Lines in Space

Part 1: The 4D Cross Product

James F. Blinn

Microsoft Research

Computer graphicists see the world as a big pile of polynomials. Piles of linear polynomials (also known as vector and matrix products) represent flat things and straight things. To get curvy things you need higher-order polynomials. In my last few columns[1,2] I've played with such higher-order polynomials and their geometric interpretations in 1D and 2D projective spaces. Before trying this in 3D space it's a good idea to make sure we understand the simple linear case. So in the next couple of columns I'm going to look at homogeneous linear polynomials and their interpretation in projective 3D space. Geometrically, this means that I'll discuss 3D points, lines, and planes and their intersection and incidence relations. These columns will basically update the ideas from an old Siggraph paper[3] with the tensor diagram notation described in past issues of *IEEE Computer Graphics and Applications*.[1,4]

I'll start by reviewing the algebraic machinery and its geometric interpretation for the lower dimensional spaces. We'll begin in two dimensions, drop down briefly to one dimension, and then bound off to three dimensions. Along the way, I'll also share my thoughts about notational conventions for elements of vectors.

Two dimensions

We represent points in projective 2D space as 3D vectors $[x, y, w]$. In fact, we generally represent points in nD projective space by vectors in $(n + 1)$ dimensions. The vector components x, y, w are called homogeneous coordinates or, perhaps more properly, Grassman coordinates.[5] The ordinary 2D coordinates of a point are $[x/w, y/w]$; points with $w = 0$ represent points at infinity. Any nonzero scalar multiple of $[x, y, w]$ represents the same geometrical point.

We also represent lines as three-element vectors $[a, b, c]$ so that all points on the line satisfy

$$ax + by + cw = 0$$

If you're under 30 or attended a private school, you'd say that a point was a column vector and a line was a row vector and write

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} = 0$$

If you're over 30 and attended a public school, you'd say that a point was a row vector and a line was a column vector and write

$$\begin{bmatrix} x & y & w \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

But if you've been reading my last few columns, you'd know that row-ness and column-ness aren't important properties. Rather, we categorize point vectors as contravariant tensors and line vectors as covariant tensors. I'll visually distinguish between these by using subscripts to label the covariant tensor components and superscripts to label the contravariant tensor components:

$$\begin{bmatrix} x & y & w \end{bmatrix} = \begin{bmatrix} P^0 & P^1 & P^2 \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} L_0 & L_1 & L_2 \end{bmatrix}$$

(In these equations, we start indexing with zero.) I do realize that the indicated identification of points (or lines) with contravariant (or covariant) tensors labeled with superscript (or subscript) indices is only one of four possible notational permutations. But I did make some effort to examine standard mathematical and physics notation and tried to match up the choices properly.

A dot product (which we now call a tensor contraction) can only happen between a covariant and a contravariant index pair:

$$\begin{bmatrix} P^0 & P^1 & P^2 \end{bmatrix} \bullet \begin{bmatrix} L_0 & L_1 & L_2 \end{bmatrix} = 0$$

$$\sum_i P^i L_i = 0$$

Then we skip writing the summation sign explicitly and make it implicit for any pair of equal covariant and contravariant indices. (This convention is called Einstein index notation, or EIN). The EIN that point P lies on line L will be

$$P^i L_i = 0$$

Published by the IEEE Computer Society

When you achieve true enlightenment you learn to embrace contradictions. You realize that a superscript can either be a contravariant index or an exponent. And a subscript can either be a covariant index or a simple name extender. For example, we can name the components of the vector **L** two ways, which emphasize either the vector's name or the components' names:

$$\mathbf{L} = \begin{bmatrix} L_a \\ L_b \\ L_c \end{bmatrix} = \begin{bmatrix} a_L \\ b_L \\ c_L \end{bmatrix}$$

We're effectively referring to the vector elements on either a first- or last-name basis, expressing various degrees of familiarity.

