Vectors for Beginners

Leo Dorst

September 6, 2007

1 Three ways of looking at linear algebra

We will always try to look at what we do in linear algebra at three levels:

- *geometric*: drawing a picture. This is often related to the application.
- *computational*: giving a formula. This is related to the implementation (in Matlab or other code).
- *symbolic*: specifying the essence of the computation, using the language elements of linear algebra. Vectors and hyperplanes are the words, products and matrices (we'll meet those later) are the verbs.

To use linear algebra effectively in computer science, you have to try and encode the geometric pictures of your application, into the symbolic level; the computational level then follows automatically. Thinking at the computational level is tiresome, and will make linear algebra appear like a bag of tricks to memorize, rather than as a language to master fluently.

So the order in the practice of computer science is typically: geometry, symbolic, computation. Often the order of definition in the Bretscher book will be: geometry, computation, symbolic; or even computation, symbolic, geometry.

2 Vector properties

What are vectors?

- *geometric*: For now, vectors are like arrows starting from a common point (the origin) in an n-dimensional space. In our examples we use 2-D and 3-D. A vector has a direction, and a length. That length is called its *norm*.
- *symbolic*: We will denote a vector as a variable with an arrow overhead: the vector $\vec{x}$. The norm is written as $\|\vec{x}\|$. We can only specify the direction relative to standard vectors. In n dimensions, you need n standard vectors, called a *basis*. We will denote those standard vectors as $\vec{e}_1, \cdots \vec{e}_n$. We often choose them perpendicular and of unit length; then they point in the direction of the coordinate axes that in high school you used to call

the x-axis, y-axis, et cetera. Since we run out of the alphabet soon, and because programs typically run over the indices of the dimensions, we will use $\vec{e}_i$ instead, with i running from 1 to n.

- *computational*: You can think of an n-D vector as n numbers. These numbers give the components relative to the standard vectors. Adding these components (see below) gives the vector. It is traditional to write the n numbers in a column, with square brackets, and label then with the basis vector they refer to. So

$$\vec{x} \mapsto \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{1}$$

  is the computational representation of the 3-dimensional vector $\vec{x}$ in the basis $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$.

You see that the geometry and the computational representation have to be specific on the dimensionality, whereas the symbolic representation need not. That is an advantage, for we are going to need spaces of many different dimensionalities in computer science.

3 Basic operations on vectors: scaling and adding

3.1 Scaling

You can change the length of a vector, making it λ times longer. Here λ is a real number, allowed to be zero or negative.

- *geometric*: Multiplying the vector $\vec{x}$ by λ makes the vector λ times as long. The resulting vector still points in the same direction is $\lambda > 0$, in the opposite direction if $\lambda < 0$, and has lost its direction when $\lambda = 0$ (it then gives the null vector: no direction, no length, so hardly a vector at all).

- *symbolic*: We denote the product of a scalar (number) λ and a vector $\vec{x}$ as $\lambda\,\vec{x}$.

- *computational*: The computational representation of $\lambda\,\vec{x}$ is obtained by multiplying each of its components by λ:

$$\lambda\,\vec{x} \mapsto \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \\ \lambda x_3 \end{bmatrix}. \tag{2}$$

  That reduces the implementation of vector scaling to the multiplication of real numbers.

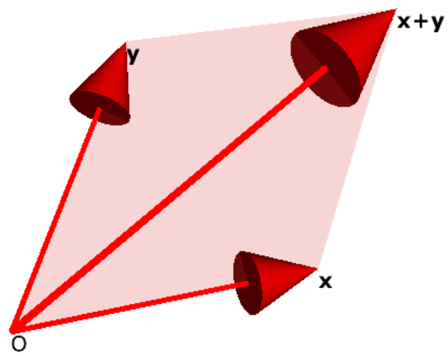


Figure 1: Addition of the vectors $\vec{x}$ and $\vec{y}$ to form the vector $\vec{x}+\vec{y}$. (In these figures, we label the vectors with bold font, rather than overhead arrows.)

You should now be able to prove:

$$\textit{distributivity:} \quad (\lambda+\mu)\,\vec{x} \quad = \quad \lambda\,\vec{x}+\mu\,\vec{x} \tag{3}$$

3.2 Addition

You can also add two vectors $\vec{x}$ and $\vec{y}$ to make a new vector which is called their *sum*.

