Einstein Index Notation for Cartesian Vectors

Grant W. Mason

Department of Physics and Astronomy, Brigham Young University, Provo, Utah 84602

GETTING STARTED

This is a brief tutorial in a simple, yet very powerful and useful way of dealing with vectors and tensors. A vector is actually a kind of tensor, so this is a brief tensorial tutorial. The language we are introducing can be extended in several directions in straightforward ways. It was used by Albert Einstein as the original language of his General Theory of Relativity and a simple modification of it is also the basis for the most elegant formalism of the Special Theory of Relativity. The mathematical formalism can be used to give the general expressions for the divergence, curl and Laplacian in generalized coordinates.

Strictly speaking, a vector is a mathematical object that is useful for representing physical concepts like velocity, acceleration and electric fields. Mathematicians pride themselves on the fact that their mathematics does not necessarily have to be useful and in that sense, vectors do not necessarily have to represent anything useful in physics. But they do.

Let's start with the idea that a mathematical vector is an ordered set of three numbers, $\mathbf{a} = (a_1, a_2, a_3)$, that transform from one Cartesian coordinate system to another by a rule, $\mathbf{a}' = \Re\mathbf{a}$. The three numbers in the vector are what you probably know as the "components" of a vector. There are three vectors that are especially important, the *unit vectors*,

$$\begin{aligned}\hat{\mathbf{e}}_1 &= (1,0,0)\\ \hat{\mathbf{e}}_2 &= (0,1,0)\\ \hat{\mathbf{e}}_3 &= (0,0,1)\end{aligned}$$

These may otherwise be familiar to you as $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$. Using the unit vectors and a vector's components, we can express an arbitrary vector as,

$$\mathbf{a} = a_1\hat{\mathbf{e}}_1 + a_2\hat{\mathbf{e}}_2 + a_3\hat{\mathbf{e}}_3 = \sum a_j\hat{\mathbf{e}}_j.$$

If you think of the vector as being the ordered set of three numbers in each of two Cartesian coordinate systems, the second rotated relative to the first, then $\Re$ is a matrix that connects the two sets of components that represent the **same** vector, but in the two different coordinates systems,

$$\begin{pmatrix} a'_1 \\ a'_2 \\ a'_3 \end{pmatrix} = \begin{pmatrix} \Re_{11} & \Re_{12} & \Re_{13} \\ \Re_{21} & \Re_{22} & \Re_{23} \\ \Re_{31} & \Re_{32} & \Re_{33} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

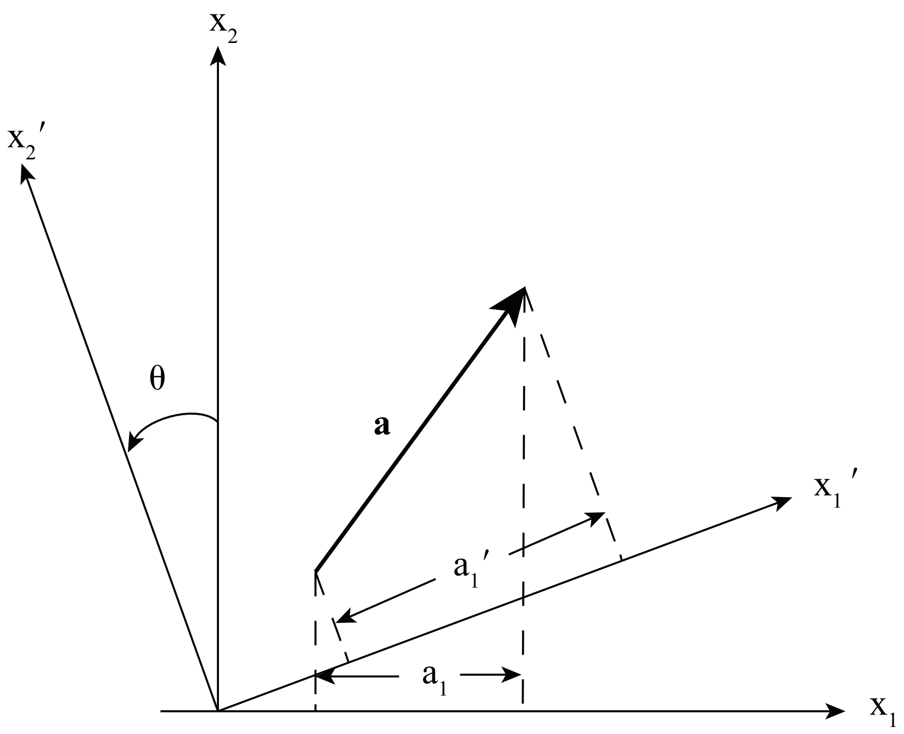

FIGURE 1. A vector **a** exists independent of a coordinate system. The vector has different components in two coordinate systems related to each other by a rotation θ.