Notation gallery

One of our goals is to gain insight into the patterns formed by algebraic combinations of tensor components as they represent various geometric situations. Sometimes one of the previously mentioned naming schemes will be more illustrative, and sometimes the other will be. Let's muse a bit on some of the possibilities, going from the familiar to the formal.

When there's only one point or one line in a problem, you can give the components simple unadorned names:

$$\mathbf{P} = \begin{bmatrix} x & y & w \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

If there's more than one point, we need more squiggles to distinguish between them. Let's look at the various ways we could do this. For comparison purposes, I'll show how a particular naming convention looks when writing out the first component of the vector cross product.

To start with, we could continue to emphasize the coordinate names by using them as the base symbol and use subscripts as name extenders to indicate which point we are talking about. For points **P** and **S** we would have

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} x_P & y_P & w_P \end{bmatrix} \\ \mathbf{S} &= \begin{bmatrix} x_S & y_S & w_S \end{bmatrix} \\ \mathbf{P} \times \mathbf{S} &= \begin{bmatrix} y_P w_S - w_P y_S & \dots & \dots \end{bmatrix} \end{aligned} \tag{1}$$

We could also apply subscripts as name extenders to the vector name itself. Rather than having subsubscripts, the components can just use this same name extender, which can be alphabetic:

$$\begin{aligned} \mathbf{P}_a &= \begin{bmatrix} x_a & y_a & w_a \end{bmatrix} \\ \mathbf{P}_b &= \begin{bmatrix} x_b & y_b & w_b \end{bmatrix} \\ \mathbf{P}_a \times \mathbf{P}_b &= \begin{bmatrix} y_a w_b - y_b w_a & \dots & \dots \end{bmatrix} \end{aligned}$$

Or the extenders can be numeric:

$$\begin{aligned} \mathbf{P}_1 &= \begin{bmatrix} x_1 & y_1 & w_1 \end{bmatrix} \\ \mathbf{P}_2 &= \begin{bmatrix} x_2 & y_2 & w_2 \end{bmatrix} \\ \mathbf{P}_1 \times \mathbf{P}_2 &= \begin{bmatrix} y_1 w_2 - w_1 y_2 & \dots & \dots \end{bmatrix} \end{aligned} \tag{2}$$

I think this is my favorite notation. Unfortunately, it doesn't generalize too well as we'll see in a minute.

Moving on to the last-name-first variants, we use the vector name as the base symbol for its components and label the positions with either letter names:

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} P^x & P^y & P^w \end{bmatrix} \\ \mathbf{S} &= \begin{bmatrix} S^x & S^y & S^w \end{bmatrix} \\ \mathbf{P} \times \mathbf{S} &= \begin{bmatrix} P^y S^w - P^w S^y & \dots & \dots \end{bmatrix} \end{aligned}$$

Or we label them with numbers:

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} P^0 & P^1 & P^2 \end{bmatrix} \\ \mathbf{S} &= \begin{bmatrix} S^0 & S^1 & S^2 \end{bmatrix} \\ \mathbf{P} \times \mathbf{S} &= \begin{bmatrix} P^1 S^2 - P^2 S^1 & \dots & \dots \end{bmatrix} \end{aligned} \tag{3}$$

Numerical indices can sometimes show relationship patterns better, but names for the elements x, y, z, and w are friendlier.

One further possibility is to label points and their components numerically. This would give us the following somewhat mixed metaphor where the superscript is a contravariant index and the subscript is a numeric name extender:

$$\begin{aligned} \mathbf{P}_1 &= \begin{bmatrix} P_1^{\,0} & P_1^{\,1} & P_1^{\,2} \end{bmatrix} \\ \mathbf{P}_2 &= \begin{bmatrix} P_2^{\,0} & P_2^{\,1} & P_2^{\,2} \end{bmatrix} \\ \mathbf{P}_1 \times \mathbf{P}_2 &= \begin{bmatrix} P_1^{\,1} P_2^{\,2} - P_1^{\,2} P_2^{\,1} & \dots & \dots \end{bmatrix} \end{aligned}$$

Mathematicians tend to like this form because it lets them generalize formulas by making the indices into algebraic expressions instead of specific numbers. I have reservations about it, though, for two reasons. First, it places all the information in the superscripts/subscripts, which are typographically small and harder to see. Second, this notation gets out of hand when applied to covariant (line-like) vectors. That is, the subscript must serve double duty as a component index and as a name extender, perhaps separated by a semicolon. Equation 2 shares this problem, which is why I don't use it here.

