Chapter 6

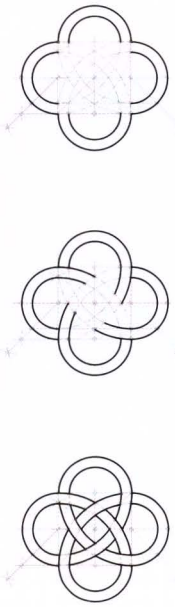
Geometry

Geometry is a very big topic. You can spend a lifetime learning it and master only a tiny fraction of the literature. It uses many mathematical concepts and formalisms, each of which takes time and effort to learn *before* you can begin to master the geometric ideas that use such concepts. Yet most of the objects made with parametric modeling systems are geometric. How can a designer ever learn anything like "enough"?

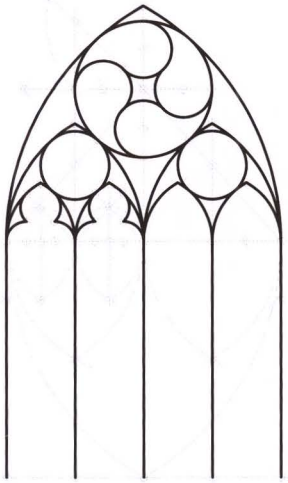
History shows that designers have always "learned enough" geometry in ways important to them. The master masons used Euclid's compass and straight-edge constructions to lay out Gothic cathedrals and their details (see Figures 6.1, 6.2 and 6.3). With a compass and straight-edge, a designer can reliably make many geometric figures, including straight lines and angles, divisions into two equal parts, isoceles triangles, and sequences of lengths in ratio to each other. Some constructions, like the trisection of an angle and the famous squaring of a circle (constructing a square of equal area to a given circle) are impossible with these tools.

The addition of rulers to the drawing toolbox allowed designers to work with scaled drawings and to make and transfer measurements within and between drawings.

In the Renaissance designers learned to construct perspectives explicitly. While the exact moment of the (re-)introduction of perspective into Western Art is a matter of debate, at its nexus were artists such as Masaccio and Masolino and painter-sculptor-architects such as Brunelleschi and Alberti. Notably, Alberti's book *On Painting* (1972) was key in disseminating the new perspective ideas. From these early beginnings, perspective became a tool for both depicting and creating architecture. Indeed, the practice of *trompe l'oeil* murals quickly came to blur the boundary between depiction and design.



6.1: Stepwise construction of a Gothic tracery.

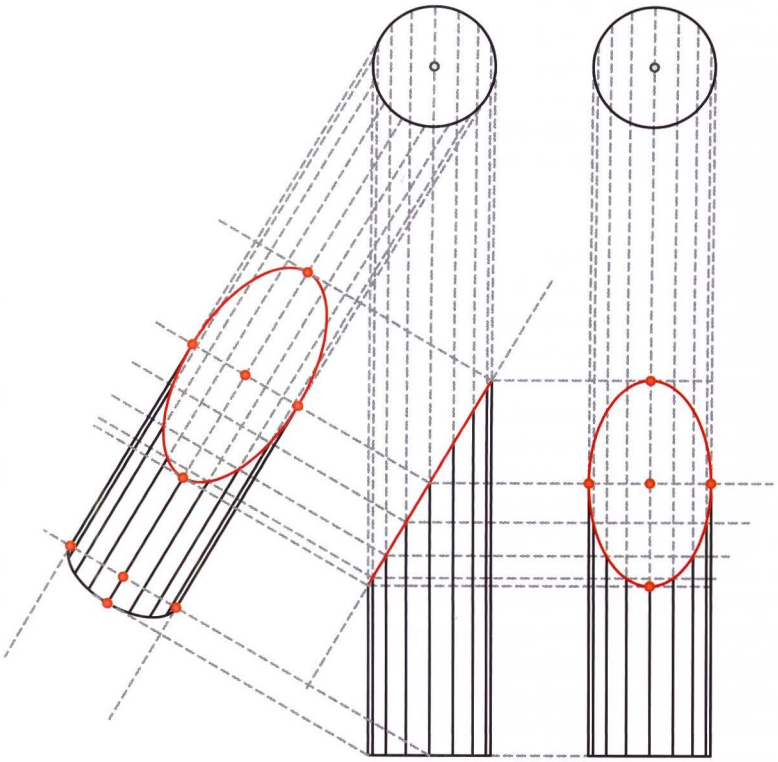


6.2: Gothic traceries were both drawn and constructed using compass and straightedge techniques. These simple media deeply influenced the forms created. In effect, they left indelible marks on the geometry.
Source: Christopher Carlson.

Introduced by Gaspard Monge in 1795, and developed throughout the 19th Century, *descriptive geometry* is a body of techniques for constructing drawings of complex intersecting objects in multiple views. Such drawings enabled new designs for the increasingly complex machinery of the Industrial Revolution. Much of manual mechanical and architectural drawing is based on Monge's principles. In the first half of the 20th Century, it was taught extensively in schools of architecture and engineering. In the last half of the century, it largely faded from the curriculum, at least as an explicit subject.

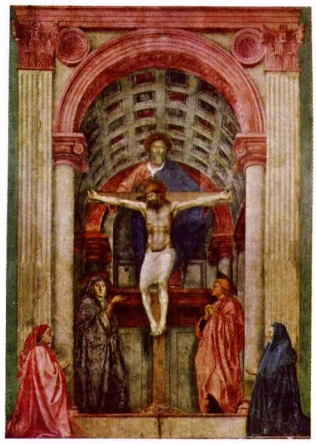
6.4: The painting *Healing of the Cripple and Raising of Tabitha* by Masolino de Panicale (some attribute also to Masaccio) from 1424 in the Brancacci Chapel in Church of Santa Maria del Carmine in Florence, shows perspectives lines whose common intersection argues for understanding and deliberate use of perspective.
Source: The Yorck Project (2002).

6.3: Examples of Gothic traceries.
Source: Christopher Carlson (1993).



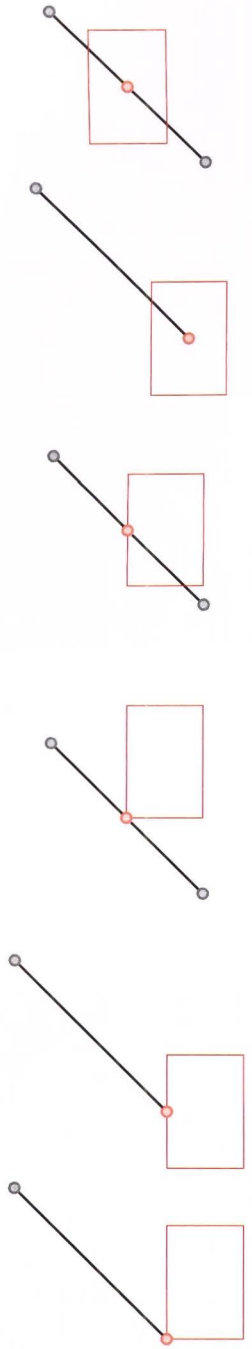
6.5: A simple example of descriptive geometry. Start with a drawing of a cut cylinder (centre) viewed along the edge of the cut. Produce a view showing the true size of the cut ellipse (left) and an orthogonal view at 90° to the original (right).

By the end of the 20th Century, CAD systems supported a wide variety of construction operations. Two principal ideas were *snapping* and *intersection*. Snapping, shown in Figure 6.8, is an interaction technique in which the system recognizes when a *source object* is moved sufficiently close to a *target object* and then places the source object coincident with the target object. If line midpoints are the target, then moving a polygon such that one of its vertices becomes close to a line midpoint will result in the system moving the polygon precisely so that the two points coincide. Intersection operators compute the precise location and result where objects intersect. The intersections produced can themselves take part in subsequent snap interactions. Combined with *global locators* such as grids, guides and reference planes, snaps and intersections play the role of the medieval compass and straight-edge construction system. Contemporary systems also provide the ability to enter numbers representing dimensions or positions, either explicitly by typing them into a dialogue box, or implicitly, through such interaction devices as *dimensions* and *rulers*.



6.7: By 1428, Masaccio's *The Holy Trinity, with the Virgin and Saint John and Donors*, in the *Church of Santa Maria Novella* in Florence, showed explicit perspectival structuring of space and choice of view.
Source: The Yorck Project (2002).

6.6: Andrea Mantegna's oculus on the ceiling of the *Camera degli Sposi, Palazzo Ducale,* in Mantua (1471–74) is an early example of *trompe l'oeil* perspective.
Source: The Yorck Project (2002).



6.8: Snapping has become essential in CAD systems and are part of a toolset that includes grids and numerical specification of location. An end point, midpoint or centre of the source (red rectangle) snaps to an endpoint or midpoint of the target (sloped line segment).

Geometry is at the core of all of these tools, and designers using them certainly became expert, if implicit, geometers within their domain. Using the "toolbox" available at the time, designers have always developed a suite of "tricks of the trade" by which they could reliably create their intended forms. Of course, the medium massaged the design. Traces of the compass and straight-edge show in the pointed arches, lancet windows and quatrefoil bosses of Gothic architecture. Many historians argue that the ability to create perspective changed the focus of Renaissance architecture from objects to expressing movement and views through space.

If you watch a designer using a contemporary CAD system, you are likely to see a combination of all of these techniques (explicit construction, perspective, descriptive geometry, snapping and intersection) and others at play. Designers do indeed use geometric tools in their work.

Parametric modeling is merely the newest toolbox for design work. At the risk of interpreting history as it happens, I'll argue that the tools in this box force a new and different relation to designers. One difference lies in persistence – once an object is placed parametrically, the operation placing it will continue to act every time the model is changed. This means that designers need to predict how the tool will work in the design as it develops. A second difference comes from diversity and abundance – there are simply more tools in the box. Each tool has a mathematical basis, which is open and available for designers to adapt. A third difference lies in the medium itself – however large the toolbox, designers will, at some point, be constrained by it. The solution here is to open the system, to allow designers to directly express new tools. Designers must explicitly translate of geometric "good sense" into precise mathematics and algorithms.

Mastering the new toolbox requires a different kind of geometric knowledge, one that enables designers to predict persistent effects, to understand (at least qualitatively) the diversity and structure of the mathematical toolbox, and to shuttle between intended effect and mathematical invention that models it. This chapter is my best guess at the set of key ideas designers need to master the new medium. Each idea may help in understanding an important group of the tools in the parametric box, as well as having a crisp mathematical and algorithmic structure.

Some ideas are more important than others, at least in practical use. In this chapter I cover a small set of important ideas. Actually, "cover" is a funny word. We usually mean it to treat a concept in detail. It means something different here – what is important is how the concept affects parametric modeling: how it helps predict effect, explain diversity and implement new ideas.

All of these concepts depend upon some basic mathematical ideas. It is these that are "covered" here in something like the traditional sense. These ideas are important because you, the parametric designer, will need them. They are the basis for much of what you do with a system and are the elements from which

you construct new tools for the parametric box. They are necessary, but not sufficient. To become truly expert, you will need to grow beyond this basic starter set. Learning about parametric tools involves bringing together four distinct ways of understanding the mathematics of spatial objects: *geometric*, *visual*, *symbolic* and *algorithmic*. We call each of these a *view* onto the toolbox.

To think geometrically is to know and apply such ideas as the non-location of vectors, the existence of tangents and normals on curves, distance and angles between objects, and perpendicularity in general.

The visual realm comprises both static and dynamic displays. It is important to be able to draw and visualize the basic objects and their relationships. Many problems can be solved with an adroit choice of diagram. To draw a diagram is to choose to leave out certain information and to add other information that is not actually true. For instance, strictly speaking, vectors cannot be drawn – they have no location. We draw them anyway, putting them at specific places and then hopefully remembering not to form conclusions based on location. Drawings are static; our visual system evolved in a dynamic world. Parametric models enable us to take advantage of movement of objects to better understand how geometric relationships actually "work".

Algorithms are recipes. In parametric modeling they spell out the practical tasks of representing and manipulating design objects. They are specific, concrete lists of instructions, meant to be followed literally and in order. We write them with particular goals in mind: move along a curve, project a point, find an areea. The medium of spatial computing is the algorithm.

We live in a world culture with over 3000 years of collective experience using symbolic representation. Symbols allow us to join ways of understanding, and the symbolic realm is where we combine geometric, visual and algorithmic views. Symbols enable inference. They support precise reasoning beyond that practical in other views. The symbolic view is complex and we might well think of it as itself comprising several views. For instance, we can represent relations between points using symbols such as $\dot{p} - \dot{q}$, trigonometric relations such as $\cos(\dot{p}\dot{q}\dot{r}) = \sqrt{2}/2$ or coordinates such as

$$\begin{bmatrix} 3 & 2 & 5 \end{bmatrix}^T - \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}^T = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix}^T$$

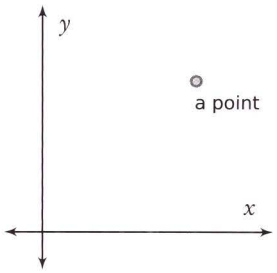
Each way of using symbols supports different insights. A very common way of using symbols is to make explicit relations between different symbolic views. For instance, the two definitions of the scalar product developed in Section 6.1.9 relate geometric and coordinate-based symbolic views and, by that act, enable many insights and proofs.

