GEOMETRY FOR COMPUTER APPLICATIONS
COURSE NOTES
MATH 431- FALL 2018
UNIVERSITY OF MARYLAND

WILLIAM M. GOLDMAN

Contents

Date: August 26, 2018.

1. Introduction

This course develops the theory of data types for computer applications. Specifically we develop algebraic data types for computer graphics, computer vision and robotics. If we take points, lines, lengths, angles, areas, etc. as the (extremely basic) building blocks of graphics, then an over arching theme is to discover ways in which to represent these geometric objects in a computer. Simultaneously, we aim to develop ways in which to represent in a computer the manner in which these geometric objects transform.

1.1. **Imagine designing a video game!**

Your avatar is flying a spaceship through a hazardous jungle populated by wild monsters, evil dinosaurs and poisonous plants, with dangerous objects zipping by. Throw in a few earthquakes, tsunamis, hurricanes and tornados, too, just for fun.

Of course enemies are chasing you, shooting rockets and subjecting your craft to waves of treacherous force fields. In addition to steering your vehicle, you need to be able to change your viewpoints and perspectives in order to fully gauge your direction and speed. Your instruments need to sense all the awful perils your adversary has aimed at you. Your survival, and the lives of millions of other people, depends on being able to manipulate — reliably, quickly, and in real time — huge amounts of graphical data by many types of geometric transformations: rotations, dilations, translations, reflections, and changes of perspective.

1.2. **Goals.**

- Due to total size of the graphical data, the data types must be compact and efficiently designed.
- Due to the demands of interactive use, the computations must be as fast as possible.
- Due to the demands of ever-changing technology, the code must be easy to debug, maintain, and update. Thus the data types and the manipulation routines must be readable, succinct and comprehensible to other programmers.
- The data will ultimately be vectors and matrices, and the mathematical routines basically linear algebra. Matrix operations are cheap, efficient and easy to implement. The compelling advantage of linear transformations is that —by using coordinates on a vector space defined by a basis — the geometric information is encoded in a *finite set* of numbers. It's only how they are manipulated which varies by their context.

Here are some examples of how abstraction and mathematical elegance are both means and end in regards to computer applications.

1.3. **Data types in plane geometry.** First consider the familiar case is that of points in the plane. Points are described uniquely by an ordered pair of numbers. Vector operations enable us to compute geometric relations (such as distance) between points. Furthermore transformations of the plane are conveniently described by matrices. The calculations are cheap to implement on a computer, easy to understand.

Trying to do the same for lines in the plane is more difficult and more interesting. However, programming a video game may require you to transform lines and, eventually, more complicated graphical objects, in a similar way. The reality is that lines in the plane are not as easy to parametrize as points in the plane: the set of lines does not admit a coordinate system as easy as just the (x, y)-coordinates which uniquely describe arbitrary points.

Here are some ways we describe lines in the plane. As you can see, none of them are as convenient as parametrizing points in the plane.

(1) Given two distinct points p_1 and p_2 (represented as 2-vectors), the line $\overleftrightarrow{p_1p_2}$ joining them is described by equations

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

or, in parametric form,

$$\begin{aligned} x &= x_1 + t(x_2 - x_1) \\ y &= y_1 + t(y_2 - y_1) \end{aligned}$$

This parametrization has the following drawbacks:

- The points p_1, p_2 are not unique and may not be so efficient to find; there are many pairs of points which determine a given line.
- Determining when two different pairs (p_1, p_2) determine the same line may be unnecesarily time-consuming.
- The initial data requires that $p_1 \neq p_2$. Checking this each time is necessary (and time-consuming).

(2) Given a slope $m \in \mathbb{R}$ and $b \in \mathbb{R}$, the line with slope m and y-intercept b has *slope-intercept form*

$$y = mx + b.$$

This efficiently parametrizes all *non-vertical* lines uniquely by the pair $(m, b) \in \mathbb{R}^2$. However, vertical lines have "infinite slope" $(m = \infty)$ and don't fit nicely into this parametrization. However they are parametrized by the x-intercept $a \in \mathbb{R}$: the vertical line corresponding to $a \in \mathbb{R}$ is given by $x = a$.

(3) One can remove (or, more accurately, "hide") the difficulty with infinite slope by replacing the slope m by the *angle of inclination,* that is, the angle θ the line makes with the x-axis. These two parameters relate by:

$$m = \tan(\theta)$$

However, like all angles, θ is only defined up to multiples of π. One can restrict θ to lie in an interval, say, $0 \leq \theta < \pi$, but this line does not vary continuously as $\theta \nearrow \pi$.

(4) Lines may also be parametrized by their *closest-points* as follows. A line L contains a unique point p which is closest to the origin O. If $p \neq O$, then this point determines L uniquely. However, the case $p = O$ (that is, when $L \ni O$) has to be handled separately, like the vertical lines in the slope-intercept parametrization.

1.4. **Topology is the villain.** The problem cannot be solved easily, since it reflects a fundamental fact, involving the *topology* of the set of lines in the plane. Unlike the points in the plane, which form a tractable algebraic object (a *vector space*), the lines in the plane form a space which is inherently more complicated. "Topology" refers to how the elements of the set are organized, and even for simple familiar objects, the topologies are very subtle.

Angles are illustrative of this phenomenon. Since the set of angles "closes up" —- when you go around a full 360^o — the set cannot be identified with a set of numbers or vectors in a completely satisfactory way. One has to introduce special cases to handle exceptions, and there is no way to get around this.

Other situations, like sets of lines in the plane or 3-space, lead to even more complicated topologies. For these two cases, the set is *nonorientable,* like a Möbius band, and this is particularly difficult to coordinatize.

Projective geometry began by the efforts of Renaissance architects and artists to deal with perspective. *Projective space* enlarges our usual space by introducing new points (called *ideal points*) which is where parallel lines eventually meet. (Imagine an aerial view of railroad tracks converging in the horizon.)

Projective space also has a very complicated topology; for example, the projective plane is nonorientable. It enjoys a set of *homogeneous coordinates,* which are only unique up to scaling — and to do calculations in projective geometry one has to work only in pieces of the space which are manageable and do admit vector coordinates. The geometric calculations all reduce to matrix operations in linear algebra, but to specify an arbitrary point in two-dimensional projective space, one needs *three* coordinates.

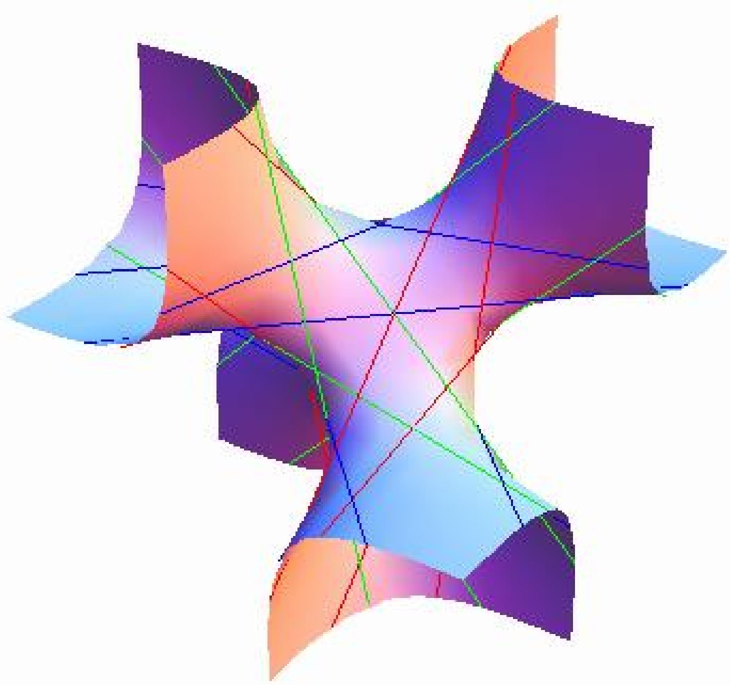
1.5. **Transformations and data types.** Certain types of transformations work better in certain coordinate systems. For example, polar coordinates behave very well under rotations, but they can be extraordinarily awkward in others. Try writing down the expression for a translation in polar coordinates and compare it to the expression in rectilinear coordinates.