The choice of which notation to use should be made based on clarity. I'll typically start out with Equation 1 for a friendly introduction to some computations, and then proceed to Equation 3 for more generality. Sometimes I'll even resort to writing points as rows and lines as columns, when I think that it will help you transition to the new notation.

Because I want to shy away from using subscripts to distinguish different tensors, I'll use the following naming convention for my tensors:

- Point names will come from near the end of the alphabet: **P**, **S**, **T**, **U** (I want to reserve **Q** for something else, and I'm skipping **R** since it looks too much like **P**).
- Line names will come from the middle of the alphabet: **L**, **M**, **N**.
- Plane names will come from near the beginning of the alphabet: **D**, **E**, **G**, **H** (I won't use **F** since it looks too much like **E**).

The line through two points

Two points determine a line. You can calculate the components for the line **L** through the points **P** and **S** by expressing the condition that both points are on **L**.

$$\begin{bmatrix} x_P & y_P & w_P \\ x_S & y_S & w_S \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (4)$$

If you temporarily stop thinking of the vectors as 2D points in projective space and think of them as 3D vectors in Euclidean space, the question changes to how to find a vector perpendicular to two given vectors. We know that the answer is the cross product. The standard mnemonic for the cross product elements is as subdeterminants of the 2×3 matrix in Equation 4. Here is where our choice of component naming helps out visually. We are most interested in which columns of the 2×3 matrix to select for each 2×2 subdeterminant, and the column indicators x, y, and w are the largest and most visible parts of this notation. The anwer is:

$$\begin{aligned} a &= \det \begin{bmatrix} y_P & w_P \\ y_S & w_S \end{bmatrix} = y_P w_S - w_P y_s \\ b &= -\det \begin{bmatrix} x_P & w_P \\ x_S & w_S \end{bmatrix} = w_P x_S - x_P w_S \\ c &= \det \begin{bmatrix} x_P & y_P \\ x_S & y_S \end{bmatrix} = x_P y_S - y_P x_S \end{aligned} \qquad (5)$$

Let's transition to EIN. If we were to write it in Equation 3's notation, we'd have

$$\begin{bmatrix} P^0 & P^1 & P^2 \end{bmatrix} \times \begin{bmatrix} S^0 & S^1 & S^2 \end{bmatrix} = \begin{bmatrix} L_0 \\ L_1 \\ L_2 \end{bmatrix}$$

In the late 1800s, the mathematician Tullio Levi-Civita devised a way to express the cross product as a tensor contraction by using a magic three-index tensor typically named epsilon. We define the elements of the epsilon tensor as

$$\begin{aligned} &\varepsilon_{012} = \varepsilon_{120} = \varepsilon_{201} = +1 \\ &\varepsilon_{210} = \varepsilon_{012} = \varepsilon_{102} = -1 \\ &\qquad\qquad \varepsilon_{ijk} = 0 \quad \text{otherwise} \end{aligned} \qquad (6)$$

Writing epsilon out in its full glory gives a $3 \times 3 \times 3$ cube of numbers. The best we can do with conventional matrix notation is as a vector of matrices:

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

Then the EIN for the cross product of contravectors **P** and **S** is

$$P^i S^j \varepsilon_{ijk} = L_k \qquad (7)$$

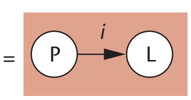
Remember that a summation is implied over indices i and j because they appear twice in the expression. Of course, most of the terms in this summation will be zero since most of the elements of the epsilon are zero. Let's take a look at the summation that generates $L_0 = a$. Of all the nine terms for $i = 0$, 1, and 2 and $j = 0$, 1, and 2, the only nonzero ones are

$$\begin{aligned} L_0 &= P^i S^j \varepsilon_{ij0} \\ &= P^1 S^2 \varepsilon_{120} + P^2 S^1 \varepsilon_{210} \\ &= P^1 S^2 - P^2 S^1 \\ a &= y_P w_s - w_P y_s \end{aligned}$$

This gives the expression for the first component of a cross product, as Equations 1 and 3 show.