To learn parametric modeling is to combine geometric, visual, symbolic and algorithmic representations of objects and especially to learn how these forms interrelate. We are in the very early days of this new medium and can predict neither the tools nor the techniques that will surely develop over time. That

said, the geometric ideas of this chapter may be a beginning. They are certainly not an end. As you learn more about geometry, your bookshelf and hard disk will fill with geometry books and papers. There are classic texts, which you would do well to have. The surprisingly readable Euclid's *Elements*, *circa* 300BC (a recent version is (Euclid, 1956)), introduces basic geometric axioms and the process of proof by construction. Books on descriptive geometry entirely fill library shelves. Notable early books include the many editions of Gaspard Monge's (1827) seminal *Geométrié Descriptive*, Charles Davies' (1859) text and Henry Miller's (1911) simply named *Descriptive Geometry* . The hand-illustrated *Natural Structure* (Williams, 1972) provides a visual introduction to symmetries in three-dimensional space, largely through polyhedra and their packings. The best mathematical textbooks are wonders of clarity. Math and proofs are presented with clear arguments, simple notation and compelling figures. But math is not done that way. It is an act of invention and discovery. *Proofs and Refutations* (Lakatos, 1991) is a fictional documentary of a seminar in mathematics. In it, a professor and his students model what really happens in mathematical work. It is surprisingly like design. The cleverly illlustrated *Architectural Geometry* (2007) explains geometric ideas particularly attuned to contemporary architectural design. It grounds its clear, visual explanations in actual design examples. Henderson's (1996) *Experiencing Geometry* gives many proofs and connects diverse topics such as symmetry and differential geometry in context of the plane, cone and sphere. The venerable *Mathematical Elements for Computer Graphics* (Rogers and Adams, 1976) constructs an early bridge from geometry to programming. More succinct is *A Programmer's Geometry* (Bowyer and Woodwark, 1983), which presents a selection of basic geometric structures and provides Fortan-like code to represent them. Twenty years later, with *Geometric Tools for Computer Graphics*, Schneider and Eberly (2003) gave the world a near-encyclopedia of algorithms for geometry. You must have this book if you are serious about geometry and computing. Vince (2005) provides hundreds of formulae, examples and proofs for fundamental geometric objects and relations. The slim volume *Interactive Curves and Surfaces* (Rockwood and Chambers 1996) may lack in depth, but it makes up for this in clarity, insight and fast pace. I especially treasure these few good books. There are hundreds of other useful ones out there.

6.1 Vectors and points

Vectors and points are the basic objects upon which three-dimensional spatial operations are performed. They form the foundation for parametric skill.



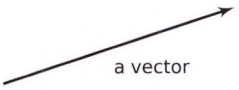
6.1.1 Points

Geometrically, a point is a position in space. Mathematics requires the space, so drawings of points usually include a *coordinate system* to define it.

Points are everywhere in a CAD system interface. Surprisingly, they mostly act as placeholders for the actual computational work. Almost all of the real work and concept is carried by vectors. Points mostly act to provide a spatial position the work done by vectors.

6.1.2 Vectors

Geometrically, a vector is a *direction* and *length* (other names for length are *norm* and *magnitude*). Mathematically, a vector is an abstract object that is part a of *vector space*, itself a mathematical object. We denote a vector space by the symbol V. The length of a vector $\overrightarrow{v}$ is denoted by $|\overrightarrow{v}|$.



Vectors carry most of the computational work in a parametric system. Yet, in most CAD, vectors are secondary objects. The representation and arithmetic of vectors is more complex than that of points, yet is nearly as simple and familiar as basic algebra.

We represent points and vectors as one-dimensional matrices. By convention, we make a choice of either column (the choice made here) or row matrices. It is often convenient to express a column vector as a row vector and vice versa; for this, the ^{T} operator specifies the matrix transpose.

We call the individual matrix elements the *components* of vectors and points.

6.1.3 Vectors and points are different

Since we represent vectors and points as identical matrices, it is easy to get them confused. Consider a tuple of three scalars x, y and z. Two interpretations are pertinent. The tuple

$$\dot{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

specifies a point – a location in some coordinate system. Points are "bound" – they refer to a specific location with respect to some datum. The identical tuple

$$\overrightarrow{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

representing points

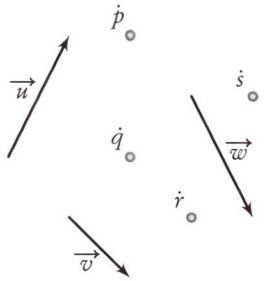
$$\dot{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix}^T$$

representing vectors

$$\overrightarrow{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix}^T$$

specifies a vector – a direction and magnitude (but no fixed location). Vectors are "free" – having no position, they are meaningful at any position in space as simply a direction and magnitude.

Some texts (like this one) treat points and vectors as column vectors. Others use row vectors. Some texts even mix the two in different sections. They mean the same thing, but the notation and order of objects in equations differ. This is life. Get used to it. Consistency, though, is far from the last refuge of the mediocre. It makes a great deal of sense to use a uniform notation in your own work. Just don't expect it elsewhere.



6.9: Vectors and points are not the same. We draw them with distinct glyphs. They obey their own mathematical rules. Yet, we represent them in very similar ways, which can (and does) cause confusion.

Geometrically, we have different intuitions about points and vectors – we draw them differently, as shown in Figure 6.9. Not only do they look different when drawn, but they behave differently. We know that we can "move vectors around" without affecting them in any material way, but that the essence of a point is its position. However, we represent them with the same syntax – a row or column vector. From this notational convenience arises one of the principal obstacles in developing an intuitive grasp of the mathematics of computer graphics. Among other things, our intention here is to cement in place an understanding that makes explicit the difference between these two fundamental kinds of objects.

Later we add a fourth row to the vector and point representation. For vectors the value in this row will always be 0; for points it will be 1. Vectors and points so represented are said to be in *homogeneous coordinates*. An example vector is

$$\vec{v} = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} x & y & z & 0 \end{bmatrix}^T$$

A point so represented is

$$\dot{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$$

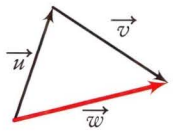
You will find more on homogenenous coordinates in almost every book on spatial computing – we introduce them here as you will see them elsewhere and as Section 6.5 uses them to represent coordinate systems.

6.1.4 The arithmetic of vectors

Basic mathematical literacy is founded on the arithmetic of numbers. Addition, subtraction, multiplication and division relate numbers to each other. Their combination into expressions with precedence rules (brackets before exponents before multiplication and division before addition and subtraction) are skills learned in grade school and used almost subconsciously in day-to-day work. Geometry is founded on arithmetic too: the arithmetic of vectors and points. This differs from number arithmetic in several ways.

Vectors (actually *vector spaces*) define two operations, *vector addition* and *scalar multiplication*.

Vector addition combines two vectors to create a third:



$$\vec{u} + \vec{v} = \vec{w}$$

Scalar multiplication ($a\vec{u}$ or $a \cdot \vec{u}$) combines a real number and a vector to produce a second vector with the identical direction, but a possibly different length:

$$a\vec{u} = \vec{v}$$
$$a = 2$$

Vectors, together with vector addition and scalar multiplication, obey rules. These are the analogues of arithmetic over the familiar real numbers and are the basis for almost all other geometry in parametric modeling.

Closure of addition

$\vec{u} + \vec{v} \in \mathsf{V}$, the space of all vectors

Adding two vectors always produces a vector.

Zero vector

$\vec{v} + \vec{0} = \vec{v}$

There is a unique *zero vector*. This plus any vector leaves the vector unchanged.

vector addition

$$\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 7 \end{bmatrix}$$

scalar multiplication

$$2\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ -6 \end{bmatrix}$$

scalar multiplication
alternate notation

$a\vec{u}$
or
$a \cdot \vec{u}$

addition is closed

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix}$$

zero vector

$$\begin{bmatrix} 1 \\ -3 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ 2 \end{bmatrix}$$

inverse vector

$$\begin{bmatrix} 1 \\ -3 \\ 2 \end{bmatrix} + \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Inverse vector

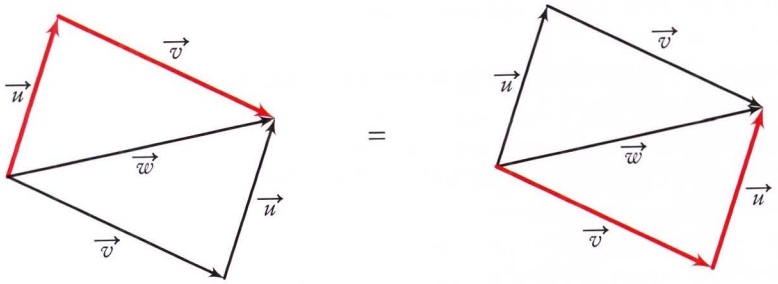
$$\vec{v} + -\vec{v} = \vec{0}$$

Every vector has an inverse. By convention, the inverse vector establishes the operation of subtraction as $\vec{v} - \vec{u} = \vec{v} + (-\vec{u})$.

Commutativity of addition

addition is commutative

$$\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$



$$\vec{u} + \vec{v} = \vec{v} + \vec{u}$$

A given addition and its reverse order produce the same result.

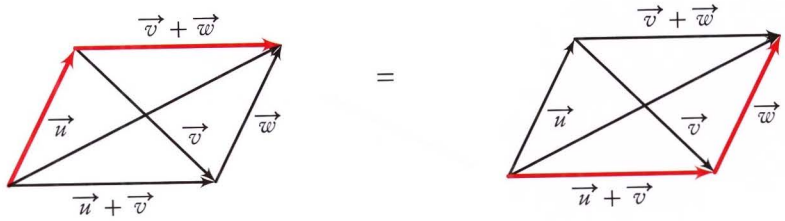
Associativity of addition

addition is associative

$$\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}\right) + \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \left(\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 6 \\ 4 \\ 9 \end{bmatrix}$$



$$\vec{u} + (\vec{v} + \vec{w}) = (\vec{u} + \vec{v}) + \vec{w}$$

The order in which a given addition is done does not affect the outcome.

Closure of scalar multiplication

multiplication is closed

$$2\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ -6 \end{bmatrix}$$

$$a\vec{v} \in V$$

Scalar multiplication always produces a result.

Identity element in scalar multiplication

$$1\vec{v} = \vec{v}$$

Multiplying a vector by 1 yields the original vector.

multiplicative identity

$$1\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix}$$

Associativity of scalar multiplication

$$(ab)\vec{v} = a(b\vec{v})$$

Scaling by a number or successively by its factors is the same.

multiplication is associative

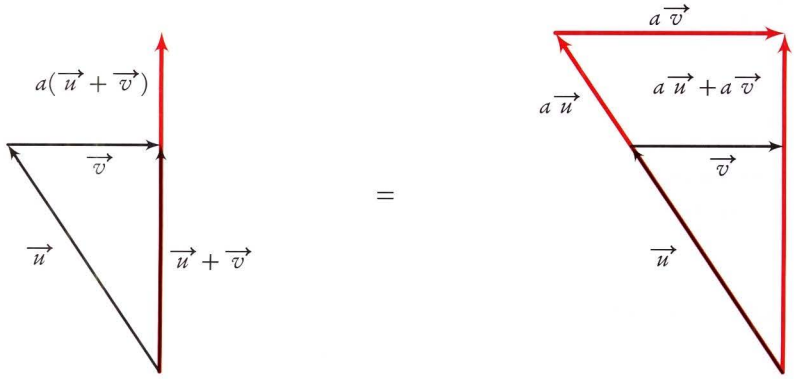
$$(3 \times 2)\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} = 3\left(2\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix}\right)$$

Left distributivity of scalar multiplication

$$(a+b)\vec{v} = a\vec{v} + b\vec{v}$$

Multiplying a vector by a sum of scaling factors is the same as adding vectors scaled by each factor. You can add scaling factors then multiply or vice versa.

Right distributivity of scalar multiplication



$$a(\vec{u}+\vec{v}) = a\vec{u} + a\vec{v}$$

Scaling a vector sum or summing the scaled components gives the same result. You can add vectors and then multiply, or vice versa.

left distributivity

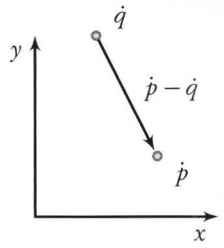
$$(3 \times 2)\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} = 3\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} + 2\begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix}$$

right distributivity

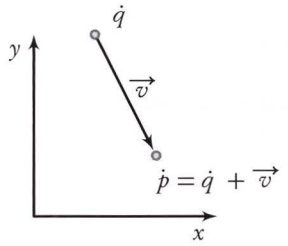
$$3\left(\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix}\right) = 3\begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} + 3\begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix}$$

6.1.5 The arithmetic of points

In sharp comparison to vectors, points have only a single operation. *Subtracting* two points yields a vector.