Transformations preserving some special geometric properties may be suitable for special data types, which can by more succinct and more efficient. For example, angle-preserving transformations can be written very elegantly in terms of complex numbers: if $z \in \mathbb{C}$ is a complex number, then the *affine transformation* $z \longmapsto \lambda z + \tau$, where $\lambda \in \mathbb{C} \setminus \{0\}$ and $\tau \in \mathbb{C}$ is the most general orientation-preserving transformation which preserves angles. Whereas general affine transformations of the plane need six numbers, angle-preserving transformations depend only on two *complex numbers,* which is equivalent to four (real) numbers. This represents a significant improvement in the storage of data.

In 3 dimensions, rotations are more complicated, but they can be represented very elegantly using *quaternions,* the four-dimensional generalization of complex numbers. A linear rotation of $\mathbb{R}^3$ admits a very nice description in terms of quaternions, generalizing the remarkable formula

$$e^{i\theta} = \cos(\theta) + i\sin(\theta)$$

Quaternions are more complicated than complex numbers, largely due to the fact that $AB \neq BA$ in general. However, they are easily implemented in terms of standard vector operations. Discovered in 1843 by W. R. Hamilton (long before the advent of computer graphics) they are now a standard component of graphical routines implemented in hardware.



2. Euclidean algebra

We begin with an algebraic approach to Euclidean geometry. We introduce the notion of *algebraic structure,* which is what is *preserved* under the transformations we wish to study. The most important algebraic structure is the *linear* structure on a vector space, and the transformations which preserve this structure are the *linear mappings.* A key point is that, using *bases,* a linear mapping is encoded by finite information: the entries of a matrix. More general *multilinear mappings* share this important property.

2.1. Equivalence Relations and Transformation Groups.

For sets $A_1, \dots, A_k$, define the *Cartesian product* $A_1 \times \cdots \times A_k$ as the set of *ordered k-tuples*an

$$(a_1, \dots, a_k), \text{ where } a_1 \in A_1, \dots, a_k \in A_k.$$

A *(binary) relation* on a set X is a subset $R \subset X \times X$. If $(x, y) \in R$, then we say that x and y are *related* (or *R-related*) and sometimes write xRy. Here are some examples of binary relations:

- *Equality:* $R = \{(x, y) \mid x = y\}$;
- *Less than:* $R = \{(x, y) \mid x < y\}$;
- *Greater than:* $R = \{(x, y) \mid x > y\}$;
- *Less than or equals:* $R = \{(x, y) \mid x \leq y\}$;
- *Greater than or equals:* $R = \{(x, y) \mid x \geq y\}$;
- *Congruence:* Here X is the set of geometric objects (for example, triangles), and define R by $(\triangle_1, \triangle) \in R$ if and only if $\triangle_1$ is congruent to $\triangle_2$. Note that the SSS-criterion for congruent triangles means that if $\triangle_i = \triangle(A_i, B_i, C_i)$, then $\triangle_1$ is congruent to $\triangle_2$ if and only if the corresponding side lengths are equal:
$$\begin{aligned} d(A_1, B_1) &= d(A_2, B_2) \\ d(B_1, C_1) &= d(B_2, C_2) \\ d(C_1, A_1) &= d(C_2, A_2). \end{aligned}$$
- *Similarity :* Here X again is the set of geometric objects (for example, triangles), and define R by $(\triangle_1, \triangle) \in R$ if and only if $\triangle_1$ is similar to $\triangle_2$. Note that the AAA-criterion for simiilar triangles means that if $\triangle_i = \triangle(A_i, B_i, C_i)$, then $\triangle_1$ is similar to $\triangle_2$ if and only if the corresponding vertex angles are equal:
$$\begin{aligned} \angle(C_1 A_1 B_1) &= \angle(C_2 A_2 B_2) \\ \angle(A_1 B_1 C_1) &= \angle(A_2 B_2 C_2) \\ \angle(B_1 C_1 A_1) &= \angle(B_2 C_2 A_2) \end{aligned}$$

- *Parallelism:* Two lines ℓ_1, ℓ_2 are *parallel* if they don't intersect, that is, if they have the "same direction." This defines a relation $\|$ on the set $\mathfrak{L}$ of lines.

An relation $R \subset X \times X$ is an *equivalence relation* if and only if it satisfies the following three properties:

- *Reflexive:* If $x \in X$, then $(x, x) \in R$.
- *Symmetric:* If $(x, y) \in R$ then $(y, x) \in R$.
- *Transitive:* If $(x, y) \in R$ and $(y, z) \in R$, then $(x, z) \in R$.

Of the above examples, equality, congruence, similarity and parallelism are equivalence relations. The four order relations are all transitive, and "less than or equals" and "greater than or equals" are both reflexive.

Equivalence relations formalize notions of "sameness." Congruent polygons have the "same geometry," rendering the question of what is "geometry" equivalent to the question of understanding "congruence". Maybe it's better to say that congruent polygons have the same *metric structure,* that is all the corresponding measurements of distance are "the same." Similarly (no pun intended), similar polygons havethe same *angular structure,* in that corresponding measure of angles are "the same."

All of these equivalence relations share an important feature. They all arise from considerations of *symmetry.* The *symmetries* of an object are the transformations which preserve the object. One may think of rotations of a circle or a regular polygon, like a square. When you transform the object by a symmetry, it stays "the same." This leads to the important notion of a *transformation group.*

A *group* is a set G with some extra structure:

- A distinguished element $e \in G$, called its *identity element*;
- A mapping $G \xrightarrow{\iota} G$ called *inversion*;
- A mapping $G \times G \xrightarrow{\mu} G$ called the *group operation.*

A good example to consider is the collection G of nonzero real numbers, where $e = 1$, $\iota(x) = 1/x$ and $\mu(x, y) = xy$. Often the group operation is called *multiplication* and $\iota(x)$ is written x^{-1}.

They satisfy some basic axioms:

- *Associative:* If $a, b, c \in G$, then $(ab)c = a(bc)$.
- *Identity Element:* If $a \in G$, then $a = 1a = a1$.
- *Inverses:* If $a \in G$, then $aa^{-1} = a^{-1}1 = a$.

Note that we don't require *commutativity,* $ab = ba$. If the group operation is commutative, then we say that G is a *commutative* (or *Abelian*) group.

Another important algebraic structure arises from *symmetry.* The invertible transformations preserving some structure themselves form an object with algebraic structure. The corresponding *algebraic object* is called a *(transformation) group.* Suppose X is a set "with some structure." Then two structure-preserving mappings

$$X \xrightarrow{A} X, \qquad X \xrightarrow{B} X$$

can be *composed,* (that is, applied successively) and the *composition* (the effect of applying one after the other)

$$\begin{aligned} X &\xrightarrow{A \circ B} X \\ x &\longmapsto A \circ B(x) := A\big(B(x)\big) \end{aligned}$$

also preserves that structure. If the structure-preserving mapping A is *invertible,* then its inverse A^{-1} also preserves the structure. Finally the *identity mapping* $\mathbf{I}$, (the transformation which leaves everything completely fixed)

$$\begin{aligned} X &\xrightarrow{\mathbf{I}} X \\ x &\longmapsto x \end{aligned}$$

is automatically structure-preserving *for any "structure" on X.* This is the general framework for a group of transformations.

2.2. **Review of Euclidean linear algebra.** *Euclidean space* is the natural setting for geometry. Many familiar geometric ideas can be expressed in terms of *linear algebra.* Furthermore expressing them algebraically facilitates numerical computations. The corresponding algebraic structure is that of a *vector space* with an inner product. This remarkable package (which we call a *Euclidean vector space,* contains all of Euclidean geometric ideas. The ideas include distance, lines and planes, angles, area, volume, parallelism as well as the groundwork to model many physical phenomena.