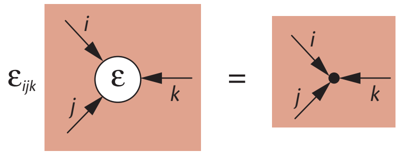
Tensor diagrams

Tensor diagram notation is simply another way of writing EIN. A tensor becomes a node and a summed-over index becomes a directed arc between nodes. The arc connects from the contravariant (point-like) index to the covariant (line-like) index. For example, the dot product of **P** and **L** would be

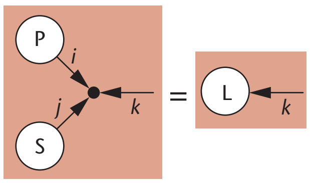
$P^i L_i =$ 

In more complicated diagrams, I'll skip labeling the arcs with index names. They're really, after all, just local variables for the summation and could be any unique temporary variable name.

Because the epsilon is a three-index tensor, its diagram node would have three arcs, and because it's covariant, the three arcs point inward. This is so common that we further abbreviate it by using just a small black dot:



The diagram form of the cross product, giving the line through two points, is then

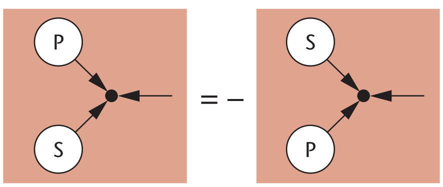


(8)

Note that indices i, j, k proceed counterclockwise around the epsilon. This takes account of the fact that a cross product has the property

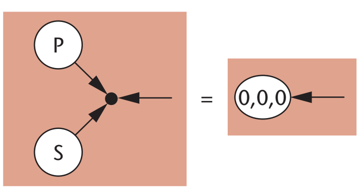
$$\mathbf{P} \times \mathbf{S} = -\mathbf{S} \times \mathbf{P}$$

This, in diagram form, means that a mirror reflection of a diagram containing an epsilon will flip the algebraic sign.



Two coincident points

The one situation where this algorithm won't work is if you feed it two vectors that actually represent the same geometrical point. That is, one is a homogeneous scale of the other. What you get back is a vector of all zeroes (a 3D vector crossed with itself gives the zero vector). We can turn this around and use Equation 9 to test the homogeneous equivalence of points **P** and **S**:



(9)

Three collinear points

The condition that a point **T** lies on the line formed from **P** and **S** is another way of saying that points **P**, **S**, and **T** are collinear. Given the technique noted previously for representing line **PS**, the test for collinearity is simply

$$(\mathbf{P} \times \mathbf{S}) \bullet \mathbf{T} = 0$$

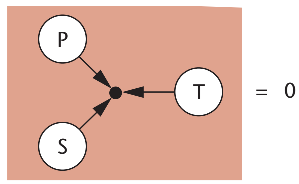
This is called the scalar triple product of **P**, **S**, and **T**. We can actually write this in several ways, which, somewhat obscures the symmetry of the relation with respect to permutations of **P**, **S**, and **T**:

$$\mathbf{P} \times \mathbf{S} \bullet \mathbf{T} = \mathbf{S} \times \mathbf{T} \bullet \mathbf{P} = \mathbf{T} \times \mathbf{P} \bullet \mathbf{S} = 0$$

The EIN form and its tensor diagram represent this symmetry more prettily, giving equal attention to all three vectors. They just involve plugging **T** into Equations 7 and 8 as follows:

$$P^i S^j T^k \varepsilon_{ijk} = 0$$

and



(10)

Another way to think of this is to say that the three vectors are linearly dependent. If you then stack them on top of each other to make a 3 × 3 matrix, the determinant of that matrix will be zero. The diagram of Equation 10 thus gives the determinant of the matrix **PST**.