Points and vectors combine with a single operation. The *addition* of a point and vector produces a point.



The overall structure of a typical arithmetic calculation in space is to start with points, use point–point subtraction to convert to vectors, do the serious work with vectors and convert back to points with point–vector addition.

6.1.6 Combining vectors

The operations of vector addition and scalar multiplication act on vectors to produce other vectors. Several terms describe the vectors so produced.

Linear combinations

A linear combination of a set of vectors is a sum arbitrarily scaling each of the vectors. Formally, a combination of vectors $\overrightarrow{u}_i, i = 0 \ldots n$ generating $\overrightarrow{v}$ is linear if it can be expressed in the following form, where a_i are arbitrary reals and are called the *coefficients* of the linear combination.

$$\overrightarrow{v} = a_0 \overrightarrow{u}_0 + \cdots + a_n \overrightarrow{u}_n$$

Linear dependence and independence

Two vectors are *linearly independent* if one is not a scalar multiple of the other. A set of vectors $\vec{u}_i, i = 0 \ldots n$ is linearly independent if no vector is a linear combination of the others.

Formally, linear independence occurs if $a_0 = \cdots = a_n = 0$ is the only solution to $a_0 \vec{u}_0 + \cdots + a_n \vec{u}_n = \vec{0}$.

Span of a set of vectors

The span S of a set of vectors B is the set generatable by a linear combination of the vectors in B.

Vector basis

A set of vectors B is a basis for a vector space V if it is linearly independent and spans V.

The symbolic idea of a vector basis captures the geometric idea that a coordinate system has three vectors. The three vectors of a 3D coordinate system are basis vectors for that system.

Uniqueness of a linear combination

Given a basis B, every vector in the space spanned by B can be expressed as a unique linear combination of the vectors in B.

There are two powerful ideas here. First, basis vectors combine to represent any other vector in the space. Second, such representation is unique: there is only one combination of the basis vectors that will do the job.

Bases for 2D and 3D

Any two linearly independent vectors form a 2D basis, which can express all 2D vectors through their linear combinations. Similarly, three linearly independent vectors form a 3D basis.

Natural basis

The *natural basis* is the most simple form. Each of its unit vectors has precisely one non-zero component. So the natural basis for $\mathbb{R}^3$ comprises three vectors.

natural basis

$$i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$
$$j = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$$
$$k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

6.1.7 Length and distance

The *norm* (or *length*) of a vector $\overrightarrow{v} = \langle v_0, \ldots, v_n \rangle$, denoted $|\overrightarrow{v}|$ is defined as the square root of the sum of squares of its components, that is,

$$|\overrightarrow{v}| = \sqrt{{v_0}^2 + \cdots + {v_n}^2}$$

Note that, in two dimensions, this is simply a statement of Pythagoras's Law. So, for two-dimensional vectors the length of a vector $\overrightarrow{v}$ is

$$|\overrightarrow{v}| = \sqrt{{\overrightarrow{v}_x}^2 + {\overrightarrow{v}_y}^2}$$

For three-dimensional vectors

$$|\overrightarrow{v}| = \sqrt{{\overrightarrow{v}_x}^2 + {\overrightarrow{v}_y}^2 + {\overrightarrow{v}_z}^2}$$

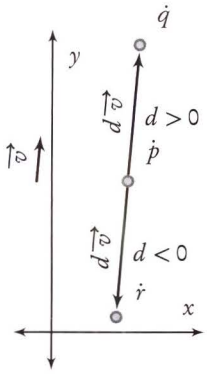
The *distance* between two points is the length of the vector that results from their subtraction.

$$|\overrightarrow{\dot{p}\dot{q}}| = |\dot{q} - \dot{p}| = \sqrt{(\dot{q}_0 - \dot{p}_0)^2 + \cdots + (\dot{q}_n - \dot{p}_n)^2}$$

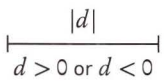
The direction of a vector $\overrightarrow{v}$ is another vector $\overrightarrow{dir}$ (called a *direction vector*) such that

$$\overrightarrow{dir}_{\overrightarrow{v}} = \frac{\overrightarrow{v}}{|\overrightarrow{v}|}$$

Any direction vector $\overrightarrow{dir}$ is of length 1 (aka *unit length*).



When dealing with vectors, it is useful to have a concept of *signed distance*. Given an initial point $\dot{p}$, a direction vector $\overrightarrow{dir}$, and a scaling factor d the point $\dot{q}$ is the signed distance d from $\dot{p}$ along $\overrightarrow{dir}$. A positive signed distance means that distance is measured "along" the vector; a negative signed distance means that distance is measured in the opposite direction. Signed distances do not translate well into drawings – dimension lines convey unsigned distances by convention. When drawn using a dimension line, a signed distance d reduces to its absolute value $|d|$.



Signed distances are comparable to our perceptions of subtraction along a real number line: $8 - 5 = 3$ is the signed distance from 5 to 8; whereas $5 - 8 = -3$ is the signed distance from 8 to 5.

6.1.8 Bound and free vectors

We usually think of vectors as being *free* or having no position in space: they give only length and direction, no matter where they are "located". Another interpretation is to treat vectors as *bound*, that is, beginning at a common point,

usually the origin, which by convention is labeled $\dot{O}$. *Position vector* is another phrase for *bound vector.* A set of bound vectors with the origin as the common point identifies a set of points, one for each bound vector.

Bound vectors require a common point. When they are specified as matrices, they require an entire *coordinate system*. One has to know to where and in which direction to apply the components of the vector. The solution is to always have in mind three vectors, called $\overrightarrow{i}$, $\overrightarrow{j}$ and $\overrightarrow{k}$ representing the x-axis, y-axis and z-axis respectively. Thus a bound vector $\overrightarrow{v} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is actually the vector sum

$$x\overrightarrow{i} + y\overrightarrow{j} + z\overrightarrow{k}$$

and the point with which it is associated is

$$\dot{p} = \dot{O} + \overrightarrow{v}$$

6.1.9 The scalar product

The *scalar product* of two vectors is a number that can be used in several ways. It provides a test for perpendicularity, a measure of the angle between two vectors, a tool for projecting one vector onto another and a measure of the length of a vector. Informally, the scalar product is also known as the *dot product.*

The scalar product of two vectors $\overrightarrow{u} \bullet \overrightarrow{v}$ is

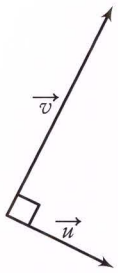
$$\overrightarrow{u} \bullet \overrightarrow{v} = \sum_{i=0}^{n} \overrightarrow{u}_i \cdot \overrightarrow{v}_i$$

For example, let $\overrightarrow{u} = \begin{bmatrix} \overrightarrow{u}_x & \overrightarrow{u}_y & \overrightarrow{u}_z \end{bmatrix}^T$ and $\overrightarrow{v} = \begin{bmatrix} \overrightarrow{v}_x & \overrightarrow{v}_y & \overrightarrow{v}_z \end{bmatrix}^T$ be two 3D vectors. The scalar product $\overrightarrow{u} \bullet \overrightarrow{v}$ is

$$\overrightarrow{u} \bullet \overrightarrow{v} = \overrightarrow{u}_x \overrightarrow{v}_x + \overrightarrow{u}_y \overrightarrow{v}_y + \overrightarrow{u}_z \overrightarrow{v}_z$$

The scalar product is defined on vectors only. By convention, it can be applied to points. When a point $\dot{p}$ is used in a scalar product, the meaning is that the vector involved is that from the origin $\dot{O}$ to the point $\dot{p}$.

The scalar product has several properties. These are useful when working with constructions and derivations involving the scalar product.



$\vec{u} \bullet \vec{v} = 0$

$\vec{u} \bullet \vec{v}$ is a number

$\vec{u} \bullet \vec{v} = \vec{v} \bullet \vec{u}$

$\vec{u} \bullet \vec{0} = 0 = \vec{0} \bullet \vec{u}$

$\vec{u} \bullet \vec{u} = |\vec{u}|^2$

$(a\vec{u}) \bullet \vec{v} = a(\vec{v} \bullet \vec{u}) = \vec{v} \bullet (a\vec{u})$

$\vec{u} \bullet (\vec{v} + \vec{w}) = \vec{u} \bullet \vec{v} + \vec{u} \bullet \vec{w}$

Perpendicularity of vectors

If $\vec{u}$ and $\vec{v}$ are two non-zero vectors, they are perpendicular if and only if *(iff)* their scalar product is equal to 0.

The angle between two vectors

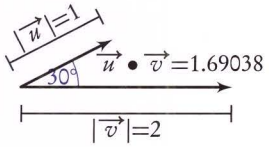
If $\vec{u}$ and $\vec{v}$ are two non-zero vectors, then they determine a unique angle α, $0 \leq \alpha \leq 180°$. It can be shown that another way of stating the scalar product includes α.

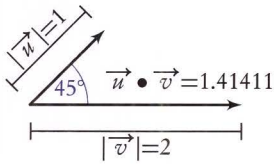
$$\vec{u} \bullet \vec{v} = |\vec{u}||\vec{v}| \cos\alpha$$

If $\vec{u}$ and $\vec{v}$ are both unit vectors then

$$\vec{u} \bullet \vec{v} = \cos\alpha$$

$$\alpha = \arccos(\vec{u} \bullet \vec{v})$$

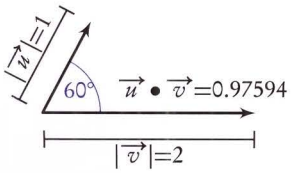




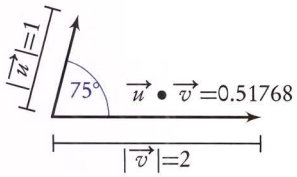
6.1.10 Projecting one vector onto another

Algebraically the scalar product is the sum of products of the components of its arguments.

Geometrically, the scalar product is a measure of the projection of one vector onto another. First, consider the case in which both $\vec{u}$ and $\vec{v}$ are unit vectors. Then the scalar product is simply $\cos\alpha$ or the projection of $\vec{v}$ onto $\vec{u}$ or vice versa.



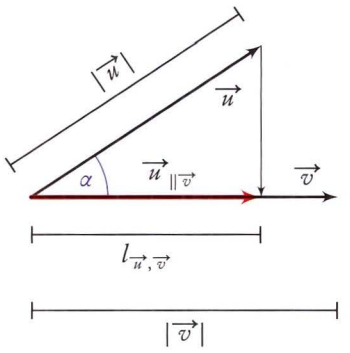
When either $\vec{u}$ or $\vec{v}$ are non-unit, the scalar product is simply scaled by their lengths. Thus, in general, the projection length $l_{\vec{u},\vec{v}}$ of a vector $\vec{u}$ onto a vector $\vec{v}$ is given by



$$l_{\vec{u},\vec{v}} = |\vec{u}| \cos\theta = \frac{|\vec{u}||\vec{v}| \cos\theta}{|\vec{v}|} = \frac{\vec{u} \bullet \vec{v}}{|\vec{v}|}$$

Note carefully that $l_{\vec{u},\vec{v}}$ is a measure of the length of the vector $\vec{u}$ projected onto the vector $\vec{v}$. Often, what is needed is the actual projected vector. There is no universal notation for such vectors. Here we modify the notation used in Schneider and Eberly (2003, p. 87). The projection of a vector u onto a vector v is given by

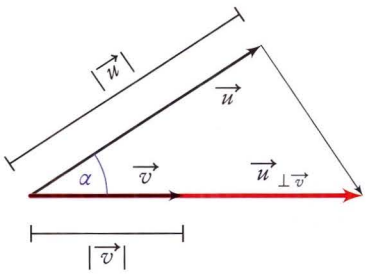
$$\vec{u}_{\parallel\vec{v}} = l_{\vec{u},\vec{v}} \frac{\vec{v}}{|\vec{v}|} = \frac{\vec{u} \bullet \vec{v}}{|\vec{v}|} \frac{\vec{v}}{|\vec{v}|} = \frac{\vec{u} \bullet \vec{v}}{\vec{v} \bullet \vec{v}} \vec{v} \tag{6.1}$$



6.10: The projection of vector $\vec{u}$ onto $\vec{v}$.

6.1.11 Converse projection

Sometimes the projection of a vector perpendicular to itself and onto another vector is useful.

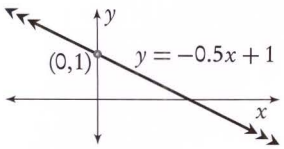


6.11: The converse projection of vector $\vec{u}$ onto $\vec{v}$.