Let n be a positive integer, representing the *dimension* of our world. For us, the *scalars* will be *real numbers,* and we denote the set of real numbers by $\mathbb{R}$. The algebraic structure on $\mathbb{R}$ is that of a *field:* Real numbers can be added, subtracted, and multiplied. Furthermore one can *divide,* as long as the denominator is nonzero.

Points in Eucldean n-space are represented by (ordered) n-tuples

$$\mathbf{x} := \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$$

where $x_i \in \mathbb{R}$ are the *coordinates* (or the *entries* of $\mathbf{x}$. If $c \in \mathbb{R}$ is a scalar, then the *scalar product* $c\mathbf{x} \in \mathbb{R}^n$ is defined by *coordinatewise multiplication:*

$$c\mathbf{x} := \begin{bmatrix} cx_1 \\ \vdots \\ cx_n \end{bmatrix}$$

If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, then their *vector sum* is also defined coordinatewise, that is by:

$$\mathbf{x} + \mathbf{y} := \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix}.$$

2.2.1. *Vector spaces and linear transformations.* The algebraic structure on our basic algebraic object — a *vector space* — is defined by algebraic operations abstracting scalar multiplication and vector addition on $\mathbb{R}^n$. The operations satisfy certain standard algebraic axioms such as commutativity, associativity, distributivity, existence of identity elements, and inverses.

If $W \subset V$ is a subset, then sums and scalar multiples of elements of W are defined. They live in V, but not necessarily in W. If W is *closed under* these operations, then we say that W is a *subspace* of V.

If V is a vector space, and $\mathbf{v}^1, \dots \mathbf{v}^k \in V$ are vectors, and $c_1, \dots c_k \in \mathbb{R}$ are scalars, then we can combine the two operations in V to define the *linear combination*

$$c_1\mathbf{v}^1 + \dots c_k\mathbf{v}^k \ \in \ V.$$

Clearly a subset $W \subset V$ is a subspace if and only if it is *closed under linear combinations,* that is, if whenever $\mathbf{v}^1, \dots \mathbf{v}^k \in W$, and scalars $c^1, \dots c^k \in \mathbb{R}$, then the linear combination $c_1\mathbf{v}^1 + \dots c_k\mathbf{v}^k \ \in \ W$.

Suppose V, W are two vector spaces, and $V \xrightarrow{L} W$ is a mapping. Then L is *linear* if and only if it preserves the algebraic structures of V and W, that is, whenever $\mathbf{v}, \mathbf{w} \in V$ are vectors and $c \in \mathbb{R}$ is a scalar, then:

- $L(\mathbf{v} + \mathbf{w}) = L(\mathbf{v}) + L(\mathbf{w})$;
- $L(c\mathbf{v}) = cL(\mathbf{v})$.

Equivalently, L *preserves linear combinations:* whenever $\mathbf{v}^1, \dots \mathbf{v}^k \in V$ and $c^1, \dots c^k \in \mathbb{R}$, then

$$L(c_1\mathbf{v}^1 + \dots c_k\mathbf{v}^k) = c_1 L(\mathbf{v}^1) + \dots c_k L(\mathbf{v}^k).$$

Here is an example of a linear mapping. Let $\mathbf{v}^1, \dots \mathbf{v}^k \in V$. Write $\vec{\mathbf{v}}$ for the k-tuple $\big(\mathbf{v}^1, \dots \mathbf{v}^k\big) \in V \times \cdots \times V$. Then the mapping

$$\mathbb{R}^k \xrightarrow{L_{\vec{\mathbf{v}}}} V$$
$$\begin{bmatrix} c^1 \\ \vdots \\ c^k \end{bmatrix} \longmapsto c_1\mathbf{v}^1 + \dots c_k\mathbf{v}^k$$

is linear.

A linear mapping $V \longrightarrow \mathbb{R}$ is called a *linear functional* or a *covector.* Just as we represent vectors in $\mathbb{R}^n$ by *column vectors,* covectors are represented by *row vectors:* the row vector $\begin{bmatrix} a_1 & \dots & a_n \end{bmatrix}$ corresponds to the linear functional

$$\mathbb{R}^n \longrightarrow \mathbb{R}$$
$$\begin{bmatrix} \mathbf{v}^1 \\ \vdots \\ \mathbf{v}^n \end{bmatrix} \longmapsto a_1\mathbf{v}^1 + \dots a_n\mathbf{v}^n.$$

In particular for each$i = 1, \dots, n$, the mapping which assigns to a vector in $\mathbb{R}^n$ its i-th entry (or *i-th coordinate*) is a linear functional, which we call the *i-th basic covector.*

2.2.2. *Bases.* If $X \subset V$ is a subset, then the set of linear combinations of elements of X forms a subspace W, called the *span* of X. We say that X *spans* W and we write $W = \mathsf{span}(X)$.

The *dual* or *opposite* notion is that of *linear independence.* A subset $X \subset V$ is *linearly dependent* if and only if there is a nonzero linear relation in X, that is, some linear combination

$$c_1\mathbf{v}^1 + \dots c_k\mathbf{v}^k = 0$$

for $\mathbf{v}^1, \dots, \mathbf{v}^k \in X$ and not all the c_i are zero. Otherwise we say that the subset X is *linearly independent.*

Theorem 2.2.1. *Let V be a vector space and $X \subset V$. The following are conditions are equivalent:*

- *X spans V and is linearly independent.*
- *X is a maximal linearly independent subset.*
- *X is a minimal spanning subset.*
- *Every element $\mathbf{v} \in V$ is a linear combination $c_1\mathbf{x}^1 + \dots c_n\mathbf{x}^k$ where $X = \{\mathbf{x}^1, \dots, \mathbf{x}^n\}$ and $(c^1, \dots, c^n) \in \mathbb{R}^n$ is unique.*

If any of these equivalent conditions hold, then we say that X is a *basis* of V (or X *bases* V). If V admits a finite basis of cardinality n, then every basis of V has cardinality n. The number n is called the *dimension* of V and we write $n = \dim(V)$.

Suppose that V, W are vector spaces of respective dimensions $m = \dim(V), n = \dim(W)$ and

$$\big(\mathbf{v}^1, \dots \mathbf{v}^m\big), \quad \big(\mathbf{w}^1, \dots \mathbf{w}^n\big)$$

base V and W respectively. Suppose that $V \xrightarrow{L} W$ is a linear transformation. Then for each $i = 1, \dots, m$, the vector $L(\mathbf{v}^i) \in W$ is uniquely a linear combination of the $\mathbf{w}^1, \dots, \mathbf{w}^m$:

$$L(\mathbf{v}^i) = \sum_{j=1}^{m} L^i_j \mathbf{w}^j$$

for scalars $L^i_j \in \mathbb{R}$. The array

$$[L] := \big(L^i_j\big)_{1 \le i \le n, 1 \le j \le m}$$

is an $n \times m$*-matrix,* which competely determines L. Furthermore every such $n \times m$-matrix defines a linear transformation $V \xrightarrow{L} W$.

In particular, using bases, linear transformations are determined by a finite set of information, namely the nm scalars L^i_j, the *matrix entries* of L.

We denote the space of $n \times m$-matrices by $\mathsf{Mat}_{n,m}$ and, when $m = n$, simply by Mat_n.