The point on two lines

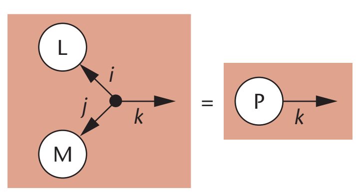
The principle of duality lets us turn the statements and equations discussed in the last section around and generate the analogous computation for finding the intersection of two lines. The coordinates of the point **P** common to two lines **M** and **N** again come from the cross product

$$\begin{bmatrix} L_0 \\ L_1 \\ L_2 \end{bmatrix} \times \begin{bmatrix} M_0 \\ M_1 \\ M_2 \end{bmatrix} = \begin{bmatrix} P^0 & P^1 & P^2 \end{bmatrix}$$

To write this in EIN you do the double summation using a contravariant version of the epsilon (same numbers, but with superscript indices):

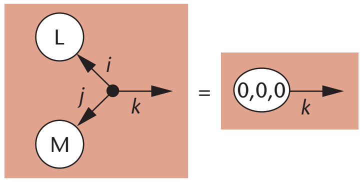
$$L_i M_j \varepsilon^{ijk} = P^k$$

The diagram for the contravariant epsilon is, again, a black dot but with outward pointing arrows:



Two coincident lines

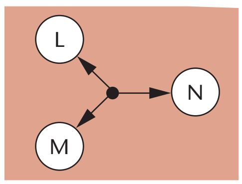
If lines **L** and **M** coincide, this cross product will produce a zero vector:



Three copointar lines

If three lines meet at one common point, then the scalar triple product of their vectors is zero. Various algebraic ways to detect this would be

$$\mathbf{L}\times\mathbf{M}\bullet\mathbf{N}=\mathbf{M}\times\mathbf{N}\bullet\mathbf{L}=\mathbf{N}\times\mathbf{L}\bullet\mathbf{M}=0$$
$$L_iM_jN_k\varepsilon^{ijk}=0$$



= 0

One dimension

Dropping down a dimension, we enter the world of homogeneous bivariate polynomials. Because we're only dealing with linear things this time, I only use the ridiculously simple linear equation

$$ax+bw=0$$
$$\begin{bmatrix}x & w\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix}=0$$

I like to think of the parameter pair $[x, w]$ as a 2D homogeneous coordinate of points on the 1D projective number line. This number line consists of all normal points at locations x/w, with the addition of a parameter at infinity at $[x, w] = [1, 0]$.

One point determines a line

Given one such 1D point, the line (that is, the linear polynomial) that it satisfies is just

$$\begin{bmatrix}a\\ b\end{bmatrix}=\begin{bmatrix}-w\\ x\end{bmatrix}$$

We can generate this with a matrix multiplication by inventing a 2D specialization of the 3D Levi-Civita epsilon:

$$\begin{bmatrix}x & w\end{bmatrix}\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}=\begin{bmatrix}-w & x\end{bmatrix}$$

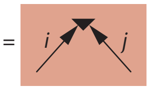
This 2×2 antisymmetric matrix is our epsilon:

$$\varepsilon_{ij}=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$$

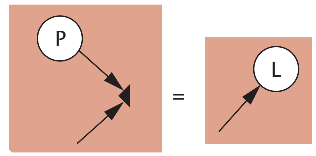
We can see that its definition is analogous to Equation 6:

$$\varepsilon_{01}=+1$$
$$\varepsilon_{10}=-1$$
$$\varepsilon_{ij}=0 \text{ otherwise}$$