The converse projection $\vec{u}_{\perp\vec{v}}$ is given by

$$\vec{u}_{\perp\vec{v}} = \frac{\vec{u} \bullet \vec{u}}{\vec{u} \bullet \vec{v}} \vec{v}$$

6.2 Lines in 2D

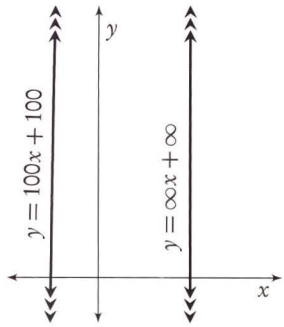


6.12: Explicit equations are simple to plot. Place a point $\dot{b}$ at $(0, b)$ on the y-axis. Draw a line with slope m through $\dot{b}$.

After vectors and points, lines are the most basic spatial objects. Lines in two-dimensional space (2D) are almost exactly analogous to planes in three-dimensional space. In 2D, lines can be represented in several ways. Each representation makes some mathematical inferences and/or algorithm steps easier than others.

From a geometric perspective there are several objects from which a line can be built. For instance, a point known to be on a line, the direction of the line, the point at which the line crosses a principal axis, the slope of the line and a direction normal to the line can all be used as part of a line representation. Each of the four equations below appeal to one or more of these geometric ideas.

6.2.1 Explicit equation



6.13: Lines of near vertical slope have high coefficients. Lines with vertical slope have infinite coefficients. Designers frequently want to use such lines.

The *explicit equation* is also called the *slope y-intercept equation*.

$$y = mx + b$$

In this equation, m is the slope of the line (the rise over the run) and b is the y-intercept. This is, perhaps, the most familiar equation. But it isn't a very good one for computing. Vertical lines have an infinite slope, so cannot be represented with the equation. Lines that are nearly vertical have slopes that approach or exceed the practical numerical precision of computation.

6.2.2 Implicit equation

The *implicit* equation is a simple linear equation.

$$ax + by + d = 0$$

Note: we use d in the equation rather than the more common c in order to make the corresponding line and plane equations (Section 6.4.2) cohere. The character d is also a reminder that of the role this variable plays in the equation – it carries information about distance – see below.

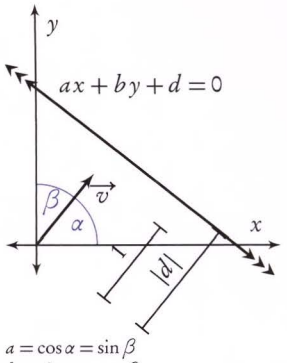
The implicit equation provides an easy test to determine if a given point is on a line. Simply substitute the point's coordinates for x and y in the equation. If the equation is satisfied, the point is on the line. In contrast, constructing a point on the line is less direct.

When $d = 0$ the line passes through the origin, as can be seen by assigning $x = 0$, $y = 0$ and $d = 0$ to the equation.

The vector $\vec{v} = \begin{bmatrix} a & b \end{bmatrix}$ is normal (perpendicular) to the line.

When $|\vec{v}| = 1$ the implicit equation is *normalized.* In this form, it has a simple geometric interpretation, in which the vector components relate directly to the angles α and β and d is a signed distance. The values $a = \cos\alpha$ and $b = \cos\beta$ are the *direction cosines* of the vector $\vec{v}$, and $-d$ is the distance along $\vec{v}$ from the origin to the line. If d is negative, the line is located in the direction pointed to by $\vec{v}$. If d is positive, the line lies at the distance d from the base of $\vec{v}$ in the opposite direction.

When $\vec{v}$ is not normalized, things are more complex. The vector $\vec{v}$ remains perpendicular to the line. The direction cosines can no longer be read directly from $\vec{v}$; they can be computed by scaling the vector by its length $1/\sqrt{a^2+b^2}$. The quantity d becomes the negative of the distance to the line multiplied by the length of $\vec{v}$, that is, $\sqrt{a^2+b^2}$. The actual signed distance from the origin to the line along $\vec{v}$ is $-d/|\vec{v}|$. When $|\vec{v}| = 1$, $-d$ is the signed distance!



$a = \cos\alpha = \sin\beta$
$b = \sin\alpha = \cos\beta$
$-d$ = signed distance from origin to line

Changing the sign of d creates a parallel line equidistant from the origin, but along the opposite vector direction.

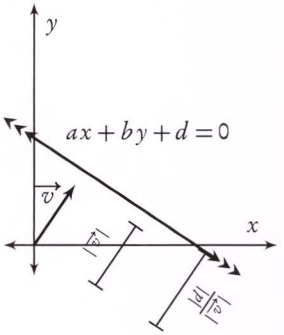
6.2.3 Line operator

The implicit equation gives a very tidy matrix form for representing lines called the *line operator.* A line is a row vector $\gamma = \begin{bmatrix} a & b & d \end{bmatrix}$ such that a point $\dot{p} = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ is on the line γ if and only if

$$\gamma\dot{p} = \begin{bmatrix} a & b & d \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$ax + by + d = 0$$

This test has all of the properties of the implicit line equation above. Its form as a matrix makes it easy to visualize other properties. It is useful to break the line operator into two parts: $\gamma_{\vec{v}}$ and γ_d, representing the vector and distance components respectively.



$$\gamma = \begin{bmatrix} \gamma_{\vec{v}} & | & \gamma_d \end{bmatrix}$$

where

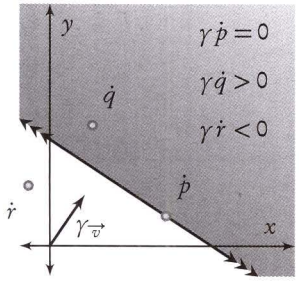
$$\gamma_{\vec{v}} = \begin{bmatrix} a & b \end{bmatrix}, \text{ and } \gamma_d = \begin{bmatrix} d \end{bmatrix}$$

In spite of the simplicity of the line operator, much can be inferred from it. The first inference is its effect on vectors. We define the result of applying the line operator to a vector as

$$\begin{aligned}\gamma\overrightarrow{v} &= \begin{bmatrix} a & b & d \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \\ &= ax + by + 0d \\ &= ax + by\end{aligned}$$

Consider any vector $\overrightarrow{v} = \begin{bmatrix} x & y & 0 \end{bmatrix}^T$. Since the third vector element is zero, the effect of $\gamma\dot{p}$ is to compute the scalar product of the first two elements of the line operator and the vector. We know that vectors are perpendicular when their scalar product is zero. Since $\gamma_{\overrightarrow{v}}$ is perpendicular to the line, $\gamma\overrightarrow{v} = 0$ for any vector parallel to the line.

The line operator can be multiplied by any real number r (except 0) without changing the line. This is equivalent to scaling both the vector $\gamma_{\overrightarrow{v}}$ and the value γ_d by r. Of course, the result of scaling the line operator by zero is undefined.



Lines are sided. The vector $\overrightarrow{v} = \begin{bmatrix} a & b \end{bmatrix}$ points towards the positive side. When the result of applying the line operator is positive, the point so tested lies on the positive side of the line. When the result is negative, the point lies on the negative side: the side away from which $\overrightarrow{v}$ points. Multiplying the operator by a negative number reverses line sidedness. Further, when the line operator is normalized ($\gamma_{|\overrightarrow{v}|} = 1$), it produces the signed distance from a point to the line; $\gamma\dot{p}$ = distance from $\dot{p}$ to γ along $\gamma_{\overrightarrow{v}}$. Sidedness is often used to represent where solid material lies in a design.

6.2.4 Normal-point equation

The *normal-point* equation of a line is defined through a point $\dot{q}$ and a non-zero vector $\overrightarrow{n}$ normal to the line. Since $\dot{q}$ is on the line, any vector between it and another point on the line must be perpendicular to the line normal $\overrightarrow{n}$ and therefore have a zero scalar product with $\overrightarrow{n}$.

$$\overrightarrow{n} \bullet (\dot{p} - \dot{q}) = 0$$

This equation provides a simple test using vectors and points as entire entities to determine if $\dot{p}$ is on the line. It is thus useful in parametric modelers that provide basic vector operations – no conversion to other equation forms or unpacking of vector and point components is needed.

6.2.5 Parametric equation

The *parametric* equation is perhaps the most widely used form. This is because it works in both 2D and 3D and because it is *constructive*, that is, it can be used to generate points on the line. In contrast, the implicit line equation is good for testing whether a point lies on a line.

A line is uniquely defined by a point and a vector. Given a point $\dot{p}$ and a vector $\overrightarrow{v}$, any point $\dot{p}(t)$ on the line has the functional equation

$$\dot{p}(t) = \dot{p} + t\,\overrightarrow{v}$$

where t is a real value that scales the vector $\overrightarrow{v}$. Each value of t picks out a distinct point on the line.

Let the point $\dot{p}_1$ be the sum of $\dot{p}_0$ and $\overrightarrow{v}$. Then

$$\begin{aligned}\dot{p}(t) &= \dot{p}_0 + t(\dot{p}_1 - \dot{p}_0) \\ &= (1-t)\dot{p}_0 + t\dot{p}_1 \end{aligned} \tag{6.2}$$

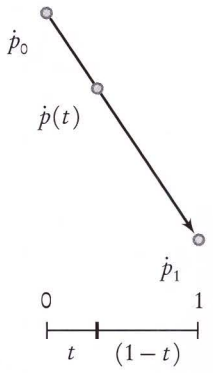
or alternatively (in vector form)

$$\dot{p}(t) = \dot{p}_0 + t(\overrightarrow{\dot{p}_0\dot{p}_1})$$

Each of the forms above is called a *parametric line equation* with parameter t.

Equation 6.2 can be rewritten as

$$\begin{aligned}\dot{p}(t) &= \dot{p}_0 + t(\dot{p}_1 - \dot{p}_0) \\ &= (1-t)\dot{p}_0 + t\dot{p}_1 \\ &= t_0\dot{p}_0 + t_1\dot{p}_1, \quad \text{where} \quad (t_0 + t_1 = 1)\end{aligned}$$



Even though they involve points, parametric line equations are drawn without coordinate systems, because the point produced $\dot{p}(t)$ depends only upon the points $\dot{p}_0$ and $\dot{p}_1$ – it is independent of where the points are located in space.

Changing the parameter t moves the point $\dot{p}(t)$ along the line. Specifically, it moves $\dot{p}(t)$ *in proportion* to t. For example, if $t = 0.4$, $\dot{p}(t)$ is 4/10ths of the distance along the line from $\dot{p}_0$ to $\dot{p}_1$. This *linear* relationship between t and $\dot{p}(t)$ holds only for lines. Section 6.9.5 shows that it does not hold for curves.

If $t = 0$ then $\dot{p}(t) = \dot{p}_0$
If $t = 1$ then $\dot{p}(t) = \dot{p}_1$
If $0 < t < 1$ then $\dot{p}(t)$ is between $\dot{p}_0$ and $\dot{p}_1$
If $t < 0$ then $\dot{p}(t)$ is to the left of $\dot{p}_0$
If $t > 1$ then $\dot{p}(t)$ is to the right of $\dot{p}_1$

6.2.6 Projecting a point to a line

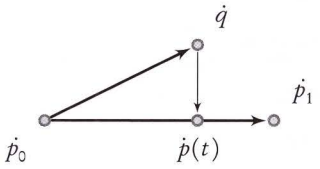
Projecting a point $\dot{p}$ onto a line $\overline{L}$ means finding the point $\dot{q}$ that on the line that is closest to $\dot{p}$. Alternatively it means finding $\dot{q}$ such that the line between $\dot{p}$ and $\dot{q}$ is perpendicular to the line $\overline{L}$.

Projection is most easily expressed when the line is in normalized line operator form, where $\gamma\dot{p}$ is the signed distance from the point to the line. The projection of $\dot{p}$ to the line is the sum of $\dot{p}$ and the normal vector to the line $\gamma_{\vec{v}}$ scaled by the result of the line operator $\gamma\dot{p}$.

$$\dot{p}_{proj} = \dot{p} + (\gamma\dot{p})\gamma_{\vec{v}}$$

Often though, it is useful to have the parametric coordinate of the projected point. Using the parametric line equation, one way to compute the projection is to appeal to Equation 6.1 on page 97 for vector projection. Given point $\dot{q}$ to project onto the parametric line at $\dot{p}(t) = \dot{p}_o + t(\dot{p}_1 - \dot{p}_0)$, simply add the projection of of $\overrightarrow{\dot{p}_0\dot{q}}$ onto $\overrightarrow{\dot{p}_0\dot{p}_1}$ to the point $\dot{p}_0$. The projected point $\dot{p}(t)$ on line $\overline{L}$ is thus:

$$\dot{p}(t) = \dot{p}_0 + \frac{\overrightarrow{\dot{p}_0\dot{q}} \bullet \overrightarrow{\dot{p}_0\dot{p}_1}}{\overrightarrow{\dot{p}_0\dot{p}_1} \bullet \overrightarrow{\dot{p}_0\dot{p}_1}} \overrightarrow{\dot{p}_0\dot{p}_1} \tag{6.3}$$



$$t = \frac{\overrightarrow{\dot{p}_0\dot{q}} \bullet \overrightarrow{\dot{p}_0\dot{p}_1}}{\overrightarrow{\dot{p}_0\dot{p}_1} \bullet \overrightarrow{\dot{p}_0\dot{p}_1}} \tag{6.4}$$

6.14: The projection of point $\dot{q}$ onto parametric line $\overline{L}$ at $\dot{p}(t)$.