2.2.3. *Mulilinear mappings.* Let $V_1, \dots, V_k, W$ be vector spaces. A mapping

$$V_1 \times \cdots \times V_k \xrightarrow{F} W$$

is *multilinear* if for each $i = 1, \dots, k$ and $v_j \in V_j$ for $j \neq i$, the mapping

$$\begin{aligned} V_i &\longrightarrow W \\ v &\longmapsto F(v_1, \dots, v_{i-1}, v, v_{i+1}, \dots, v_k) \end{aligned}$$

is linear. That is, F is linear *in each variable separately.*

Multilinear mappings abound. If all the V_i are identical, then a multilinear map as above is called k*-linear.* A 2-linear map is called *bilinear,* a 3-linear map is called *trilinear,* and a 4-linear map is called *quadrilinear.*

Example 2.2.2. *The* inner product *or* dot product *in* $\mathbb{R}^n$ *is defined as:*

$$\begin{aligned}\mathbf{v}\cdot\mathbf{w} &:= v^1w^1 + v^2w^2 + \cdots + v^nw^n \\ &= \mathbf{v}^\dagger\mathbf{w} = \begin{bmatrix}\mathbf{v}^1 & \mathbf{v}^2 & \dots & \mathbf{v}^n\end{bmatrix}\begin{bmatrix}\mathbf{w}^1\\ \mathbf{w}^2\\ \vdots\\ \mathbf{w}^n\end{bmatrix}\end{aligned}$$

Geometrically, $\mathbf{v}\cdot\mathbf{v} = \|\mathbf{v}\|^2$ *where*

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n}\left(v^i\right)^2}$$

denotes the length *of the vector* $\mathbf{v}$*. If* p, q *are points (represented by* vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n$*, then the* distance $d(p,q)$ *between them equals the length* $\|\mathbf{v}\|$ *where* $\mathbf{v} = \mathbf{p} - \mathbf{q}$*. Intuitively* $d(p,q)$ *represents how much work it takes to "move"* p *to* q*. "Moving* p *to* q*" is effected by applying the translation from* p *to* q*which corresponds to the vector* $\mathbf{v}$*:*

$$q = p + \mathbf{v}$$

Inner products also determine angles: $\mathbf{v}, \mathbf{w}$ *are unit vectors (that is,* $\|\mathbf{v}\| = \|\mathbf{w}\| = 1$*), then their angle* $\angle(\mathbf{v},\mathbf{w})$ *satisfies:*

$$\cos\big(\angle(\mathbf{v},\mathbf{w})\big) = \mathbf{v}\cdot\mathbf{w}$$

In particular two vectors are orthogonal *if and only if* $\mathbf{w}\cdot\mathbf{v} = 0$*, that is, if* $\angle(\mathbf{v},\mathbf{w})\big) = \pi/2 = 90^o$*.*

Example 2.2.3. *The* two-dimensional determinant *in* $\mathbb{R}^2$ *is defined as:*

$$\mathbf{v}\wedge\mathbf{w} := \mathsf{Det}\begin{bmatrix}\mathbf{v}^1 & \mathbf{w}^1\\ \mathbf{v}^2 & \mathbf{w}^2\end{bmatrix} = \begin{vmatrix}\mathbf{v}^1 & \mathbf{w}^1\\ \mathbf{v}^2 & \mathbf{w}^2\end{vmatrix} = v^1w^2 - v^2w^1$$

Geometrically, $\mathbf{v}\wedge\mathbf{w}$ *represents the* oriented area *of the parallelogram defined by* $\mathbf{v}$ *and* $\mathbf{w}$*.*

Example 2.2.4. *The* vector product *or* cross-product *is special to three dimensions; it is a bilinear map*

$$\begin{aligned}\mathbb{R}^3\times\mathbb{R}^3 &\xrightarrow{\times} \mathbb{R}^3\\ \left(\begin{bmatrix}\mathbf{v}^1\\ \mathbf{v}^2\\ \mathbf{v}^3\end{bmatrix}, \begin{bmatrix}\mathbf{w}^1\\ \mathbf{w}^2\\ \mathbf{w}^3\end{bmatrix}\right) &\longmapsto \begin{vmatrix}\hat\imath & \mathbf{v}^1 & \mathbf{w}^1\\ \hat\jmath & \mathbf{v}^2 & \mathbf{w}^2\\ \hat k & \mathbf{v}^3 & \mathbf{w}^3\end{vmatrix} = \begin{bmatrix}\mathbf{v}^2\mathbf{w}^3 - \mathbf{w}^2\mathbf{v}^3\\ \mathbf{w}^1\mathbf{w}^3 - \mathbf{v}^3\mathbf{v}^1\\ \mathbf{v}^1\mathbf{w}^2 - \mathbf{w}^1\mathbf{v}^2\end{bmatrix}\end{aligned}$$

Example 2.2.5. *The* three-dimensional determinant *is a* trilinear mapping:

$$\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \xrightarrow{\mathsf{Det}} \mathbb{R}$$
$$\left(\begin{bmatrix} \mathbf{v}^1 \\ \mathbf{v}^2 \\ \mathbf{v}^3 \end{bmatrix}, \begin{bmatrix} \mathbf{w}^1 \\ \mathbf{w}^2 \\ \mathbf{w}^3 \end{bmatrix}, \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \mathbf{u}^3 \end{bmatrix} \right) \longmapsto \mathbf{v}^1\mathbf{w}^2\mathbf{u}^3 + \mathbf{v}^2\mathbf{w}^3\mathbf{u}^1 + \mathbf{v}^3\mathbf{w}^1\mathbf{u}^2$$
$$- \mathbf{v}^2\mathbf{w}^1\mathbf{u}^3 - \mathbf{v}^3\mathbf{w}^2\mathbf{u}^1 - \mathbf{v}^1\mathbf{w}^3\mathbf{u}^2.$$

Geometrically $\mathsf{Det}(\mathbf{v}, \mathbf{w}, \mathbf{u})$ represents the *oriented volume* of the *oriented parallelipiped* determined by $(\mathbf{v}, \mathbf{w}, \mathbf{u})$. In particular the sign of the determinant (positive or negative) corresponds to *orientation* in $\mathbb{R}^3$.

An important property of Det is that it is *alternating:* namely, if you interchange two of the arguments you end up picking up a sign -1, that is,

$$\mathsf{Det}(\mathbf{v}, \mathbf{w}, \mathbf{u}) = -\mathsf{Det}(\mathbf{w}, \mathbf{v}, \mathbf{u}) = -\mathsf{Det}(\mathbf{v}, \mathbf{u}, \mathbf{w}) = -\mathsf{Det}(\mathbf{u}, \mathbf{w}, \mathbf{v}).$$

By repeating two of these operations, such an alternating map is invariant under *cyclic permutations:*

$$\mathsf{Det}(\mathbf{v}, \mathbf{w}, \mathbf{u}) = \mathsf{Det}(\mathbf{w}, \mathbf{u}, \mathbf{v}) = \mathsf{Det}(\mathbf{u}, \mathbf{v}, \mathbf{w})$$

Given a trilinear mapping as above, the alternating property is equivalent to:

$$\mathsf{Det}(\mathbf{v}, \mathbf{v}, \mathbf{w}) = \mathsf{Det}(\mathbf{v}, \mathbf{w}, \mathbf{v}) = \mathsf{Det}(\mathbf{v}, \mathbf{w}, \mathbf{v})$$

Bases are useful in representing multilinear mappings by multidimensional *arrays* — finite sets of numbers. In its simplest case, a bilinear mapping $V \times V \xrightarrow{\mathcal{B}} \mathbb{R}$ corresponds to an $n \times n$-matrix B (where $n = \dim(V)$) by defining

$$B_{ij} := \mathcal{B}(\mathbf{v}_i, \mathbf{v}_j),$$

where $(\mathbf{v}_1, \dots, \mathbf{v}_n)$ bases V. Then, if $\mathbf{u}, \mathbf{w} \in V$ are vectors with coordinates $\mathbf{u}^1, \dots, \mathbf{u}^n$ and $\mathbf{w}^1, \dots, \mathbf{w}^n$ respectively,

$$\begin{aligned} \mathcal{B}(\mathbf{u}, \mathbf{w}) &= \mathcal{B}(\mathbf{u}^1\mathbf{v}_1 + \dots + \mathbf{u}^n\mathbf{v}_n, \mathbf{w}^1\mathbf{v}_1 + \dots + \mathbf{w}^n\mathbf{v}_n) \\ &= \sum_{i,j=1}^{n} \mathbf{u}^i B_{ij} \mathbf{w}^j \end{aligned}$$

by bilinearity. Clearly general multilinear mappings also enjoy this basic property.