For the diagram version of the epsilon I use

$\varepsilon_{ij}=$ 

The 2D specialization of Equation 8 is then



Two coincident points

Two vectors, $[x_p\ w_p]$ and $[x_s\ w_s]$, represent the same point on the 1D projective number line if

$$\frac{x_P}{w_P}=\frac{x_S}{w_S}$$

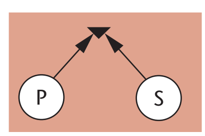
Of course, this isn't the best expression of this equivalency test because it dies horribly for parameters at infinity (where $w = 0$). A better test is the homogeneous equivalent:

$$x_pw_s-w_px_s=0$$

This is a specialization of the coincidence condition on two 2D points—that is, a specialization of the 3D cross product. We can write this as a matrix product with the epsilon:

$$\begin{bmatrix}x_P & w_P\end{bmatrix}\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\begin{bmatrix}x_S\\ w_S\end{bmatrix}=0$$

So, the condition that two 1D points are equivalent (the 2D analog to Equation 9) is

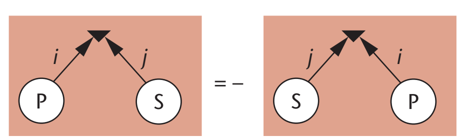


= 0

The definition of the epsilon implies that swapping its indices flips its sign:

$$P^i\varepsilon_{ij}S^j=-S^j\varepsilon_{ji}P^i$$

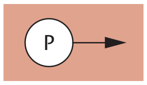
This is reflected in the following diagram by noting that a mirror reflection flips its sign, while a rotation doesn't:



Three dimensions

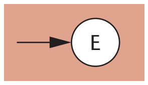
Now that we have the pattern, let's generalize the concepts from one and two dimensions into three dimensions. We represent 3D points by 4D contravariant vectors:

$$\mathrm{P} = [xyzw] = [P^0 P^1 P^2 P^3] =$$



And we represent 3D planes by 4D covariant vectors:

$$\mathrm{E} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = [E_0 E_1 E_2 E_3] =$$



Three points make a plane

Three points determine a plane. To generate plane **E** common to three points **P**, **S**, and **T**, we solve (by analogy to Equation 4) for a, b, c, and d in

$$\begin{bmatrix} x_P & y_P & z_P & w_P \\ x_S & y_S & z_S & w_S \\ x_T & y_T & z_T & w_T \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

We can think of the solution as a 4D generalization of the 3D cross product. Again, by analogy to Equation 5, the answer comes from the four 3×3 subdeterminants of the previous 3×4 matrix.

$$a = \det \begin{bmatrix} y_P & z_P & w_P \\ y_S & z_S & w_S \\ y_T & z_T & w_T \end{bmatrix}$$
$$b = -\det \begin{bmatrix} x_P & z_P & w_P \\ x_S & z_S & w_S \\ x_T & z_T & w_T \end{bmatrix}$$
$$c = \det \begin{bmatrix} x_P & y_P & w_P \\ x_S & y_S & w_S \\ x_T & y_T & w_T \end{bmatrix}$$
$$d = -\det \begin{bmatrix} x_P & y_P & z_P \\ x_S & y_S & z_S \\ x_T & y_T & z_T \end{bmatrix} \quad (11)$$

Let's write this in EIN. We first turn the previous equation into the tensor-friendly notation:

$$E_0 = \det \begin{bmatrix} P^1 & P^2 & P^3 \\ S^1 & S^2 & S^3 \\ T^1 & T^2 & T^3 \end{bmatrix}$$
$$E_1 = -\det \begin{bmatrix} P^0 & P^2 & P^3 \\ S^0 & S^2 & S^3 \\ T^0 & T^2 & T^3 \end{bmatrix} \quad (12)$$
$$E_2 = \text{similar}, E_3 = \text{similar}$$

Then we define a 4D generalization of the epsilon tensor. This will be a four-index gadget: a $4 \times 4 \times 4 \times 4$ element array with elements defined by

$\varepsilon_{ijkl} = +1$ if $ijkl$ is an even permuation of 0123
$\varepsilon_{ijkl} = -1$ if $ijkl$ is an odd permuation of 0123
$\varepsilon_{ijkl} = 0$ otherwise