The parameter t in Equation 6.4 is exactly the same as the scale factor for the vector in Equation 6.3.

6.3 Lines in 3D

In three dimensions, lines have neither an explicit nor an implicit equation. For almost all practical purposes, the parametric equation dominates. In it, a point and a vector defines a line. Its form is exactly the same as in two dimensions – given a point $\dot{p}_1$ and a vector $\overrightarrow{v}$, any point $\dot{p}(t)$ on the line has the equation

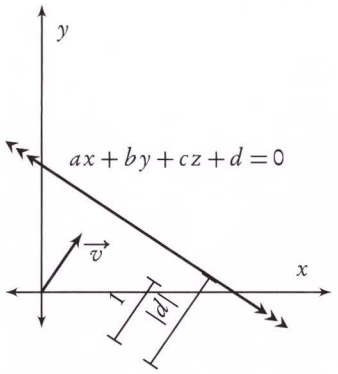
$$\dot{p}(t) = \dot{p}_1 + t\overrightarrow{v}$$

The only difference is that the points and vectors have three components rather than two.

6.4 Planes

Planes in 3D are the natural counterpart to lines in 2D. The implicit and parametric line equations easily expand to represent planes.

6.4.1 Normal vector



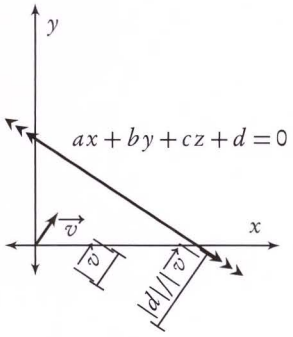
$a = \cos\alpha$
$b = \cos\beta$
$c = \cos\gamma$
$-d$ = signed distance from origin to line

There are several ways to define a plane: three non-collinear points; a vector normal to the plane plus a point on the plane; and two non-collinear vectors parallel to the plane plus a point on the plane all suffice.

Given a vector $\overrightarrow{n}$ normal to the plane, and a point $\dot{p}$ on the plane, any vector parallel to the plane will be perpendicular to the plane normal $\overrightarrow{n}$. The scalar product provides an easy test for parallelism. The known point $\dot{p}$ on the plane defines a vector to any other point $\dot{q}$ in space. If this vector is perpendicular to $\overrightarrow{n}$, then $\dot{q}$ is on the plane.

Without any loss of generality, exactly the same drawings explain both planes and lines. The third dimension is simply suppressed by using an orthogonal drawing along one of the principal spatial axes. The only information that this fails to reveal concerns the angles that vectors make with the principal axes.

6.4.2 Implicit equation



The implicit equation for a plane is

$$ax + by + cz + d = 0$$

Just as for a two-dimensional line, all but the last term describes a vector. When the vector is of unit-length, the equation is normalized. $a = \cos\alpha$, $b = \cos\beta$ and $c = \cos\gamma$ are the *direction cosines* of the vector $\overrightarrow{v}$, and $-d$ is the signed distance along $\overrightarrow{v}$ from the origin to the line.

6.4.3 Normal-point equation

The *normal-point* equation of a plane takes a point $\dot{q}$ and a non-zero vector $\overrightarrow{n}$ normal to the plane. The equation is exactly the same as for lines; the objects involved simply have a z-component.

$$\overrightarrow{n} \bullet (\dot{p} - \dot{q}) = 0$$

6.4.4 Plane operator

Just as for 2D lines, the implicit equation gives a tidy matrix form to represent planes.

A plane is represented as a row vector $\gamma = \begin{bmatrix} a & b & c & d \end{bmatrix}$ such that a point $\dot{p} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ is on the plane γ if and only if

$$\gamma \dot{p} = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0 \tag{6.5}$$

$$ax + by + cz + d = 0$$

This test, which is a matrix representation of the implicit plane equation, is known as the *plane operator*. It has all of the properties of the implicit plane equation above.

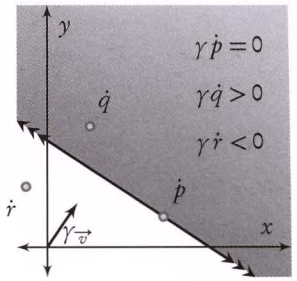
The direction cosines of the normalized plane operator are the cosines of the angles α, β and γ between the vector $\overrightarrow{v}$ and the x-, y- and z-axes respectively.

For vectors parallel to the plane, the plane operator is equal to 0.

$$\begin{aligned} \gamma \overrightarrow{v} &= \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \\ &= ax + by + cz + 0d \\ &= ax + by + cz \\ &= 0, \text{ iff } \overrightarrow{v} \text{ is parallel to the plane.} \end{aligned}$$

The plane operator can be multiplied by any real number r (except 0) without changing the plane.

Plane have sides. As for lines, the vector $\overrightarrow{v} = \begin{bmatrix} a & b & c \end{bmatrix}$ points towards the positive side. When the plane operator is normalized, the number $\gamma\,\dot{p}$ itself gives the signed distance of $\dot{p}$ to the plane.



6.4.5 Parametric equation

A plane is defined by two vectors and a point $\dot{p}$ known to be on the plane. The plane comprises all points that can be reached by binding a linear combination of the two vectors to the point $\dot{p}$.

The parametric equation of a plane takes two parameters; each acts as a scaling factor for one of two vectors defining the plane.

$$\dot{p}(u,v) = \dot{p} + (u \cdot \overrightarrow{u} + v \cdot \overrightarrow{v})$$

Typically, the vectors $\overrightarrow{u}$ and $\overrightarrow{v}$ are chosen to be mutually perpendicular and of unit length. Such a choice establishes a two-dimensional coordinate system on the plane. Points can then be represented locally with respect to the plane.

The parametric plane equation is easily derived from three points defining a plane. Given three non-collinear points $\dot{p}_0$, $\dot{p}_1$ and $\dot{p}_2$.

$$\dot{p}_{(u,v)} = \dot{p}_0 + (u \cdot \overrightarrow{\dot{p}_0\dot{p}_1}) + (v \cdot \overrightarrow{\dot{p}_0\dot{p}_2})$$

6.4.6 Projecting a point onto a plane

Projecting a point $\dot{q}$ onto a plane means finding the closest point on the plane to $\dot{q}$. Equivalently, it means finding the point on the plane that intersects the line given by $\dot{q}$ and the normal vector to the plane. The latter definition gives a strong hint for using the plane operator.

Just as for lines, in the normalized plane operator, the signed distance between a point $\dot{q}$ and the plane γ is given by $\gamma\dot{q}$. So, the projection of $\dot{q}$ to the plane is the sum of $\dot{q}$ and the normal vector to the plane $\gamma_{\overrightarrow{v}}$ scaled by the result of the plane operator $\gamma\dot{q}$.

$$\dot{q}_{proj} = \dot{q} + (\gamma\dot{q}) \cdot \gamma_{\overrightarrow{v}}$$

If the parameters of the projected point are needed use the parametric plane equation. It is best if the plane vectors are mutually perpendicular and of unit length. Then the scalar products of the vector $\overrightarrow{\dot{p}\dot{q}}$ and each of $\overrightarrow{u}$ and $\overrightarrow{v}$ give the parameters u and v of the projected point on the plane.

$$\dot{q}(u,v) = \dot{p} + (\overrightarrow{\dot{p}\dot{q}} \bullet \overrightarrow{u}) \cdot \overrightarrow{u} + (\overrightarrow{\dot{p}\dot{q}} \bullet \overrightarrow{v}) \cdot \overrightarrow{v}$$

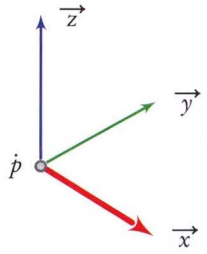
6.5 Coordinate systems ≡ frames

What is a coordinate system?

You likely know the answer informally. Coordinate systems define the axes of a space. The x- and y-axes on a graph define the dimensions of a two-dimensional coordinate system. Add a z-axis to get a three-dimensional system. Systems are located in space – they move and the objects in them move along. A coordinate system carries exactly and all of the information needed to place a rigid body in space. Thus the coordinate system, not the point, is the quintessential concept of location.

Hopefully you will not be surprised that, taking a geometric view, we represent the coordinate system axes as vectors and the location as a point. Formally, a coordinate system in 3D is three vectors and a point. In 2D, it is two vectors and a point. The vectors must be linearly independent. Collectively, they form a *basis* for the space – all vectors in the space can be expressed as a unique linear combination of these basis vectors.

Much of the literature uses the term *frame* instead of *coordinate system.* It is shorter, so we use it here too.

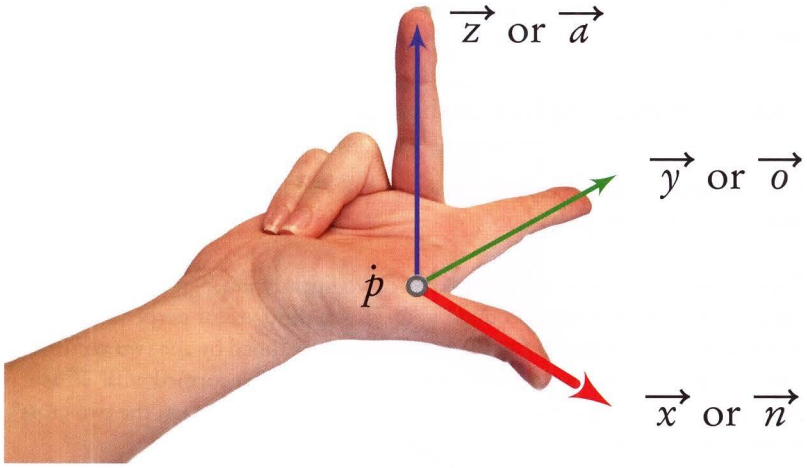


6.15: The x-, y- and z-axes of a frame have colours *red, green* and *blue* respectively. Point $\dot{p}$ locates the frame in space.

By putting constraints on a frame's vectors and point we can create special kinds of frames. For instance, a frame in which the vectors form a *natural basis*, that is, they are unit length and oriented to the principal global directions can be thought of as representing a simple translation of amount given by the frame's point.

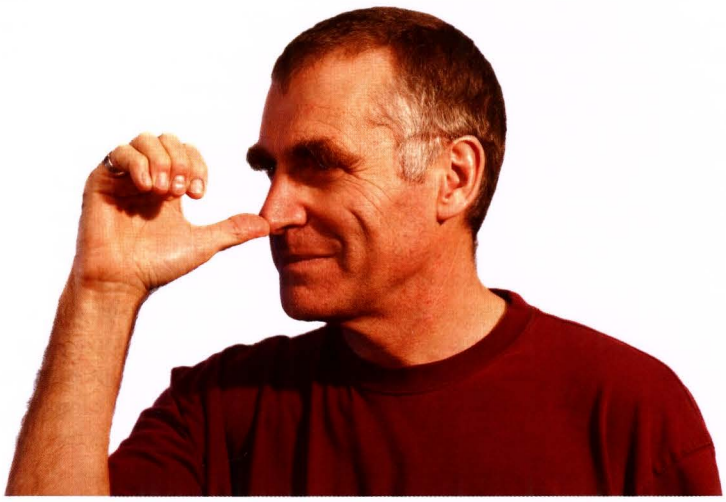
You might have noticed something. Phrases such as "principal global directions" imply that frames are relative to other frames. Geometrically, there is no master frame, no universal frame of reference. Practically, choosing a particular frame and relating all points to it creates an effective master frame for a particular computation.

By convention we consider only *right-handed frames*. We refer to the three frame vectors either as $\begin{bmatrix} \vec{x} & \vec{y} & \vec{z} \end{bmatrix}$ or as $\begin{bmatrix} \vec{n} & \vec{o} & \vec{a} \end{bmatrix}$. The former is in reference to the x-, y-, and z-axes of Euclidean space, the latter to the words $\vec{n}$*ormal,* $\vec{o}$*rientation* and $\vec{a}$*pproach.* To see the relevance of these words, extend your right hand in front of you with the index finger pointing at something, the thumb at right angles to the index finger and parallel to the line between your eyes, and the middle finger vertically at right angles to both index finger and thumb. You count the $\vec{x}$(or $\vec{n}$), $\vec{y}$(or $\vec{o}$) and $\vec{z}$(or $\vec{a}$) axes from thumb to middle finger: thumb = $\vec{x}$(or $\vec{n}$), index finger = $\vec{y}$(or $\vec{o}$) and middle finger = $\vec{z}$(or $\vec{a}$). The terms *normal*, *orientation* and *approach* relate to robotics where they are used to describe the position of a right-handed frame at the effector end of a robotic arm. Why have two ways of describing the axes? In some situations, x, y and z makes sense, for instance when you are indexing a named frame. Other times, for instance, when describing a frame's internal components, expressions such as $\vec{x}_x$ (the x-component of the $\vec{x}$ vector) are confusing and the $\vec{n}$, $\vec{o}$ and $\vec{a}$ notation is better.