2.2.4. *Duality and covectors.* The set of linear mappings $V \longrightarrow \mathbb{R}$, or *linear functionals* or *covectors* forms a vector space, called the vector space *dual* to V and denoted V^*. If V corresponds to column vectors, then V^* is the represented by *row vectors.* Namely, the row vector $\psi = \begin{bmatrix}\psi^1 & \dots & \psi^n\end{bmatrix}$ acts on column vectors in V by matrix multiplication:

$$\begin{aligned} V &\xrightarrow{\psi} \mathbb{R} \\ \begin{bmatrix}\mathbf{w}^1 \\ \vdots \\ \mathbf{w}^n\end{bmatrix} &\longmapsto \begin{bmatrix}\psi^1 & \dots & \psi^n\end{bmatrix} \begin{bmatrix}\mathbf{w}^1 \\ \vdots \\ \mathbf{w}^n\end{bmatrix} \\ &= \psi^1\mathbf{w}^1 + \dots + \psi^n\mathbf{w}^n \end{aligned}$$

Alternatively, replace this matrix product with the *inner product:*

$$\psi\mathbf{w} = \psi^\dagger \cdot \mathbf{w}$$

where the column vector $\psi^\dagger$ is the *transpose* of the row vector ψ.

Another way of saying this is that the inner product defines an *isomorphism* of V with V^*:

$$V \xrightarrow{\mathcal{I}} V^*$$

defined by:

$$\begin{aligned} V &\xrightarrow{\mathcal{I}(\mathbf{v})} \mathbb{R} \\ \mathbf{w} &\longmapsto \mathbf{v}^\dagger\mathbf{w} = \mathbf{v} \cdot \mathbf{w} \end{aligned}$$

which is described in terms of the matrix operation of *transpose.*

2.3. **Euclidean geometry.** We are interested in geometric transformations which preserve some geometric structure, such as distance, linearity, parellelism, angle, area, or volume. We also want to be able to calculate geometric quantities, and use the powerful tools of linear algebra to compute. This requires coordinates, and Euclidean coordinates naturally arise from transformations called *translations.*

We want our coordinates to be as natural and robust as possible. We want to be *fair,* and treat all the points on an equal footing (before making assumptions according to the specific problems). If p, q are points, then the *vector between them*

$$\mathbf{v} = \mathbf{q} - \mathbf{p} \in \mathbb{R}^n$$

determines a translation which "moves p to q, in the sense that the transformation

$$\begin{aligned} \mathbb{R}^n &\xrightarrow{\tau^{\mathbf{v}}} \mathbb{R}^n \\ \mathbf{x} &\longmapsto \mathbf{x} + \mathbf{v} \end{aligned}$$

"parallel translates" the point p to the point q. We can say that two objects $A, B \subset \mathbb{R}^n$ are *parallel* if there is some vector $\mathbf{v}$ such $\tau^{\mathbf{v}}(A) = B$.

2.3.1. *Isometries and orthogonal matrices.* Translations are special examples of *isometries,* those transformations which preserve distance. Explicitly, a transformation f is an isometry if and only if for all points p, q,

$$d\big(f(p), f(q)\big) = d(p, q).$$

Whenever some "structure" (like "distance") is preserved by transformations A and B, then their composition

$$A \circ B : x \xmapsto{A \circ B} A\big(B(x)\big)$$

also preserves this structure. Thus the composition of two isometries is an isometry. Similarly the inverse of an isometry is an isometry and the identity transformation is an isometry. Thus isometries form a
em transformation group.

3. Complex numbers

In this section we discuss the complex numbers, which correspond to vectors in $\mathbb{R}^2$, but with an extra algebraic structure, given by *multiplication.* This structure has many useful and important features, and is particular to dimension two.

3.1. **The field of complex numbers.** Complex numbers form a *field,* a set F with algebraic structure defined by two inter-related operations. Both operations (called *addition* and *multiplication*) are commutative, associative and possess identity elements. The additive identity element is called *zero* and denoted 0. The multiplicatitive identity element is called *one* and denoted 1. Under addition, F is a commutative group. Multiplication relates to addition by *distributivity:*

$$a(b + c) = ab + ac$$

In other words, the mapping

$$F \xrightarrow{M_a} F$$
$$x \longmapsto ax$$

is linear when $a \in F$. One consequence is that $0x = 0$ for all $x \in F$:

$$x + 0x = (1 + 0)x = 1x = x$$

so $0x = 0$. In particular 0 cannot have a multiplicative inverse.

Finally, we assume that every nonzero element of F admits a multiplicative inverse, that is, the set of *nonzero* elements of F forms a group. The set of rational numbers forms a field, denoted $\mathcal{Q}$, and the set of real numbers forms a field denoted $\mathbb{R}$. Denote the field of complex numbers by $\mathbb{C}$.

Exercise 3.1.1. *Let F be a field. Denote the additive inverse of 1 by -1. Prove that $(-1)x = -x$ is the additive inverse of x for any $x \in F$.*

3.2. **Euclidean geometry and complex numbers.** The group of *orientation-preserving Euclidean similiarities* identifies with the group of *complex affine transformations*

$$\begin{aligned} \mathbb{C} &\xrightarrow{\phi} \mathbb{C} \\ z &\longmapsto az + b \end{aligned}$$

where $a, b \in \mathbb{C}$ and $a \neq 0$. Here $+b$ corresponds to *translation* by the *vector* b and multiplication $z \longmapsto az$ corresponds to the composition of *scaling* by $r > 0$ and *rotation* by θ where:

$$r = |a|, \qquad e^{i\theta} = a/|a|.$$

Reflection in a line reverses orientation, so it is *not* represented by a complex-affine transformation as above. Rather, it is the composition of a complex-affine transformation with complex conjugation (reflection in the *real axis*), that is, a transformation of the form

$$z \longmapsto a\bar{z} + b$$

where $|a| = 1$ and $b \in \mathbb{C}$ is arbitrary.

Exercise 3.2.1. *The transformation $z \stackrel{\phi}{\longmapsto} a\bar{z} + b$ of $\mathbb{C}$ is* not *complex-affine, but under the usual identification $\mathbb{C} \leftrightarrow \mathbb{R}^2 = \mathbb{A}^2$, it is* affine *over $\mathbb{R}$.*

- *Compute $\overline{\phi(\bar{z})}$.*
- *What is the line fixed by the rotation above?*
- *What kind of transformation arises if $|a| \neq 1$? Does it preserve or reverse orientation?*

3.2.1. *The extended complex line.* The *complex projective line* is a sphere, obtained from the *complex affine line* (just $\mathbb{C}$ with a forgotten origini) by adding a single ideal point ∞. Specifically, consider homogeneous coordinates $\big[z_1 : z_2\big]$ where not both z_1, z_2 are zero, up to (complex) projective equivalence. In the affine patch where $z_2 \neq 0$, choose the affine coordinate