- *geometric*: Vector addition is done by completing the parallelogram of which $\vec{x}$ and $\vec{y}$ are the sides, and using the diagonal from the origin as the resulting sum vector (see Figure 1).
- *symbolic*: We denote the sum as $\vec{x}+\vec{y}$.
- *computational*: We compute the sum as:

$$\vec{x}+\vec{y} \mapsto \begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ x_3+y_3 \end{bmatrix}. \tag{4}$$

  That reduces the vector addition to n additions of real numbers.

Since the computational definition defines the sum of vectors, it can be used to prove the following symbolic properties:

$$\begin{array}{llcl} \textit{commutativity:} & \vec{x}+\vec{y} & = & \vec{y}+\vec{x} \\ \textit{associativity:} & (\vec{x}+\vec{y})+\vec{z} & = & \vec{x}+(\vec{y}+\vec{z}) \end{array} \tag{5}$$

Prove this for yourself, and also draw the geometry of these properties.

The scaling of a vector behaves nicely relative to the addition:

$$\textit{distributivity:} \quad \lambda\,(\vec{x}+\vec{y}) \quad = \quad \lambda\,\vec{x}+\lambda\,\vec{y} \tag{6}$$

Prove this for yourself.

3.3 The meaning of coordinates

Now that we have scaling and addition, we can specify the connection between the coordinates and the symbolic notation, through the the basis vectors. It is:

$$\vec{x} = x_1\,\vec{e}_1 + x_2\,\vec{e}_2 + x_2\,\vec{e}_2 \tag{7}$$

3.4 Linearity

In linear algebra, we are going to be interested in operations on vectors. For example, rotating a vector, to point in another direction. For the moment, we denote the rotation of a vector $\vec{x}$ as $R(\vec{x})$. We are not going to define it computationally yet, but you may check geometrically that this rotation operation has the properties:

$$\begin{array}{lrcl} \textit{scaling:} & R(\lambda\,\vec{x}) & = & \lambda\,R(\vec{x}) \\ \textit{addition:} & R(\vec{x}+\vec{y}) & = & R(\vec{x})+R(\vec{y}) \end{array} \tag{8}$$

(Draw the pictures associated with these identities!)

These are called *the linearity properties*, and an operation with these properties is called a *linear operation* (or linear mapping, or linear transformation). Linear algebra is all about these kinds of operations, and gives handy symbolic theory and computational techniques for them.

Linear operations occur a lot in practice, and that makes linear algebra very useful. In fact, it is so useful that if an operation is not linear, we try to *linearize* it, i.e., make it linear. Sometimes we succeed exactly, by a clever trick of choosing our vectors; sometimes we can only linearize approximately. But linearization enormously extends the usefulness of linear algebra. Because this is such a powerful approach to all sorts of problems (not only obviously geometrical ones!), we teach linear algebra in computer science.

When you have a hammer, everything looks like a nail.

4 A product for vectors

There is a product for vectors, called the *dot product* (or *inner product*), which produces a scalar number from two vectors. Since this is new, we go the opposite direction, from computation to geometry. We define it first, and then try to interpret the meaning.

- *computational*: The dot product of two 3-D vectors $\vec{x}$ and $\vec{y}$ is defined as:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3. \tag{9}$$

  It is a number. In linear algebra, a number is called a *scalar*, since it has only a scale, not a direction (in contrast to vectors). For the dot product of vectors in n-D, just add more terms to this definition.

- *symbolic*: From the computational definition, you can prove for yourself the following properties at the symbolic level:

$$\begin{array}{lrcl} \textit{symmetry:} & \vec{x} \cdot \vec{y} & = & \vec{y} \cdot \vec{x} \\ \textit{scaling:} & (\lambda\, \vec{x}) \cdot \vec{y} & = & \lambda\, (\vec{x} \cdot \vec{y}) \\ \textit{distributivity:} & (\vec{x} + \vec{y}) \cdot \vec{z} & = & (\vec{x} \cdot \vec{z}) + (\vec{y} \cdot \vec{z}) \end{array} \tag{10}$$

  The dot product therefore behaves very much like multiplication of numbers.

  However, it is not associative. We cannot even express associativity, for in $(\vec{x} \cdot \vec{y}) \cdot \vec{z}$ is not defined: $(\vec{x} \cdot \vec{y})$ is a scalar, and we do not know how to take the dot product of a scalar and a vector $\vec{z}$. Because of such difficulties, *you should be very accurate in your notation of the products.* Do not use $\cdot$ for scalar multiplication. Do not use $\times$ (we will give it a different meaning later). Just use a space for regular multiplication of scalars, and reserve $\cdot$ for a true dot product, of vectors.