6.16: The $\vec{x}$ *(*$\vec{n}$*ormal),* $\vec{y}$ *(*$\vec{o}$*rientation)* and $\vec{z}$ *(*$\vec{a}$*pproach)* vectors of a right-handed frame superimposed on a person's right hand.

A second convention takes a counterclockwise rotation about a coordinate axis to have a positive rotation angle if we look along the positive axis toward the coordinate origin. This is well-known as the *right-hand rule.* A way to stamp it indelibly into memory is to thumb your nose with your right hand (Go ahead! Do it! I do in Figure 6.17) and curl your fingers. If you are looking at the origin, your fingers show the direction of positive rotation. *Right-handed frames* and the *right-hand rule* cohere together well. You use the same hand to understand both.



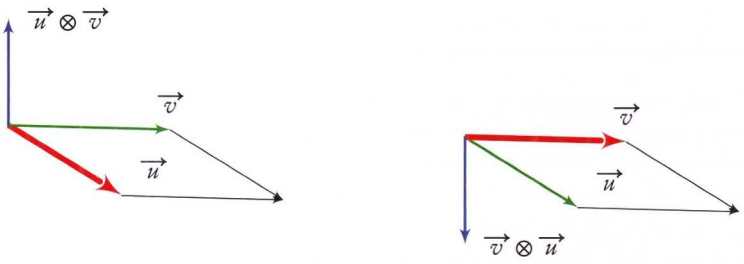
6.17: An unforgettable way of remembering the right-hand rule for rotation.

There are three really important things to know about frames: how to generate them, how to represent them and how to compose them.

6.5.1 Generating frames: the cross product

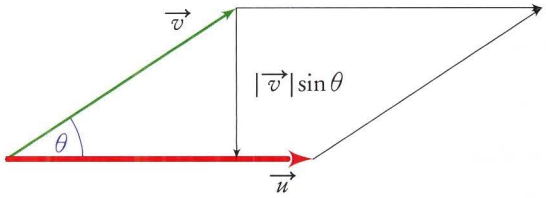
The *cross product* constructs one of two unique vectors given any two linearly independent vectors.

Taking a geometric view, let $\vec{u}$ and $\vec{v}$ be two linearly independent vectors. Figure 6.18 shows the cross product $\vec{u} \otimes \vec{v}$ as a third vector perpendicular to both. Its length is the area of the parallelogram spanned by the two vectors. The cross product forms the z-axis of a *right-handed frame* formed with $\vec{u}$ as the x-axis and $\vec{v}$ as the y-axis. Thus $\vec{u} \otimes \vec{v} \neq \vec{v} \otimes \vec{u}$. In fact, the two cross products are vector inverses $\vec{u} \otimes \vec{v} = -1\vec{v} \otimes \vec{u}$.



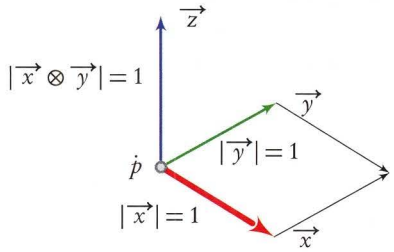
6.18: The cross product of two linearly independent vectors. Since the cross product always produces a right-handed frame, changing the order of the arguments to the cross product simply inverts the result vector: $\vec{u} \otimes \vec{v} = -1\vec{v} \otimes \vec{u}$.

The length of the cross product is the area of the parallelogram defined by $\vec{u}$ and $\vec{v}$. The area of a parallelogram is the base times the height. Taking $\vec{u}$ as the base, then basic trigonometry (Figure 6.19) shows that the height is $|\vec{v}|\sin\theta$ where θ is the angle between $\vec{u}$ and $\vec{v}$. Thus the area is $|\vec{u}||\vec{v}|\sin\theta$, and $|\vec{u}\otimes\vec{v}| = |\vec{u}||\vec{v}|\sin\theta$.



6.19: The area of a parallelogram.

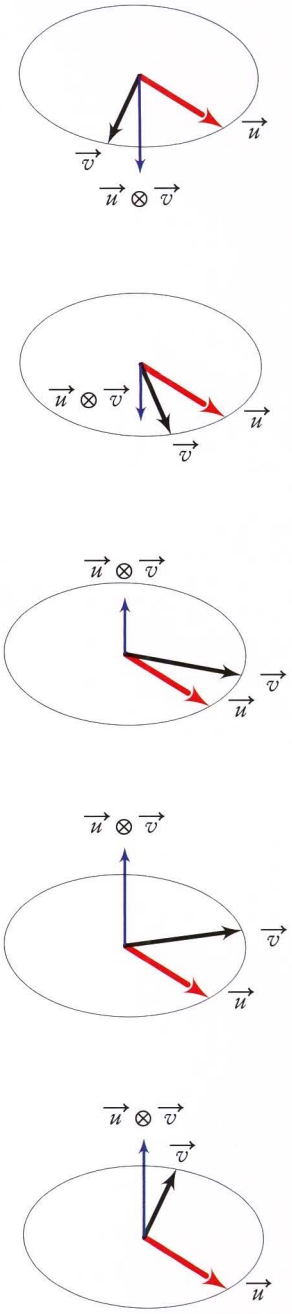
If the argument vectors $\vec{u}$ and $\vec{v}$ are unit-length, the cross product's length is the sine of the angle between $\vec{u}$ and $\vec{v}$. If the vectors are mutually perpendicular, then the parallelogram is a square with unit-length sides and area 1. The cross product vector thus has a length of 1. Frames having mutually orthogonal and unit length vectors are called *orthonormal*. If the vectors form a basis for the space they occupy, they are collectively an *orthonormal basis* for the space. Orthonormal bases have several nice mathematical properties (such as the scalar product of any arbitrary vector $\vec{u}$ with a basis vector being the length of the projection of the vector $\vec{u}$ onto the basis vector) that make them the main form for representing vector bases.



6.20: An orthonormal basis. Each axis is perpendicular to the others and is of unit length. The area of the parallelogram defined by the x- and y-axes is 1.

In terms of the components of $\vec{u}$ and $\vec{v}$, the cross product $\vec{w} = \vec{u}\otimes\vec{v}$ is given by the formulae

$$\begin{aligned}\vec{w}_x &= \vec{u}_y\vec{v}_z - \vec{u}_z\vec{v}_y\\ \vec{w}_y &= \vec{u}_z\vec{v}_x - \vec{u}_x\vec{v}_z\\ \vec{w}_z &= \vec{u}_x\vec{v}_y - \vec{u}_y\vec{v}_x\end{aligned} \qquad (6.6)$$



6.21: The length of the cross product of two unit vectors is the sine of the angle between them $|\vec{u}\otimes\vec{v}| = \sin\theta$. In this figure, both $\vec{u}$ and $\vec{v}$ are located on a circle of unit radius.

The cross product and plane equations

The cross product of the two vectors defining a plane is normal to the plane. This gives a way to compute the implicit plane equation from a point and two vectors (or from three points on the plane). Given a point $\dot{p}$ on the plane and two non-collinear vectors $\vec{u}$ and $\vec{v}$ in the plane, the implicit equation of the plane is as follows. Given

$$\vec{w} = \vec{u} \otimes \vec{v} \tag{6.7}$$

then the implicit equation is

$$\vec{w}_x \dot{p}_x + \vec{w}_y \dot{p}_y + \vec{w}_z \dot{p}_z - (\vec{w}_x \dot{p}_x + \vec{w}_y \dot{p}_y + \vec{w}_z \dot{p}_z) = 0$$

and the plane operator is

$$\begin{bmatrix} \vec{w}_x & \vec{w}_y & \vec{w}_z & -(\vec{w}_x \dot{p}_x + \vec{w}_y \dot{p}_y + \vec{w}_z \dot{p}_z) \end{bmatrix} \tag{6.8}$$

Using the cross product

It is extremely common to need a frame somewhere when defining a model. If the model is to be reused, or moved without restriction in space, such a frame should be *internal* to the model. If it is external, then the model can depend on its position, and sometimes in surprising ways. Whenever a model has three non-collinear points, or a single plane in parametric form, it is easy to construct such a frame. Vector bases (and therefore frames) are best when orthonormal – remember the discussion on page 109.

The process above produces a result equivalent to what is called the *Gram–Schmit orthonormalization process*. Given three linearly independent vectors $\vec{u}$, $\vec{v}$ and $\vec{w}$, the orthonormal frame with vectors $\vec{x}$, $\vec{y}$ and $\vec{z}$ is computed as follows:

$$\vec{x} = \frac{\vec{u}}{|\vec{u}|}$$

$$\vec{y}_{pre} = \vec{v} - (\vec{v} \bullet \vec{x})\vec{x}$$

$$\vec{y} = \frac{\vec{y}_{pre}}{|\vec{y}_{pre}|}$$

$$\vec{z}_{pre} = w - (\vec{w} \bullet \vec{x})\vec{x} - (\vec{w} \bullet \vec{y})\vec{y}$$

$$\vec{z} = \frac{\vec{z}_{pre}}{|\vec{z}_{pre}|}$$

6.5.2 Representing frames

Up until now, we have treated objects as if they are located in some universal space. While all geometry can be represented this way, modeling, mathematics and programming quickly become cumbersome and tedious. Frames provide an essential practical tool for organizing objects in space.

There are several modeling tasks that all require the ability to represent objects locally and to relate local representations to each other.

- We want to refer to things in differing frames of reference. A bicycle wheel is most easily described if we think of it as being located at the origin of some frame with the centreline of its axle coincident with one of the primary axes.
- We want to move things around. Positioning the bicycle wheel with respect to its frame can be achieved more easily by moving its frame than by moving all of its points. Rotating the bicycle wheel is simply done by rotating its frame.
- We want to be able to draw images of three-dimensional objects on a two-dimensional screen. This involves creating a sequence of frames: the world, the camera and the screen.

To represent frames is to go from the geometric idea of a frame as three vectors (a basis) and a point to a notation for representing vectors and vector bases as matrices. Representation of and operations on frames are largely a matter of structuring information to apply vector operations in appropriate ways. It is usually a very good idea to understand everything about frames by constructing representations from vectors on up. In other words, do not treat operations and transformations as black boxes – understand the ideas.

The first structuring step is to use a matrix to represent the vectors of a frame – its vector basis. Given that every vector basis over vectors of n elements has n vectors there exists a natural representation of a vector basis as an $n \times n$ matrix as shown for the two-dimensional case in Equation 6.9.

$$\left[\vec{u} \quad \vec{v} \right] = \left[\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right] = \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix} \tag{6.9}$$

By convention, the columns of the matrix are the basis vectors. Representing a basis as a matrix is thus simple: take each basis vector as the column of a matrix. For two dimensions the array is 2×2; for three dimensions it is 3×3.

The well-known identity matrix from linear algebra represents the natural basis.

$$\begin{bmatrix} \vec{x} & \vec{y} & \vec{z} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using the interpretation of the column vectors being the basis vectors simply read off the natural basis vectors.

The three vectors that represent a space give us everything but the location of the vector basis. Remember? A frame is three vectors and a point – its location. The location gets added as a fourth column in the matrix. A frame matrix thus has two components. The first records the vector basis of the frame, the second the point of origin. Using the $\vec{n}$, $\vec{o}$ and $\vec{a}$ notation for the frame vectors (so the x, y and z coordinates do not get mixed up with the vectors), this gives

$$\left[\begin{array}{ccc|c} \vec{n} & \vec{o} & \vec{a} & \dot{p} \end{array}\right] = \left[\begin{array}{ccc|c} \vec{n}_x & \vec{o}_x & \vec{a}_x & \dot{p}_x \\ \vec{n}_y & \vec{o}_y & \vec{a}_y & \dot{p}_y \\ \vec{n}_z & \vec{o}_z & \vec{a}_z & \dot{p}_z \end{array}\right]$$

where $\vec{n}_x$ represents the x coordinate of the n basis vector of the frame and $\dot{p}_x$ represents the x coordinate of the point of origin of the frame.

This matrix is not square, and square matrices are "nice" in the sense that they enter into algebraic formulae with less dimension checking, have determinants (a very useful number that describes matrix properties) and potentially inverses. To recover the useful property of squareness, we add a row to the bottom of the matrix.

We now distinguish vectors and points by augmenting their representation with a single number: 0 for vectors and 1 for points. Vectors become $\begin{bmatrix} x & y & z & 0 \end{bmatrix}^T$ and points become $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$.

The non-square matrix suffers this augmentation to become

$$\left[\begin{array}{ccc|c} \vec{n}_x & \vec{n}_y & \vec{n}_z & \dot{p}_x \\ \vec{o}_x & \vec{o}_y & \vec{o}_z & \dot{p}_y \\ \vec{a}_x & \vec{a}_y & \vec{a}_z & \dot{p}_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

The first three columns can be read off as vectors (signaled by the fourth row being 0) and the fourth column is a point (signaled by its fourth row being 1). The vectors and points actually mean something. In fact, they mean three things. A matrix holding them can be seen as either a *representation*, a *mapping operator* between frames, or a *transformation operator* that changes objects.