$$z := z_1/z_2,$$

where the ideal point

$$\infty \leftrightarrow \big[\zeta : 0\big] = \big[1 : 0\big]$$

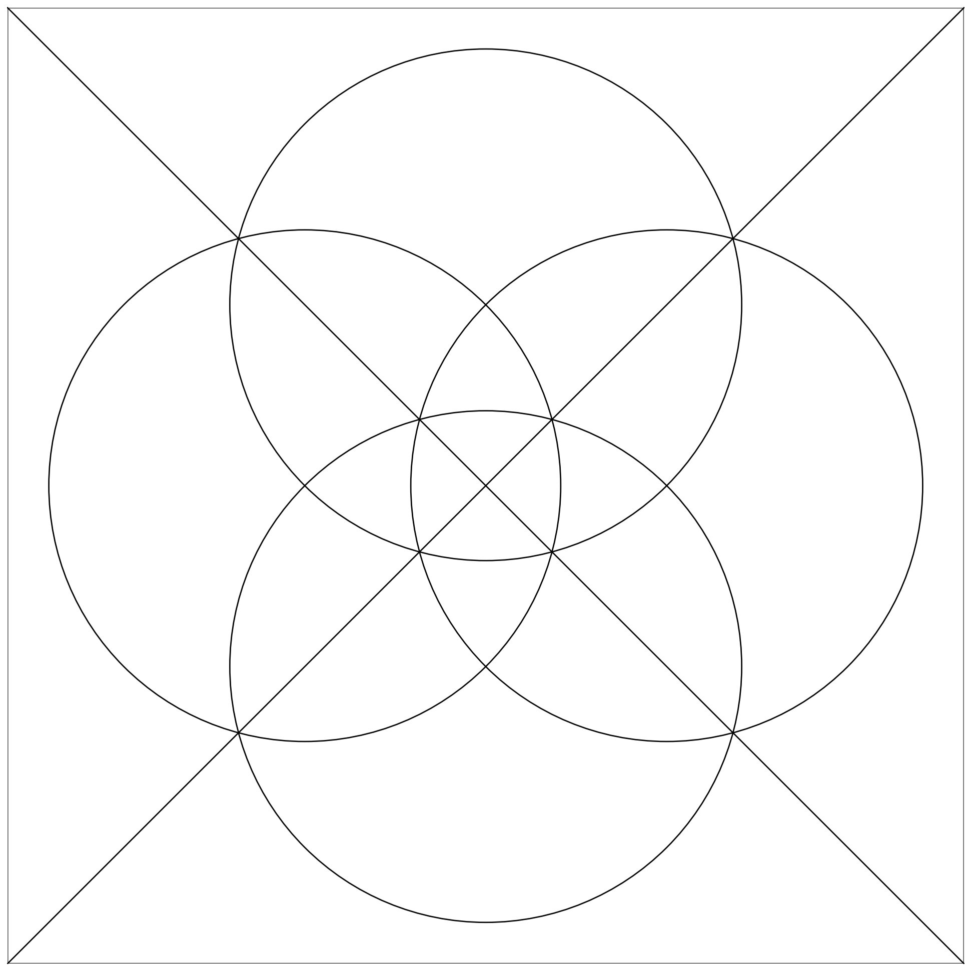


FIGURE 1. Circles in a tetrahedral configuration

for any $\zeta \neq 0$.

Like the the real case, the group of collineations is extremely interesting and complicated. Indeed, it's much more complicated than the group of real linear fractional transformations. We will only consider a few special cases.

Complex-Inversion is the transformation of $\mathbb{CP}^1$ defined by

$$z \xmapsto{\iota} 1/z$$

or $[Z_1 : Z_2] \longmapsto [Z_2 : Z_1]$ in homogeneous coordinates.

Exercise 3.2.2. *Let ℓ be the line $1+i\mathbb{R}$ whose closest-point to the origin is 1. Show that $\iota(\ell)$ is the circle centered at $1/2$ of radius $1/2$, that is the circle passing through 0 and 1 and orthogonal to the horizontal axis*

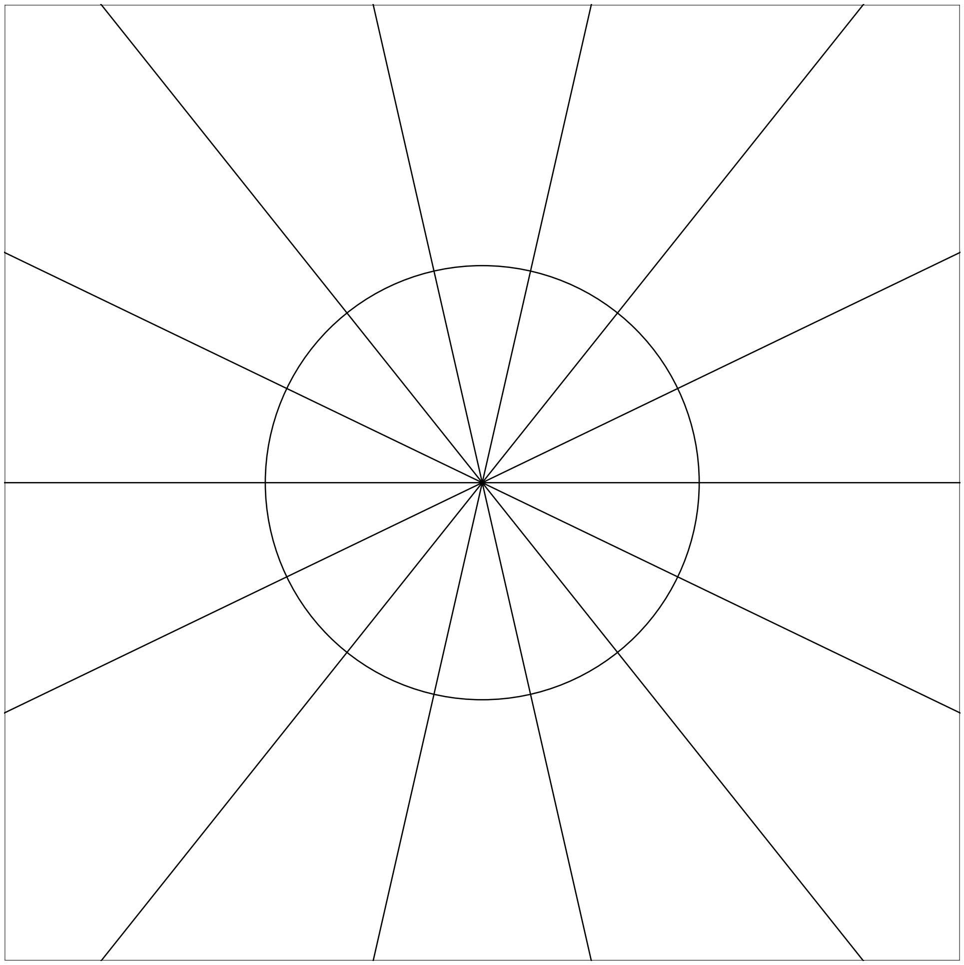


FIGURE 2. Seven circles in a dihedral configuration

$\mathbb{R} \subset \mathbb{C}$. *This is the circle defined by the equation*

$$\left| z - \frac{1}{2} \right| = \frac{1}{2}.$$

Thus the natural objects in *complex projective one-dimensional geometry* are *circles,* where a circle which passes through ∞ is defined to be a Euclidean straight line.

Closely related to complex-inversion is *inversion* in a circle; perhaps we should call this operation *reflection* in a circle. Taking our basic object as the real axis $\mathbb{R}$, reflection in $\mathbb{R}$ is just complex-conjugation $z \longmapsto \overline{z}$ or $z \longmapsto \bar{z}$ (first overline, second bar).