- *geometric*: It turns out that the geometric interpretation of the dot product is an interesting combination of angles and lengths. Let the length (norm) of the vector $\vec{x}$ be $\|\vec{x}\|$, similar for $\vec{y}$, and let the angle from $\vec{x}$ to $\vec{y}$ be ϕ. Then it can be shown that: [1]

$$\vec{x} \cdot \vec{y} = \|\vec{x}\|\, \|\vec{y}\|\, \cos(\phi). \tag{11}$$

  So two things that we normally think of as different come together here: distance and angle.

Let us play some more with the interpretations. If we take the dot product of a vector with itself, we get computationally:

$$\vec{x} \cdot \vec{x} \mapsto x_1^2 + x_2^2 + x_3^2 \tag{12}$$

and from (11), since the angle between $\vec{x}$ and itself is zero:

$$\vec{x} \cdot \vec{x} = \|\vec{x}\|^2. \tag{13}$$

[1] But we will not show this here, we just give the result. We will encounter it later in Bretscher. You may attempt a proof now...

Combined, this gives us a formula to compute the norm of a vector $\vec{x}$:

$$\|\vec{x}\| = \sqrt{\vec{x}\cdot\vec{x}} = \sqrt{x_1^2 + x_2^2 + x_3^2}. \tag{14}$$

This is of course Pythagoras' theorem, applied to the coordinates of $\vec{x}$, now in 3-D. Since $\vec{x}\cdot\vec{x}$ can be computed in n-D, the symbolic formula is valid in n-D, and the computational formula can be extended simply to that case.

Another special case of (11) is when $\vec{x}$ and $\vec{y}$ are perpendicular. Then $\phi = \pm\pi/2$ (in high school, you called this ± 90 degrees but from now on you should work in radians), so $\cos(\phi) = 0$. Conversely, if the dot product of two non-null vectors is zero, they must be perpendicular. So we get:

$$\vec{x} \text{ perpendicular to } \vec{y} \;\Leftrightarrow\; \vec{x}\cdot\vec{y} = 0.$$

This is rather unlike what we know of multiplication of numbers: if the product of numbers satisfies $x\,y = 0$, then $x = 0$ or $y = 0$ (or both). Vectors can have a zero dot product without being zero.

We can also use (11) to make a formula for the angle ϕ between two vectors:

$$\cos(\phi) = \frac{\vec{x}\cdot\vec{y}}{\|\vec{x}\|\,\|\vec{y}\|} = \frac{x_1y_1 + x_2y_2 + x_3y_3}{\sqrt{x_1^2 + x_2^2 + x_3^2}\,\sqrt{y_1^2 + y_2^2 + y_3^2}}. \tag{15}$$

This can also be extended to n-dimensional space.

A useful special case is if $\vec{y}$ is a *unit vector* $\vec{e}$, which is a vector with unit norm $\|\vec{e}\| = 1$. Then we get:

$$\vec{x}\cdot\vec{e} = \|\vec{x}\|\,\cos(\phi). \tag{16}$$

This is precisely the length of the perpendicular projection of $\vec{x}$ on $\vec{e}$, see Figure 2. Therefore, if we multiple the unit vector $\vec{e}$ by this length, we get the component $(\vec{x}\cdot\vec{e})\,\vec{e}$ of $\vec{x}$ along the $\vec{e}$ direction. If you write all this out in coordinates by the computational method (even in 2-D), you will realize the advantage of the symbolic notation, which moreover holds for n dimensional space as well.

5 Applications of the dot product

5.1 Proof of Pythagoras

You can now prove Pythagoras' theorem for vectors:

$$\text{if } \vec{x} \text{ is perpendicular to } \vec{y} \quad\Leftrightarrow\quad \|\vec{x}+\vec{y}\|^2 = \|\vec{x}\|^2 + \|\vec{y}\|^2. \tag{17}$$

Exercise:

1. Prove this at the symbolic level, so encode both sides in terms of things at that level (hint: use the dot product). Avoid using the computational approach, do not spell everything out in coordinates!

2. Draw a picture (the geometric approach). It is not quite the Pythagoras triangle, for vectors should start at the origin!