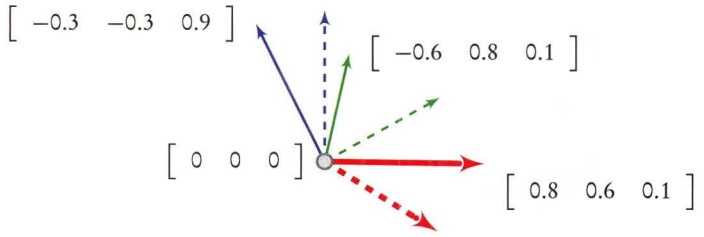
6.5.3 Matrices as representations

The first interpretation is that a matrix specifies a frame in terms of another frame. In this interpretation, the first three column vectors of a matrix are the vectors of its corresponding basis, written in terms of the vectors of some other frame. The fourth vector is the location of the frame, again written in terms of the other frame. This idea can be represented with perfect correspondence as both drawing and matrix.

If you have a drawing (or physical model) you can make a matrix representing a frame. The vectors of the frame are columns in the upper left block of the matrix. The frame location is the upper right block. If you have a matrix you can make a drawing of the corresponding frame. Simply use the columns of the upper left block matrix as the x-, y- and z- coordinates of the vectors in the drawing. Use the column in the upper right block as the frame location.

The relation between matrix and drawing is one of the very rare perfect correspondences between mathematics and diagrams. Usually drawings both remove information (they abstract) and introduce extraneous information. In this case even the fiction that the vectors are bound to a point has significance. Projecting a vector from the frame origin to a point $\dot{p}$ onto the frame vectors and dividing by the length of the frame vectors yields coordinates in the frame.

Frame:



Matrix:

$$\left[\begin{array}{ccc|c} 0.8 & -0.6 & -0.3 & 0 \\ 0.6 & 0.8 & -0.3 & 0 \\ 0.1 & 0.1 & 0.9 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.22: The correspondence between a frame drawing and its matrix is perfect and literal. The dashed frame is the reference in which the frame is drawn and in which the matrix is defined. (Careful reading of this frame shows that it is not perfectly orthonormal – here accuracy is sacrificed to gain short numbers.)

6.5.4 Matrices as mappings

Representing frames as matrices suggests matrix multiplication might be useful. Remember that points and vectors are column matrices so must appear *after* the matrix in any matrix multiplication (the basic rule of matrix multiplication conformality is a matrix of dimension $m \times n$ can only be multiplied by a matrix of dimension $n \times p$).

If T is a matrix representation of a frame, then

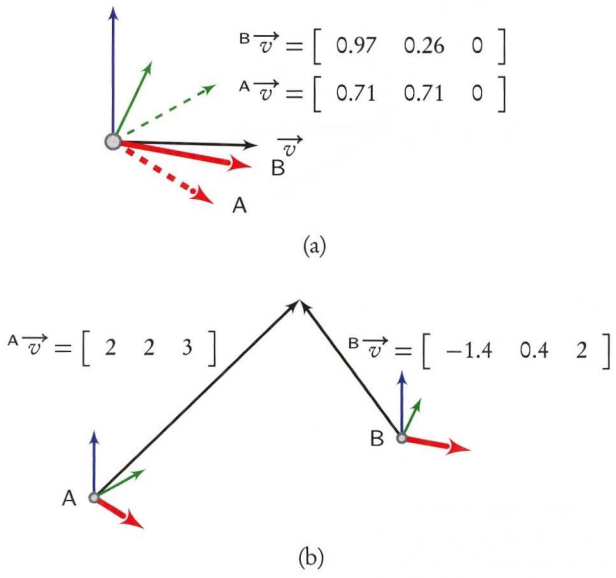
$$\vec{v}' = \mathsf{T}\vec{v}$$

means that some new vector $\vec{v}'$ is produced by the expression $\mathsf{T}\vec{v}$.

A useful interpretation is that $\vec{v}'$ represents a vector in a *reference frame* A, that $\vec{v}$ represents the same vector in a *represented frame* B and that T represents the represented frame in terms of the reference frame. A notation that makes this clear can really help. Write ${}^{\mathsf{A}}_{\mathsf{B}}\mathsf{T}$ to denote the represention of frame B in terms of the frame A and ${}^{\mathsf{A}}\vec{v}$ to denote the representation of vector $\vec{v}$ in terms of frame A. To complete the picture write ${}^{\mathsf{B}}\vec{v}$ as the representation of $\vec{v}$ in terms of the frame B.

$${}^{\mathsf{A}}\vec{v} = {}^{\mathsf{A}}_{\mathsf{B}}\mathsf{T}\,{}^{\mathsf{B}}\vec{v} \tag{6.10}$$

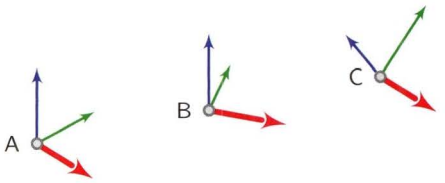
Figure 6.23 shows that both ${}^{\mathsf{A}}\vec{v}$ and ${}^{\mathsf{B}}\vec{v}$ represent the same geometric vector $\vec{v}$ – they differ numerically because they are representations of this vector in their respective frames.



6.23: The equation ${}^{\mathsf{A}}\vec{v} = {}^{\mathsf{A}}_{\mathsf{B}}\mathsf{T}\,{}^{\mathsf{B}}\vec{v}$ maps the geometric vector $\vec{v}$ from its representation as ${}^{\mathsf{B}}\vec{v}$ in frame B to its representation as ${}^{\mathsf{A}}\vec{v}$ in frame A. In (a) the two frames share an origin and differ by a z-axis rotation. In (b) the two frames also differ by a translation.

Geometrically this means that every point represented within a frame B has at least two sets of coordinates: those in frame B itself and those in the frame A that "holds" B. This explains the usual representation in a parametric modeling system in which a point has `X`, `Y` and `Z` properties as well as `XLocal`, `YLocal` and `ZLocal` properties [1]. Of course, every point has an implicit representation in terms of every frame specified in a model. Usually a system computes these only if needed.

A frame B that holds another frame C might itself be held in a third frame A.



The frame notation above conveys an advantage in reading chains of mappings. For example, the following chain

$$^{A}\overrightarrow{v} = {}^{A}_{B}T\,{}^{B}_{C}T\,{}^{C}_{D}T\,{}^{D}_{E}T\,{}^{E}_{F}T\,{}^{F}_{G}T\,{}^{G}\overrightarrow{v}$$

can be checked for consistency by "canceling" any mapping that represents a frame G if it is to the left of another mapping whose representation is written in terms of G.

$$^{A}\overrightarrow{v} = {}^{A}_{\not B}T\,{}^{\not B}_{\not C}T\,{}^{\not C}_{\not D}T\,{}^{\not D}_{\not E}T\,{}^{\not E}_{\not F}T\,{}^{\not F}_{\not G}T\,{}^{\not G}\overrightarrow{v}$$

Of course, the canceling is a notational trick with no mathematical significance in and of itself. It does allow us to check if a chain is well-formed. If cancellation does not work, we cannot multiply the matrices and expect a sensible result.

Equation 6.10 gives a second meaning to the matrix representation of a vector frame: as a mapping between frames. We use matrices as both a representation of frames and as a mapping between them.

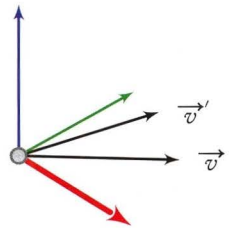
[1] The conventions on these names vary widely. For instance, global and local names might respectively be `X:XLocal`, `XGlobal:X`, `XGlobal:XLocal`, `X:XTranslation` or use another convention. You have to get used to the system you use.

6.5.5 Matrices as transformations

There is yet a third meaning to the matrix representation of a frame. This is as an operation that produces a new vector in the same frame as the original vector. We say that the new vector is a transformation of the old. The operation differs not in mathematical form, but in interpretation.

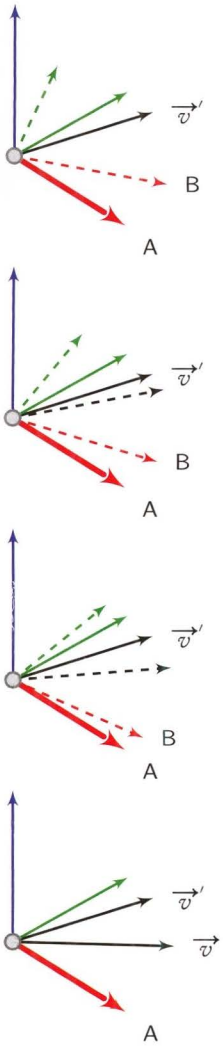
Write $\vec{v}' = \mathsf{T}\vec{v}$ to denote that both vectors are defined within a single frame. Figure 6.24 shows vectors $\vec{v}$ and $\vec{v}'$ as being on the xy-plane of a reference frame and $\vec{v}'$ being rotated about the z-axis by 30° from $\vec{v}$. The matrix gives just this 30° rotation about the z-axis.

$$\left[\begin{array}{ccc|c} \cos 30 & -\sin 30 & 0 & 0 \\ \sin 30 & \cos 30 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$



6.24: Vector $\vec{v}'$ is the result of rotating vector $\vec{v}$ around the z-axis with both vectors represented in the same frame.

Figure 6.25 shows a mental trick that helps in understanding transformation. Think about the mapping equation ${}^{\mathsf{A}}\vec{v} = {}^{\mathsf{A}}_{\mathsf{B}}\mathsf{T}\,{}^{\mathsf{B}}\vec{v}$ and then move frame B to coincide with frame A, bringing the vector $\vec{v}'$ along with it. When the frames coincide, that is B becomes A, the vector $\vec{v}$ will have moved into the correct position. In essence, transformation is the same as mapping, except that the original vector is assigned to frame A from the outset, not frame B.



6.25: When frame B rotates, it brings its representation of the vector with it. When it coincides with frame A, its vector coincides with vector $\vec{v}$ of the equation $\vec{v}' = \mathsf{T}\vec{v}$.

6.6 Geometrically significant vector bases

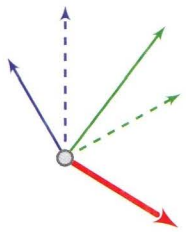
Typically, a frame is composed by expressing it as a sequence of more simple frames. For example, a rotation around the z-axis of a translated frame A in space can be thought of a translation that takes A to the global origin, then a rotation about the global origin and finally a translation back to the original

location of A. The simple frames involved in such compositions are *rotation, scaling, shearing* and *translation.*

The three primitive rotations are about the x-, y- and z-axes. Composition of rotation about a single axis is commutative. Composition of rotations about multiple axes is not commutative.

Rotation about the x-axis

Frame:

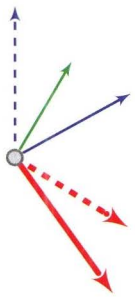


Matrix:

$$\mathrm{Rot}_x(\theta) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Rotation about the y-axis

Frame:

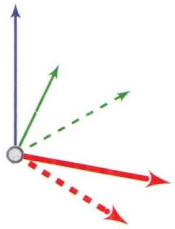


Matrix:

$$\mathrm{Rot}_y(\theta) = \left[\begin{array}{ccc|c} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Rotation about the z-axis

Frame:



Matrix:

$$\mathrm{Rot}_z(\theta) = \left[\begin{array}{ccc|c} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

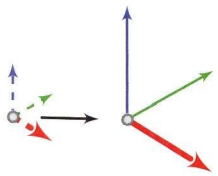
Why is rotation about the y-axis different in form from rotation about the x- or z-axis? Look at the figures and imagine looking down the respective axes. The

y-axis rotation presents a different geometry than the other two – the x- and z-axes are comparatively in a different order than the other cases.

Uniform scaling

Uniform scaling increases the "size" of a frame. Composition of uniform scaling is commutative.

Frame: **Matrix:**



$$\text{Scale}_{uniform}(\theta) = \left[\begin{array}{ccc|c} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.26: Uniform scaling of a frame B about the origin of another frame A. Note that the reference frame and the represented frame are not coincident in the drawing – visual coincidence would be confusing. No translation is implied.

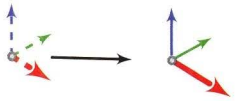
Scaling along one axis

Scaling along one axis is also known as *non-uniform scaling*. Such scaling on all three axes can be combined in a single matrix. Each of the axes is scaled by the corresponding factor. Composition of non-uniform scaling is commutative.

Composition of scaling in general is commutative.

$$\text{Scale}_{x,y,z}(s_x, s_y, s_z) = \left[\begin{array}{ccc|c} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Frame: **Matrix:**



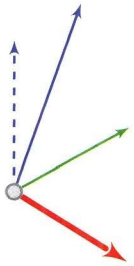
$$\text{Scale}_x(\theta) = \left[\begin{array}{ccc|c} s & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.27: Scaling of a frame B about the origin and along the x-axis of another frame A. Note that the reference frame and the represented frame are not coincident in the drawing – visual coincidence would be confusing. No translation is implied.