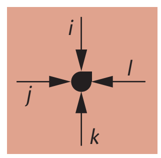
So we have plane **E** generated by three points **P**, **S**, and **T** in EIN as

$$P^i S^j T^k \varepsilon_{ijkl} = E_l$$

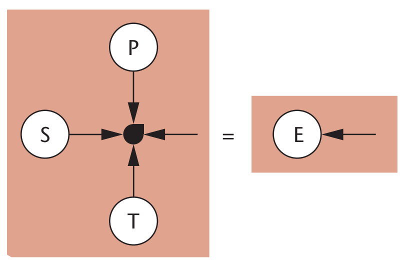
For a bit of intuition, notice that each term in this implied summation contains one component from each of **P**, **S**, and **T**. And the indices must be unequal or the epsilon factor will be zero. Now look at Equation 12 and think of the standard algorithm for calculating the determinant of a 3×3 matrix.

In diagram form, we represent the epsilon as a spot with four inward arrows. (In my next column, I'll comment on the shape of the spot.)

$$\varepsilon_{ijkl} =$$

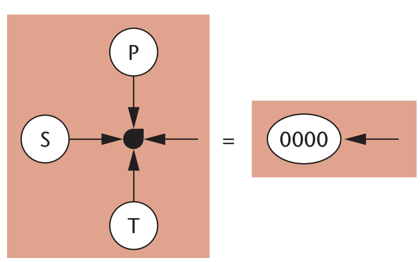


So the diagram form for the plane **E** through points **P**, **S**, and **T** is



Detecting three collinear points

If we feed three collinear points into this calculation we won't get a reasonable plane as a result. Sure enough, the components of plane $[a\ b\ c\ d]$ will all be zero:

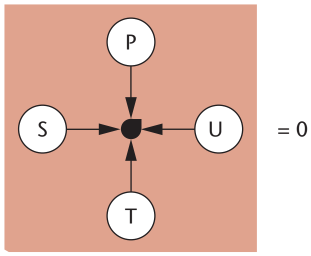


(13)

This is the 3D analog to Equation 10, which represents collinearity in 2D. Note that in 2D it took one value equaling zero to mean collinear. In 3D, it takes four values equaling zero to determine collinearity.

Detecting four coplanar points

Four points will be coplanar (again by analogy to Equation 10) if their scalar quadruple product is zero. This is just the determinant of the 4 × 4 matrix formed by stacking the four point vectors on top of each other. The diagram for this is



The point common to three planes

Now let's list the dual statements in the last three sections. We simply swap the terms point and plane and swap the covariant and contravariant indices.

First, given three planes **E**, **G**, and **H**, the point **P** common to them must satisfy

$$\begin{bmatrix} x & y & z & w \end{bmatrix} \begin{bmatrix} a_E & a_G & a_H \\ b_E & b_G & b_H \\ c_E & c_G & c_H \\ d_E & d_G & d_H \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

By analogy to Equation 11, the solution is

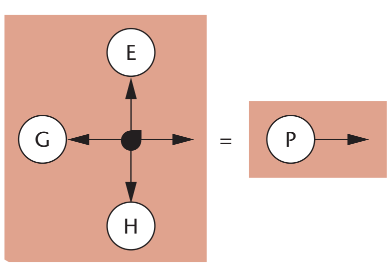
$$x = \det \begin{bmatrix} b_E & b_G & b_H \\ c_E & c_G & c_H \\ d_E & d_G & d_H \end{bmatrix}$$

$$y = -\det \begin{bmatrix} a_E & a_G & a_H \\ c_E & c_G & c_H \\ d_E & d_G & d_H \end{bmatrix}$$

$$z = \det \begin{bmatrix} a_E & a_G & a_H \\ b_E & b_G & b_H \\ d_E & d_G & d_H \end{bmatrix}$$

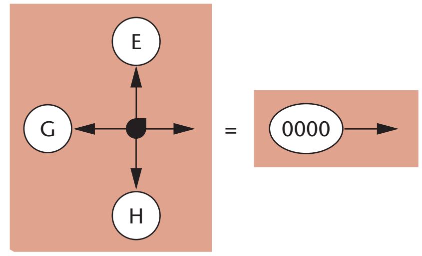
$$w = -\det \begin{bmatrix} a_E & a_G & a_H \\ b_E & b_G & b_H \\ c_E & c_G & c_H \end{bmatrix}$$

