Vectors

Mauro Mongiardo[1]

[1] Department of Engineering, University of Perugia, Perugia, Italy.

Table of contents

Vectors

Vectors

A linear space is one upon which addition and scalar multiplication are defined.

Although such a space is often called a "vector space", we will use the term "vector" for the geometric concept of a directed line segment.

A *vector* is a quantity having both direction and magnitude in space.

Examples of vector quantities are force, velocity, acceleration, etc.

We require linearity, so that for any vectors **a** and **b** we must be able to define their vector sum $\mathbf{a} + \mathbf{b}$.

We define the product of a scalar α and a vector **a** as $\alpha\mathbf{a}$. The multiplication of a vector times a scalar leaves the direction unchanged and the resulting vector is simply with a different magnitude.

Inner product

Inner product

In order to express algebraically the geometric idea of magnitude, it is convenient to define an Inner product.

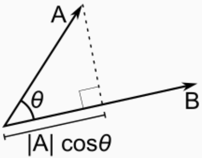

Figure 1: Inner product.

The *inner product* $\mathbf{a} \cdot \mathbf{b}$

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}| \cos \theta \tag{1}$$

is the scalar product of two vectors $\mathbf{a}$ and $\mathbf{b}$ and is a scalar with magnitude $|\mathbf{a}||\mathbf{b}| \cos \theta$, where $|\mathbf{a}|$ and $|\mathbf{b}|$ are the lengths of the vectors, and θ is the angle between them.

Cross product

The *cross product* $\mathbf{a} \times \mathbf{b}$ of two vectors is

$$\mathbf{a} \times \mathbf{b} = |\mathbf{a}||\mathbf{b}| \sin\theta \tag{2}$$

i.e. a vector of magnitude $|\mathbf{a}||\mathbf{b}| \sin\theta$ in the direction perpendicular to $\mathbf{a}$ and $\mathbf{b}$, such that $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{a} \times \mathbf{b}$ form a right-handed set.

However, the vector cross product exists only in our 3-dimensional world; in two dimensions there is simply no direction perpendicular to $\mathbf{a}$ and $\mathbf{b}$, and in four or more dimensions that direction is ambiguous.

A more general concept is needed, so that full information about relative directions can still be encoded in all dimensions. Before introducing the external product we would like to summarize the vector identities.

Vector identities

Vector identities

A summary of the vector identities is given in the following

$$a = \sqrt{\mathbf{a} \cdot \mathbf{a}} \tag{3}$$

$$\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a} \tag{4}$$

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a} \tag{5}$$

$$\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a} \tag{6}$$

$$(\mathbf{a} + \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot \mathbf{c} + \mathbf{b} \cdot \mathbf{c} \tag{7}$$

$$\mathbf{a} \cdot \mathbf{b} \times \mathbf{c} = \mathbf{b} \cdot \mathbf{c} \times \mathbf{a} = \mathbf{c} \cdot \mathbf{a} \times \mathbf{b} \tag{8}$$

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c} \tag{9}$$

Equation (5) is called the *commutative law*

while (7) is the *distributive law*.

It is seen from (6) that the cross product is *anticommutative*.

In (8) it is stated that it is possible to exchange the dot with the cross (naturally the cross product has to be performed first).

In the last equation (9) it is reported a well known identity (often memorized as bac minus cab).

It is noted that the above expressions are independent from the coordinate system. If one identity is proved in a coordinate system, it is valid in all the coordinate systems.

Computation of vector expression may often become quite tedious and it is of considerable advantage to be able to perform such computations, either symbolically and numerically, with a computer algebra system.

Vector identities with Computer Algebra Systems

Several Computer Algebra Systems (CAS) are presently available.

In the following we will make use of *wxMaxima* for three reasons:

- it is freely available;
- it runs on several operating systems,
- it allows to copy the result as a LaTeX expression.

The first example will refer to a segment of code for performing the dot and cross product. The file is:

`vectors_v02.wxm`

```
/* [wxMaxima batch file version 1] [ DO NOT EDIT BY HAND! ]*/
/* [ Created with wxMaxima version 11.08.0 ] */

/* [wxMaxima: input   start ] */
kill(all)$
load(vect);

print("Given the vectors")$
a:[a1, a2, a3];
b:[b1, b2, b3];
c:[c1, c2, c3];

print("dot product")$
dab : a . b;

print(" cross product")$
axb : a ~ b;
axb : express(axb);
bxa : express(b ~ a);
print("find  axb + bxa")$
diff : axb + bxa;

print("Perform triple scalar product")$
print("(A X B) . C")$
try1 : express(a ~ b) . c;
print("(C X A) . B")$
try2 : express(c ~ a) . b;
print("(B X C) . a")$
try3 : express(b ~ c) . a;
ratsimp(try1-try2);
ratsimp(try3-try2);

print("Perform triple vector product")$
print("first verify the rule A X ( B X C ) = B(A.C) -C(A.B)")$
print("A X ( B X C ) = ")$
r1 : express(a ~ express(b ~ c));
print(" B(A.C) -C(A.B) = ")$
r2 : b * (a . c) - c*(a . b);
ratsimp(r1-r2);

print("numerical example")$
A : [1,-1,2];
B : [0,1,1] ;
C : [-2,0,3] ;

print(" A X B")$
apb : express(A~B);
print(" (A X B) X C")$
apbpc : express(apb~ C);

print(" B X C")$
bpc : express(B~C);
print("A X (B X C)")$
apbpc2 : express(A ~ bpc);
print(" (A X B) X C  -  A X (B X C)")$
diff : (apbpc- apbpc2);

print("End")$
/* [wxMaxima: input   end   ] */

/* Maxima can't load/batch files which end with a comment! */
"Created with wxMaxima"$
```