Shear

The shearing factor gives the amount of shear along the shearing coordinate per unit of distance from the origin along the sheared coordinate. This is the definition of the tangent of the angle between the old and new sheared axes.

Frame: **Matrix:**



$$\text{Shear}_{zy}(\alpha) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & \tan\alpha & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.28: Shearing of the z-axis along the y-axis of a frame B. The original z-axis is said to be the *sheared coordinate* and the original y-axis the *shearing coordinate*. The shearing factor is the tangent of the angle between the old and new z-axes.

There are six possible primitive shears, corresponding to the six zero values off the diagonal of the basis identity matrix.

$$\left[\begin{array}{ccc|c} 1 & Sh_{yx} & Sh_{zx} & 0 \\ Sh_{xy} & 1 & Sh_{zy} & 0 \\ Sh_{xz} & Sh_{yz} & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

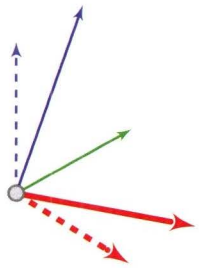
To model a primitive shear only one of these may be non-zero. To compose shears combine primitive shears with matrix multiplication. In general, the composition of primitive shears is not the simple setting of two values in a single shear matrix. Equation 6.11 (showing just the basis components of the frames) shows that the composition of a 45° zx shear with a 45° xz shear yields a matrix with non-unit values on the diagonal.

In general, composition of shears is not commutative.

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{6.11}$$

However, two axes can be sheared parallel to the third axis, and this combination is straightforward.

Frame: **Matrix:**



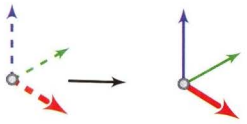
$$\text{Shear}_y(\alpha,\beta) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ \tan\alpha & 1 & \tan\beta & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.29: Shearing of the x- and z-axes along the y-axis of a frame B. The original x- and z-axes are said to be the *sheared coordinates* and the original y-axis the *shearing coordinate*. The shearing factors are the tangent of the angle between the old and new x- and z-axes.

Translation

Translation represents the position of one coordinate system with respect to another, by distances along the x-, y- and z-axes. Composing translations is commutative.

Frame: **Matrix:**



$$\text{Trans}(x,y,z) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.7 Composing vector bases

All of the primitive geometrically significant vector bases can be composed to model more complex geometry and there is a general mental technique for doing this. First, imagine that the geometry is located in a universal, global frame. Second, use a sequence of geometric operations to bring the geometry into alignment with the global frame. Third, model the appropriate geometry. Fourth, use the inverse of the sequence of geometric operations from the second step to restore the geometry to its original location with the modeled change.

Matrices representing vector bases compose from *RIGHT* to *LEFT*. Why is this so? Consider a series of matrices each implementing an operation on vectors as follows:

$$R_0\ R_1 \ldots R_{n-1}\ R_n$$

Any vector $\vec{v}$ that is operated on by this sequence is placed at the end of the sequence. Thus,

$$\vec{v}' = R_0 \, R_1 \ldots R_{n-1} \, R_n \vec{v}$$

Since matrix multiplication is associative, this can be rewritten as

$$\vec{v}' = (R_0 \, (R_1 \ldots (R_{n-1} \, (R_n \vec{v}))))$$

Each successive operation, reading from right to left, produces another vector that has been transformed with respect to the global frame that is implied by the operator view of matrix representation of vector bases. Thus, reading from right to left, apply operator R_n, then $R_{n-1} \ldots$ then R_1 then finally R_0.

Matrices representing vector bases compose from *RIGHT* to *LEFT*. Reading from the right, each successive transformation changes the relationship between itself and the frame to its left. Thinking about matrices as operations this has the effect of moving the frame and everything to its right. As you add bases on the left, each effects a movement with respect to the global origin.

6.7.1 Which comes first? Translation or rotation?

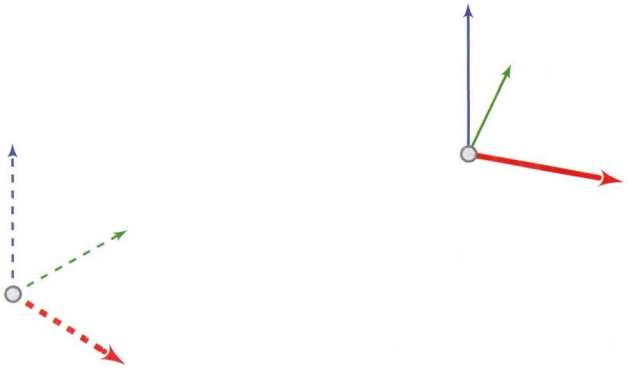
Frames combine three vectors and a point. Consider these two parts as motions; then a frame might combine, say, a rotation and a translation. Which comes first? The answer reveals itself in both geometric and matrix views.

First geometry. As in frame representation, a matrix is a direct representation of the frame it implements. So Figure 6.30 represents the frame given by the matrix

$$\left[\begin{array}{ccc|c} \cos 30 & -\sin 30 & 0 & 2 \\ \sin 30 & \cos 30 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Remember that successive matrix operations taken right-to-left model motions with respect to the global origin. Thus, if we want to think of the rotation and the translation acting separately, the frame must be generated by first a rotation of 30° about the z-axis, followed by a translation of $\left[\begin{array}{ccc} 2 & 2 & 1 \end{array}\right]^T$. The vector component acts first in a frame representation!

Frame:



Matrix:

$$\text{Trans}_{x,y,z}(\theta) = \left[\begin{array}{ccc|c} \cos 30 & -\sin 30 & 0 & 2 \\ \sin 30 & \cos 30 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.30: The frame represesenting a rotation of 30° about the z-axis followed by a translation of $[2\,2\,1]^T$.

A second view using matrix multiplication confirms this geometric insight.

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|c} \cos 30 & -\sin 30 & 0 & 0 \\ \sin 30 & \cos 30 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} \cos 30 & -\sin 30 & 0 & 2 \\ \sin 30 & \cos 30 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

whereas

$$\left[\begin{array}{ccc|c} \cos 30 & -\sin 30 & 0 & 0 \\ \sin 30 & \cos 30 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} \cos 30 & -\sin 30 & 0 & 2(\cos 30 - \sin 30) \\ \sin 30 & \cos 30 & 0 & 2(\sin 30 + \cos 30) \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

6.8 Intersections

The intersection of geometric primitives is a fundamental construct in many computer graphics and modeling applications. Its most simple form involves the linear elements *points* (1D), *lines* (2D) and *planes* (3D). In general, two elements may intersect (or not) as elements of equal or one lesser dimension than the lowest dimension of the elements involved. Thus two planes may intersect at a plane or a line (but not at a point), lines may intersect at a line or a point, and points may intersect at a point (this is point identity). With mixed elements, a line may intersect a plane at either a line or a point.

The number of intersection conditions is large, and the mathematics is, in cases, complex. Most parametric modeling systems provide a large set of intersection operators. Most of the time, these suffice. Note the phrase "most of the time". The rest of the time it helps to be able to reason through intersection problems. This section presents a small suite of intersection problems and their solutions. The intent is to demonstrate approaches to framing and solving some simple problems. In more complex cases it really helps to have a good text at hand, for example, Schneider and Eberly (2003).

Recall the available three-dimensional representations for points, lines and planes. Geometrically, a point is a point. A line is defined parametrically as a point and a vector (or alternatively two points). A plane has many definitions occurring in two families, one related to the normal vector (implicit, plane operator) and the other parametric to a point and two linearly independent vectors (point–vector and three–point).

There are several kinds of intersection-related questions.

- Determine if two objects intersect, without actually generating the intersection.
- Generate an object (point, line, plane) that lies on another object.
- Determine the object of intersection of two other objects, if it exists.
- Determine the kind of intersection (point, line or plane) at which two objects intersect.
- Determine the object that most closely joins two objects, for example, the line between a point and a line or the line between two lines.

All require similar thinking and similar mathematics to solve. The previous secions cover the needed mathematical basics. This section presents problems and discussion of how to go about solving each one. It takes a constructive approach, that is, it presents a solution as a series of steps, each geometric and visual. Such solutions are seldom the most efficient and often have cases in which they may not be as stable as possible. Indeed, when intersections (and

other problems) are professionally programmed in CAD systems, developers go to great lengths to ensure the code is robust, accurate and efficient. Amateur programmers do not and generally cannot program this way. They usually work by imagining a series of steps, each using simple constructs, that solve the immediate geometric problem.

6.8.1 Do two objects intersect?

Point on line ≡ point collinearity

Are three points $\dot{p}, \dot{q}$ and $\dot{r}$ collinear?
Are $\dot{p}(5,2,4)$, $\dot{q}(2,-4,1)$ and $\dot{r}(4,0,3)$ collinear?
Are $\dot{p}(5,2,4)$, $\dot{q}(2,-4,1)$ and $\dot{r}(3,-1,2)$ collinear?

Discussion. Three points are collinear if the triangle they form has zero area. The cross product is twice the area of the triangle defined by two vectors. If $|\overrightarrow{\dot{p}\dot{q}} \otimes \overrightarrow{\dot{p}\dot{r}}| = 0$, $\dot{p}, \dot{q}$ and $\dot{r}$ are collinear. In English, if the cross product produces the zero vector, the points are collinear.

Using Equation 6.6 on page 109

$$\overrightarrow{\dot{p}\dot{q}} \otimes \overrightarrow{\dot{p}\dot{r}} = \begin{bmatrix} \vec{u}_y \vec{v}_z - \vec{u}_z \vec{v}_y & \vec{u}_z \vec{v}_x - \vec{u}_x \vec{v}_z & \vec{u}_x \vec{v}_y - \vec{u}_y \vec{v}_x \end{bmatrix}^T$$

For $\dot{r}(4,0,3)$, $\vec{u} = \overrightarrow{\dot{p}\dot{q}} = \begin{bmatrix} 3 & 6 & 3 \end{bmatrix}^T$, $\vec{v} = \overrightarrow{\dot{p}\dot{r}} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T$

$$\overrightarrow{\dot{p}\dot{q}} \otimes \overrightarrow{\dot{p}\dot{r}} = \begin{bmatrix} 61-32 & 31-31 & 32-61 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

The points $\dot{p}, \dot{q}$ and $\dot{r}$ are collinear.

For $\dot{r}(3,-1,2)$, $\vec{u} = \overrightarrow{\dot{p}\dot{q}} = \begin{bmatrix} 3 & 6 & 3 \end{bmatrix}^T$, $\vec{v} = \overrightarrow{\dot{p}\dot{r}} = \begin{bmatrix} 2 & 3 & 2 \end{bmatrix}^T$

$$\overrightarrow{\dot{p}\dot{q}} \otimes \overrightarrow{\dot{p}\dot{r}} = \begin{bmatrix} 62-33 & 32-32 & 33-62 \end{bmatrix}^T = \begin{bmatrix} 3 & 0 & -3 \end{bmatrix}^T$$

The points $\dot{p}, \dot{q}$ and $\dot{r}$ are not collinear.

Point in plane

Does the plane $2x - 3y + z = 11$ contain the point $\dot{p}(1,-2,3)$?
The point $\dot{q}(5,-6,0)$?

Discussion. This is easily solved if the plane is represented in any form related to the implicit, as it is here. Simply plug the point values into the equation.

For $\dot{p}(1,-2,3)$, $2 \cdot 1 - 3 \cdot (-2) + 3 = 11$, so $\dot{p}$ is on the plane.

For $\dot{q}(5,-6,0)$, $2 \cdot 5 - 3 \cdot (-6) + 0 = 28$, so $\dot{q}$ is not on the plane.

Computationally, a result "close enough" to zero means that the point is on the plane. We gloss over what "close enough" means. Geometry uses real numbers, which are only approximately represented in computers. A simple threshold δ, where $-\delta < a < \delta \Longrightarrow a = 0$, suffices for most design applications.

Line in plane

Determine if a line lies in a plane.

Line through $\dot{p}_{start}(1,3,1)$

with direction vector $\overrightarrow{d} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$

Plane through $\dot{q}(0,1,0)$ with normal vector $\overrightarrow{n} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$

Discussion. It suffices here to check each of the end points of the line. These are $\dot{p}_{start}(1,3,1)$ and $\dot{p}_{end}(3,6,2)$. If they are both in the plane then the line is in the plane. A point is in a plane if its product with the plane's operator is zero.

The plane operator is $\gamma = \begin{bmatrix} \overrightarrow{n}^T & | & d \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & d \end{bmatrix}$. Since Q is in the plane $\gamma Q = 0$, therefore,

$$\begin{bmatrix} 1 & -1 & 1 & d \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = 0 \Longrightarrow d = 1$$

For $\dot{p}_{start}$

$$\begin{bmatrix} 1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 1 \\ 1 \end{bmatrix} = 0 \Longrightarrow \dot{p}_{start} \text{ is in the plane}$$