In diagram form this is



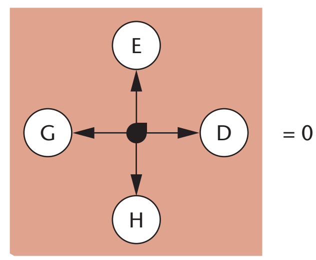
Three collinear planes

If the three planes intersect along one common line, finding the single point common to them will result in four zeroes. By analogy to Equation 13 we have



Detecting four copointar planes

Intersecting four planes will generally generate four points. In the special case where they all meet at one point (by analogy to Equation 10), their scalar quadruple product is zero. This is just the determinant of the 4 × 4 matrix formed by placing the four plane vectors next to each other. The diagram for this is



Lines in space

Now we come to the point of this series of columns. We have an algebraic representation for points in 3D (a four-element contravariant vector) and for planes in 3D (a four-element covariant vector). All the typical geometric operations correspond to various tensor contractions between these vectors and the epsilon tensor. What, then, is a reasonable algebraic representation for lines in 3D? We want some other tensor-like object that will generate tensor-contraction-like answers to all the usual geometric questions.

Table 1 lists some of the geometric questions we might want to answer. Each question has a dual nature (formed by swapping the terms point and plane), so I'll write them in pairs. Then I'll write the desired calculation as an overload function named ε to preview our ultimate answer. The list appears in Table 1. The answer will turn out to be fairly straightforward, but there are some subtleties and surprises. We'll see them all next time. ■

References

1. J.F. Blinn, "Quartic Discriminants and Tensor Invariants," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, Mar./Apr. 2002, pp. 86-91.
2. J.F. Blinn, "Visualize Whirled 2 × 2 Matrices," *IEEE Computer Graphics and Applications*, July/Aug. 2002, vol. 22, no. 4, pp. 98-102.

Table 1. Interesting questions about lines.

Questions	Calculations	Questions	Calculations
Are two points coincident?	$\varepsilon(\text{point}_1, \text{point}_2) = 0$	Are two planes coincident?	$\varepsilon(\text{plane}_1, \text{plane}_2) = 0$
If not, find the line through the two points.	$\varepsilon(\text{point}_1, \text{point}_2) = \text{line}$	If not, find the line at the intersection of the two planes.	$\varepsilon(\text{plane}_1, \text{plane}_2) = \text{line}$
Is a given point on a given line?	$\varepsilon(\text{point}, \text{line}) = 0$	Does a given plane contain a given line?	$\varepsilon(\text{plane}, \text{line}) = 0$
If not, what is the plane containing them both?	$\varepsilon(\text{point}, \text{line}) = \text{plane}$	If not, at what point does the line intersect the plane?	$\varepsilon(\text{plane}, \text{line}) = \text{point}$

Questions	Calculations
Do two lines intersect (or are they skew)?	$\varepsilon(\text{line}_1, \text{line}_2) = 0$
If they intersect, what is the point of intersection and the plane containing them both?	$\varepsilon(\text{line}_1, \text{line}_2) = \text{point}$ $\varepsilon(\text{line}_1, \text{line}_2) = \text{plane}$

3. J.F. Blinn, "A Homogeneous Formulation for Lines in 3 Space," *Computer Graphics* (Proc. Siggraph 77), vol. 11, no. 2, 1977, p. 237.
4. J.F. Blinn, "Polynomial Discriminants—Part 2: Tensor Diagrams," *IEEE Computer Graphics and Applications*, vol. 21, no. 1, Jan./Feb. 2001, pp. 86-92.
5. R. Goldman, "On the Algebraic and Geometric Foundations of Computer Graphics," *ACM Trans. Graphics* (TOG), vol. 21, no. 1, Jan. 2002.

Readers may contact Jim Blinn by email at blinn@microsoft.com.