3.3. **Lines in terms of complex numbers.**

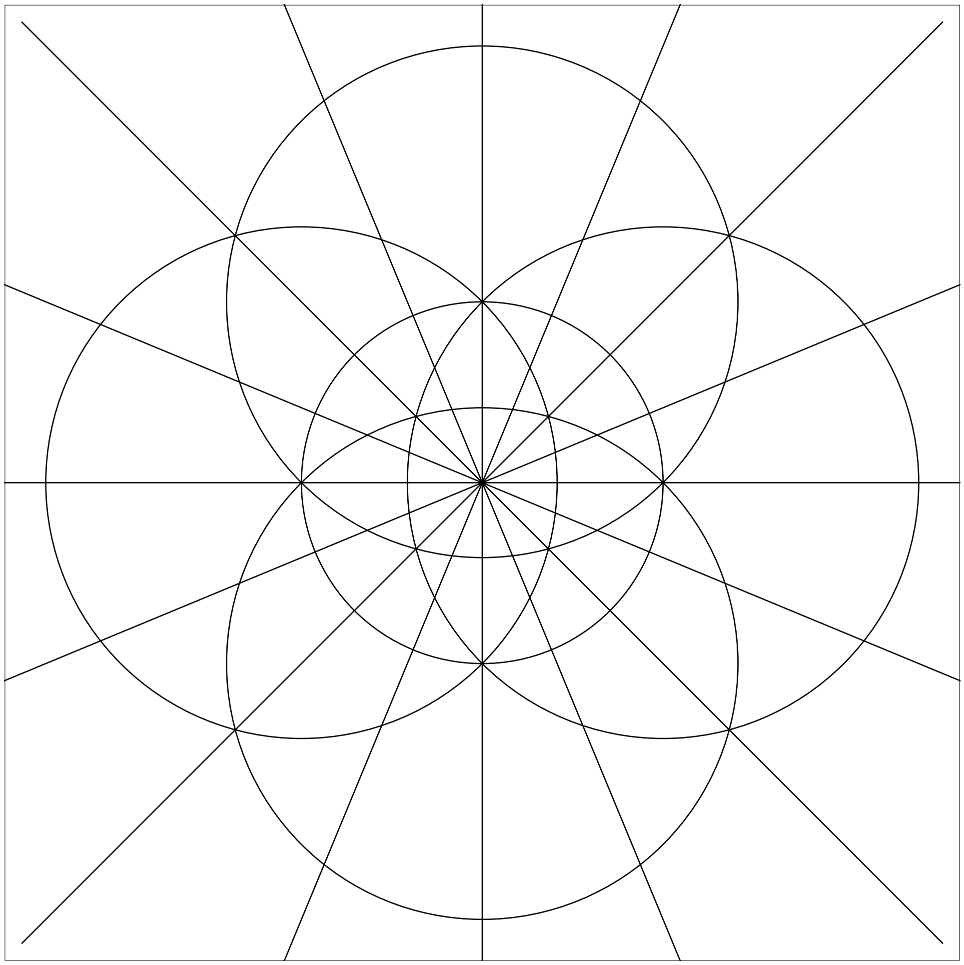


FIGURE 3. Eight circles in a octahedral configuration

3.3.1. *Closest points to a planar line.* A convenient way to parametrize Lines in the Euclidean plane $\mathbb{E}^2$ can be conveniently parametrized by their closest point to a fixed origin. By a "fixed origin" I mean an arbitrary point which we choose once and for all to make calculations. (We apply translations to move between different points.) Once we choose the origin, points in $\mathbb{E}^2$ correspond to vectors in $\mathbb{R}^2$, and the origin in $\mathbb{E}^2$ corresponds to the zero vector $\mathbf{0} \in \mathbb{R}^2$.

The notion of "closest" refers to distance, so the natural context is *Euclidean geometry.* Thus it is easier to see how this parametrization behaves with respect to Euclidean isometries and similarities.

Theorem 3.3.1. *Let $\ell \subset \mathbb{E}^2$ be a line. There exists a unique point on ℓ closest to the origin $\mathbf{0}$.*

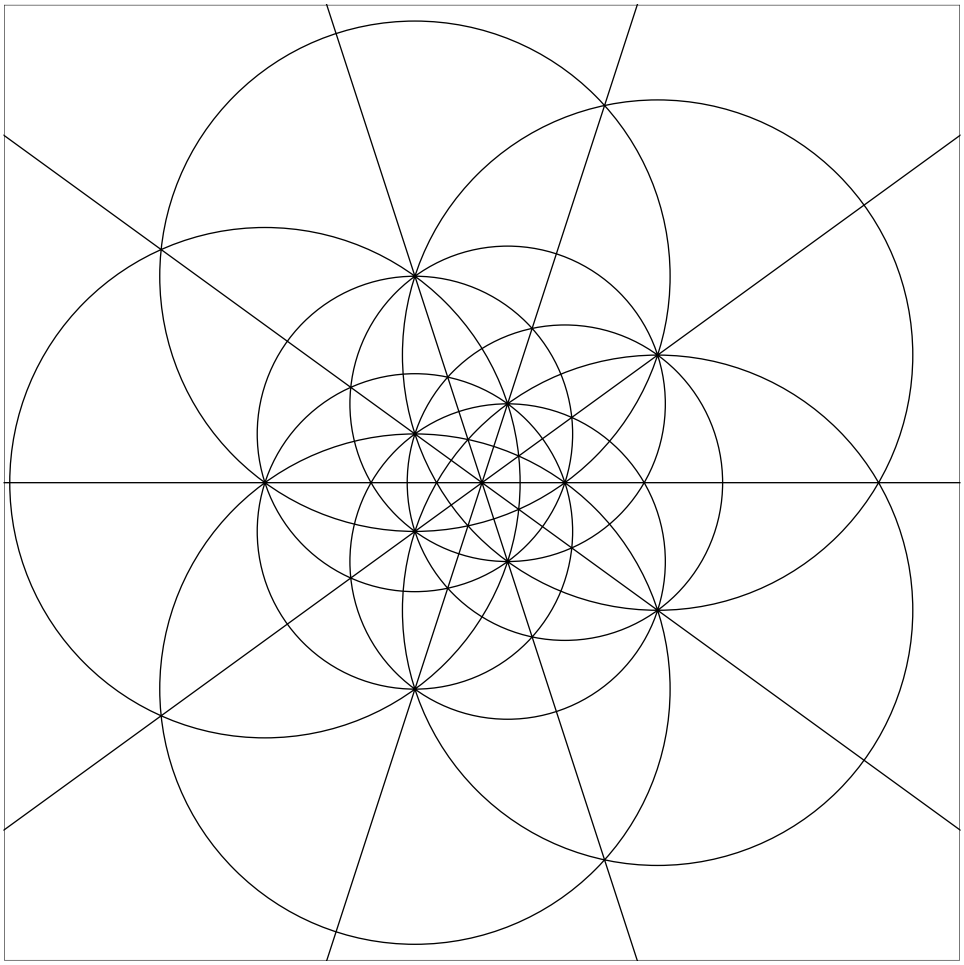


FIGURE 4. Twenty circles in a icosahdral configuration

Proof. Suppose that $p_0 \in \mathbb{E}^2$ is a point on ℓ and $\mathbf{v} \in \mathbb{R}^2$ is a vector parallel to ℓ. Then $\mathbf{v}$ specifies the direction of ℓ and $\mathbf{v} \neq \mathbf{0}$. Then a general point on ℓ is given in parametric form by:

$$p(t) := p_0 + t\mathbf{v}$$

for $t \in \mathbb{R}$. The distance $d\big(p(t), \mathbf{0}\big)$ is minimized exactly when its square $d\big(p(t), \mathbf{0}\big)^2$ is minimized. Since

$$d\big(p(t), \mathbf{0}\big)^2 = \|p(t)\|^2 = p(t) \cdot p(t),$$

$d\big(p(t), \mathbf{0}\big)^2$ will be easier to work with. Let's call this function $f(t)$.

Now

$$f(t) = d\big(p(t), \mathbf{0}\big)^2 \;=\; (p_0 + t\mathbf{v}) \cdot (p_0 + t\mathbf{v}) \;=\; a + bt + ct^2$$

where

$$\begin{aligned} a &:= p_0 \cdot p_0 \\ b &:= 2(p_0 \cdot \mathbf{v}) \\ c &= \mathbf{v} \cdot \mathbf{v} \end{aligned}$$

Since $\mathbf{v} \neq \mathbf{0}$, the quadratic term $c > 0$ and the graph of $f(t)$ is an upward pointing parabola.

The function $f(t)$ has a unique minimum at $t_0 = -2c/b$ (solve for $f'(t) = 0$), as desired. □

Exercise 3.3.2. *Prove that the vector from* $\mathbf{0}$ *to the closest point* $p(t_0)$ *is perpendicular to the vector* $\mathbf{v}$ *specifying the direction of* ℓ.

Exercise 3.3.3. *Show that this proof directly generalizes to lines and planes in* $\mathbb{E}^3$.