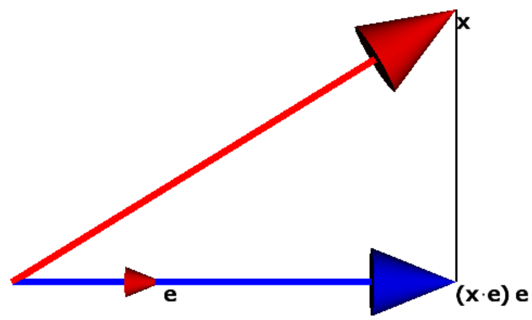
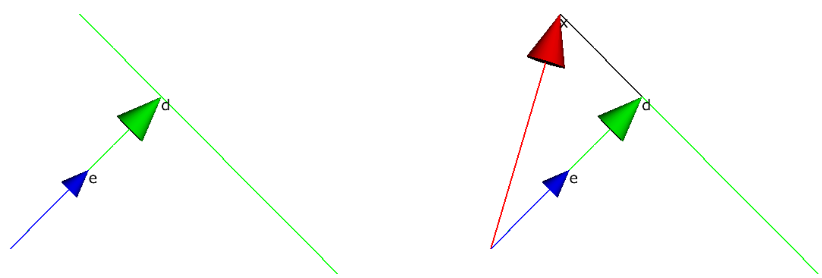


Figure 2: Projection of $\vec{x}$ onto a unit vector $\vec{e}$ using the inner product, resulting in $(\vec{x} \cdot \vec{e})\,\vec{e}$.



(a) (b)

Figure 3: Characterizing a line in 2D by a support vector $\vec{d}$ or a normal vector $\vec{e}$ and a distance δ such that $\vec{d} = \delta\,\vec{e}$.

5.2 A line in the plane

A line in the plane can be characterized using vectors.

- *geometric*: Look at Figure 3a. We have an origin O, and a green line which we want to characterize relative to this origin. A good idea is to use the vector $\vec{d}$, from the origin perpendicular to the line. All points of the line are obtained by "going to the end of $\vec{d}$ and looking left and right".

- *symbolic*: What does a vector $\vec{x}$ need to satisfy to point exactly to a point on the line? To encode that with previous constructions, let us make a unit vector $\vec{e}$ in the direction of $\vec{d}$, so that $\vec{d} = \delta\,\vec{e}$. Then δ is the distance of the line to the origin, in the $\vec{e}$-direction.

  Now we use (16): for any $\vec{x}$ on the line, the projection of $\vec{x}$ onto $\vec{e}$ should be have the length δ. You can see this clearly by comparing Figure 2 and

Figure 3(b).

When we express that geometric insight symbolically, we simple get:

$$\vec{x} \cdot \vec{e} = \delta \tag{18}$$

as the characterization of the line. Now we can get back to the original vector $\vec{d}$ by multiplying both sides by δ. The left hand side then gives $\vec{x} \cdot \vec{d}$ (why?), the right hand side δ^2, which is the same as $\vec{d} \cdot \vec{d}$ (why?).

So $\vec{x}$ points to a point on the line characterized by $\vec{d}$ if and only if:

$$\vec{x} \cdot \vec{d} = \vec{d} \cdot \vec{d}. \tag{19}$$

Actually, (18) is a bit more general, for it still works when $\delta = 0$, whereas (19) then has problems. How would you characterize the problem geometrically (take a line through the origin; what property of the line can (19) not encode?).

In either (18) or (19), the vectors $\vec{e}$ or $\vec{d}$ are called a *normal vector* for the line. This is jargon, 'normal' here means 'perpendicular'. Since $\vec{e}$ is a unit vector, it is called a *unit normal (vector)* for the line, and since $\vec{d}$ 'carries' the line it is sometimes called the *support vector* of the line.

- *computational*: Now let us look at his computationally, taking (18) in 2-D. Then the unit vector $\vec{e}$ can be specified relative to some fixed coordinate basis $\{\vec{e}_1, \vec{e}_2\}$. Let us have it make an angle ϕ with $\vec{e}_1$, then its components are:

$$\vec{e} \mapsto \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix} \tag{20}$$

