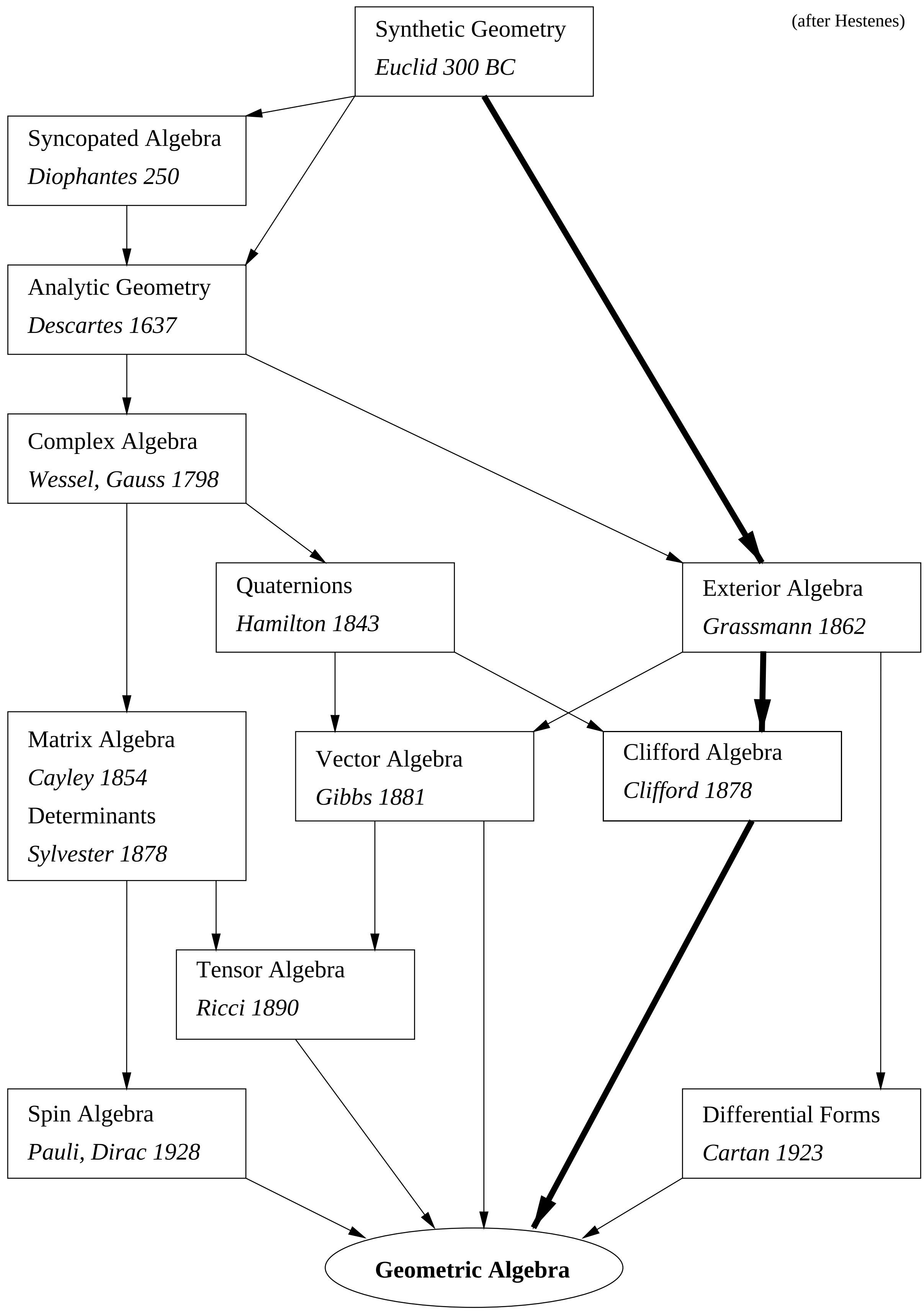
Geometric algebra gives us a wonderful set of tools to do geometry:

- subspaces as basic elements, replacing normal vectors etc.,
- a full algebra: we can divide, span, project, intersect, etc.
- by selecting the proper Clifford algebra we can represent different geometric objects and get their algebraic (and therefore algorithmic) properties for free.
- many common transformations (isometries, conformal mappings, wave propagation) permit an efficient versor representation in a properly selected algebra

All we need to study and implement are the Clifford algebras; they will fulfill all our geometrical needs.

For completeness: there is also a powerful geometric calculus; that is for another talk.

The Pedigree of Geometric Algebra



38 Notes on history

Intended by Clifford (1878) as a "grammar of space".

Why not used before?

- Vectors in V^n (by Gibbs 1881) worked well enough in 3D.
- Got absorbed into algebra, geometrical aspects neglected.

Why used now?

- Theoretical physics rediscovered parts (such as spinors).
- Hestenes 1968 re-emphasized geometrical interpretation.
- Computationally more efficient than linear algebra.
- Algorithmically constructive geometry requires unification of disparate classical analytic branches: linear algebra, differential geometry, Lie algebra, algebraic geometry.

Use for computer science?

- All computations can be done in single Clifford toolbox with unique and universal operators and clear data structures.
- Larger suite of operations, such as division by multivectors.
- No `if-then-else` and special cases, and n-D programs.
- Faster software (by 20% says NASA for 3D rotations).
- More natural, object-oriented teaching of geometry.

39 The internal structure of geometric algebra