3.4. **Lines in terms of complex numbers.** The group $\mathsf{Isom}^+(\mathbb{E}^2)$ of orientation-preserving isometries consists of translations and rotations. They are easily represented by operations on complex numbers. Thus we shall identify $\mathbb{E}^2$ with its fixed origin with the set of complex numbers, and the origin corresponds to $\mathbf{0} \in \mathbb{C}$. Vectors in $\mathbb{R}^2$ correspond to complex numbers $\zeta \in \mathbb{C}$, and translation by ζ corresponds to complex addition:

$$\begin{aligned} \mathbb{C} &\xrightarrow{\tau_\zeta} \mathbb{C} \\ z &\longmapsto z + \zeta \end{aligned}$$

Rotation ρ_θ through angle θ corresponds to complex multiplication:

$$e^{i\theta} = \cos(\theta) + i\sin(\theta),$$

that is,

$$\begin{aligned} \mathbb{C} &\xrightarrow{\rho_\theta} \mathbb{C} \\ z &\longmapsto e^{i\theta} z \end{aligned}$$

which corresponds to the matrix

$$\exp \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Then a line through a point $z_0 \in \mathbb{C}$ in the direction of $\zeta \in \mathbb{C}$ will be given by:

$$\ell = z_0 + \zeta\mathbb{R}$$

In many cases, taking ζ to have unit length (that is, $|\zeta| = 1$) is useful, and we may write $\zeta = e^{i\theta} = \cos(\theta) + i\sin(\theta)$.

3.4.1. *Lines not containing the origin.* Suppose that ℓ does not contain $\mathbf{0}$. Let z_0 be the point on ℓ closest to the origin. Then z_0 determines ℓ uniquely:

Exercise 3.4.1. *If $\ell \not\ni \mathbf{0}$, and z_0 is the closest point on ℓ to $\mathbf{0}$, then*

$$\ell = z_0 + iz_0\mathbb{R}.$$

I prefer using complex numbers here, since the operation of 90^o-rotation is conveniently expressed by multiplication by i. In the above formula, iz_0 can be replaced by any complex number orthogonal to z_0, but we don't have a "standard" vector operation in $\mathbb{R}^2$ which effects this. The operation corresponding to multiplication by i is the linear transformation of $\mathbb{R}^2$ defined by the matrix

$$\mathbf{J} := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

3.4.2. *Lines through the origin: Angles.* Lines ℓ through the origin are conveniently parametrized by their inclination angles $\theta \in \mathbb{R}$. In terms of complex numbers:

$$\ell = \ell_\theta := e^{i\theta}\mathbb{R}.$$

It is an immutable fact of life that such lines can't be parametrized as nicely as points on a line, plane, or 3-space. The problem is:

Exercise 3.4.2. *$\ell_\theta = \ell_{\theta+\pi}$, and in fact, $\ell_\theta = \ell_{\theta+n\pi}$ for every integer θ.*

We could fix θ to lie in some interval such as $[0, \pi)$, but then as $\theta \nearrow \pi$, the line $\ell_\theta \to \ell_0$, and the dependence is *not continuous*. (Another possible choice might be $(-\pi/2, \pi/2]$, but the same problem arises when $\theta \searrow -\pi/2$.) Strictly speaking, we should take θ to be a *residue class modulo* π. In other words, numbers $\theta, \phi \in \mathbb{R}$ define the same angle if and only if their difference $\theta - \phi$ is an integer multiple of π ("θ and ϕ are *congruent modulo* π").

Exercise 3.4.3. *Using the notation above for lines through the origin, rotate ℓ_θ by angle ϕ, and derive a formula for $\rho_\phi(\ell_\theta)$.*

Reflections in lines reverse orientation but preserve distance. Recall that the reflection R_ϕ in ℓ_ϕ is:

$$\begin{aligned} \mathbb{C} &\xrightarrow{R_\phi} \mathbb{C} \\ z &\longmapsto e^{2i\phi}\,\overline{z} \end{aligned}$$

Exercise 3.4.4. *Show that $R_\phi(\ell_\theta) = \ell_{2\phi-\theta}$.*

Exercise 3.4.5. *Let $\zeta \in \mathbb{C}$ define a translation τ_ζ and ℓ_θ be a line through the origin. When does $\tau_\zeta(\ell_\theta) \ni \mathbf{0}$? In that case, express $\tau_\zeta(\ell_\theta)$ as ℓ_ϕ for some angle ϕ.*

3.4.3. *Translations.* Since a translation τ takes ℓ to a line parallel to ℓ, the direction parameter iz_0 in $z_0 + iz_0\mathbb{R}$ is essentially unchanged. Finding the closest-point z_0 on $\tau_\zeta(\ell_\theta)$ is more difficult, but I think you can do it:

Exercise 3.4.6. *Show that*

$$\tau_\zeta(\ell_\theta) = z_0 + iz_0\mathbb{R}$$

where

$$z_0 = \zeta - \mathsf{Re}(\zeta e^{-i\theta})e^{i\theta}.$$

Suppose that $\ell \not\ni \mathbf{0}$ (so $\ell = z_0 + iz_0\mathbb{R}$ for $z_0 \neq 0$). Then I computed, assuming that $\tau_\zeta(\ell) \not\ni \mathbf{0}$, that the closest point on $\tau_\zeta(\ell)$ to the origin equals:

$$\zeta - i\frac{\mathsf{Im}(\zeta\overline{z_0})}{|z_0|^2}$$

Exercise 3.4.7. *Check my computation.*

That leaves only one case: when $\ell \not\ni \mathbf{0}$ but $\tau(\ell) \ni \mathbf{0}$. Let's do it.

Exercise 3.4.8. *Since $\ell \not\ni \mathbf{0}$, the line $\ell = z_0 + iz_0\mathbb{R}$ for some $z_0 \neq 0$. Suppose $\zeta \in \mathbb{C}$ determines a translation τ_ζ such that $\tau_\zeta(\ell) \ni \mathbf{0}$. Compute $\tau_\zeta(\ell)$.*

3.4.4. *Scalings.* Although the *scaling*

$$\begin{aligned} \mathbb{C} &\xrightarrow{S_r} \mathbb{C} \\ z &\longmapsto rz \end{aligned}$$

is not isometric, the closest-point parametrization behaves well with respect to it:

Exercise 3.4.9. *Compute the parametrizations of $S_r(\ell_\theta)$ and $S_r\big(z_0 + iz_0\mathbb{R}\big)$ where $z_0 \neq 0$. What happens when r is allowed to be a complex number?*

3.5. **Old stuff.** Let $z_0 \in \mathbb{C}$. Suppose that $z_0 \neq 0$. Then the line $\ell(z_0) \subset \mathbb{C}$ whose closest point to the origin 0 is z_0 equals

$$\ell(z_0) := z_0 + (iz_0)\mathbb{R} \ \subset \mathbb{C}$$

(You can write this out in real coordinates if you prefer.)

Consider an affine transformation of the plane

$$\begin{aligned} \mathbb{R}^2 &\xrightarrow{T} \mathbb{R}^2 \\ \mathbf{v} &\longmapsto M\mathbf{v} + \mathbf{b} \end{aligned}$$

where $\mathbf{v}$ is the vector

$$\mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2$$

corresponding to the point $p := (x, y) \in \mathbb{E}^2$, the linear part $M \in \mathsf{Mat}_2^2(\mathbb{R})$ is a 2×2-matrix, and the translational part $\mathbf{b} \in \mathbb{R}^2$. Find the closest-point parameter for $T\big(\ell(z_0)\big)$, assuming that $0 \notin T\big(\ell(z_0)\big)$.

Extend this calculation to the case of lines containing the origin:

- When ℓ passes through the origin, $\ell = e^{i\theta}\mathbb{R}$ for a unique $0 \leq \theta < \pi$. Determine $T(\ell)$.
- Determine the parameter θ for $T\big(\ell(z_0)\big)$ when $0 \in T\big(\ell(z_0)\big)$.

How does this calculation simplify when the linear part M is a complex 1×1 matrix, that is,

$$M := \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \longleftrightarrow a + bi$$

Think about the best way to organize this into a computer program so that other programmers can use this. For what applications would it be enough to consider the case that M is complex as above?

Goldman: Department of Mathematics, University of Maryland, College Park, MD 20742 USA