Verify this, using your high school knowledge of a right-angled triangle with angle ϕ! Then (18) becomes, computationally:

$$x_1 \cos(\phi) + x_2 \sin(\phi) = \delta. \tag{21}$$

You may not recognize that yet. But in high school, you called x_1 (which is the component of $\vec{x}$ in the $\vec{e}_1$-direction) the x-coordinate of the point $\vec{x}$ points to, and x_2 the y-coordinate. So we get:

$$x \cos(\phi) + y \sin(\phi) = \delta. \tag{22}$$

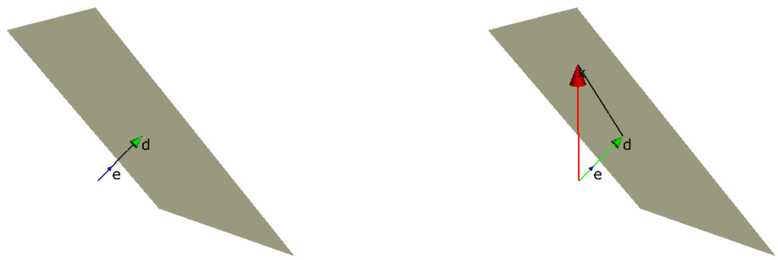
Also in high school, you were used to having things of the form $y = f(x)$, since you were dealing with functions. We are dealing with geometry, but let us write our equation in the same form:

$$y = -\frac{\cos(\phi)}{\sin(\phi)}\, x + \frac{\delta}{\sin(\phi)}, \tag{23}$$

though we can only do this when $\sin\phi \neq 0$. This is clearly something of the form

$$y = a\, x + b, \tag{24}$$

if we make the correct identifications by defining a and b in terms of ϕ and δ. So indeed, the original (18) defines a line in its more classical form.



(a) (b)

Figure 4: The representation of a plane in 3-D.

Yet (24) has problems. It is not defined when $\sin\phi = 0$, i.e. when the normal vector is in the y-direction. The line is then vertical, and a would have to be infinite, or we would have to switch over to a line of the form $x = c$. For hand computations that is OK, but in a computer we would prefer to have a representation that always works. So (18) is much better, for it has no problems with lines in any direction or location. Verify that for the example of a vertical line!

Exercises

3. Give the vector equation (18) of the line $y = x + 1$.
4. Give the vector equation (18) of the line $y = -x - 1$, but using $\vec{e} = (1/\sqrt{2}, 1/\sqrt{2})$.
5. Draw the line with $\vec{e} = (1/\sqrt{2}, 1/\sqrt{2})$ and $\delta = 1$.
6. Draw the line with $\vec{d} = (1, -1)$.
7. What is the distance to the origin (δ) of the line $y = x + 1$?
8. What is the distance to the origin (δ) of the line $y = a\,x + b$?
9. What is the distance to the origin (δ) of the line $x_1 - 2x_2 = 1$?
10. What is the distance to the origin (δ) of the line $a_1x_1 + a_2x_2 = c$?

5.3 A plane in space

We can use this also to represent planes in space.

- *symbolic*: In the symbolic equations we got, (18) and (19), there is nothing stating that we are working with vectors in the plane. So we can apply these equations in 3-D space as well.

- *geometric*: They then are still the equations for the perpendicular projection of $\vec{x}$ along $\vec{e}$ to have the length δ. That suggests the picture of Figure 4(b). It therefore seems to characterize a plane in space, with support vector $\vec{d}$ as in Figure 4(a).

- *computational*: It is more difficult to define directions and angles in space, and we will not do that here. But it is simple to verify that the symbolic equation resulting from the dot product has the form:

$$a_1x_1 + a_2x_2 + a_3x_3 = c. \tag{25}$$

  This represents a plane with a normal vector

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \tag{26}$$

  at a distance

$$\delta = \frac{c}{\sqrt{a_1^2 + a_2^2 + a_3^2}}. \tag{27}$$

Of course, it does not stop with 3-D. The equation (18) defines a *hyperplane* an n-dimensional space, i.e. a flat element of dimensionality $(n-1)$. In 2-D, the hyperplane is 1-D, and we are used to calling it a line; in 3-D, the hyperplane is 2-D and we call that a plane.

Exercises:

11. Derive the formula (27) for the distance of a plane with equation (25) to the origin.

12. Give the equation of a plane with normal vector $\begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$ at distance -1 from the origin.

13. Where does the plane with equation $2x_1 + 3x_2 - 5x_3 = 6$ cut the $\vec{e}_1$-axis?

14. The plane with equation $2x_1 + 3x_2 - 5x_3 = 6$ cuts the plane with equation $x_3 = 0$ in a line. What is the equation of that line?