A specific example is that of a primed coordinate system that is rotated relative to an unprimed system by a rotation around the "z" axis by an angle θ.:

$$\begin{pmatrix} a'_1 \\ a'_2 \\ a'_3 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

Writing this all out as matrices displays every detail explicitly, but it is certainly not as efficient and economical as writing it as $\mathbf{a}' = \Re\mathbf{a}$. There is yet a third way of writing it. We could write statements for each row of the elements of the column vector on the left,

$$a'_1 = \sum_j \Re_{1j} a_j$$

$$a'_2 = \sum_j \Re_{2j} a_j$$

$$a'_3 = \sum_j \Re_{3j} a_j$$

This latter form is somewhere in the middle in efficiency and economy, but can be improved significantly. Observe that there are two kinds of indices attached to the elements in the arrays: *connecting* and *dummy*. The connecting indices in each term are the ones that must be the same on both sides of the equal sign, i.e., they “connect” across the equal sign. In the first equation it is the “1”, in the second it is the “2” and in the third it is the “3”. The dummy index is the “j” in each of the three equations because there is a sum on that index and it doesn’t actually survive. Indeed, it doesn’t actually have to be a “j” at all. You could just as well write,

$$a'_1 = \sum_k \Re_{1k} a_k$$

or

$$a'_1 = \sum_{\heartsuit} \Re_{1\heartsuit} a_{\heartsuit}.$$

A dummy index just tells you that there is a sum with three terms and what numbers actually occur in each term. In each individual term, each of the two dummy indices in that term must match. A $\heartsuit$ serves that purpose perfectly well as could a $\diamondsuit$. Since the presence of the dummy index tells us that there is a sum, we could also dispense with writing the $\sum$ symbol. It is actually superfluous. We could then let the presence of a repeated dummy index tell us that there is a sum and write the three expressions for a'_1, a'_2, a'_3 more compactly as

$$a'_i = \Re_{i\heartsuit} a_{\heartsuit}$$

or

$$a'_i = \Re_{i\diamondsuit} a_{\diamondsuit}.$$

Since drawing hearts and diamonds is awkward and limited in variety, one can also use letters like “j” and “k” and “m” in place of the $\heartsuit$ and $\diamondsuit$ for the dummy indices:

$$a'_i = \Re_{ij} a_j.$$

The connecting index can be assigned values of 1,2, or 3 to generate the three components (a'_1, a'_2, a'_3), but until you actually need to, the one expression stands equally well for all three. The notation is efficient and economical, but it turns out to have surprising power. Einstein is said to have thought that it was one of his most valuable contributions (but that was probably in the context of the economy in which it allowed General Relativity to be expressed.)

To see how it helps us with vector expressions we have to introduce two new objects: the *Kronecker delta* and the *Levi-Civita tensor*. But don’t let the word “tensor” scare you. For our purposes, these are just simply arrays of numbers.

The symbol δ_{ij} is called the Kronecker delta. It stands for the elements of a 3x3 matrix whose elements are 1 if $i = j$ and 0 if $i \neq j$. You will probably recognize this as the 3x3 identity matrix, I.

The symbol ε_{ijk} is called the Levi-Civita tensor. It is a 3x3x3 matrix whose elements are +1 if (ijk) stand for an even permutation of (123), -1 if (ijk) stand for an odd permutation of (123), and 0 if any two indices are the same. The (123) are labels for the axes of a right-handed, Cartesian coordinate system.

To understand what an even permutation of 123 is, imagine that you have three beads on a string bracelet and that they are ordered 123. An even permutation of the beads is any ordering you can arrange by sliding the beads along and around the bracelet. The ordering 231 is such an even permutation and so is 312. An odd permutation is an ordering of the beads that would require that you cut the bracelet to put the beads into the new order. The ordering 213 is an odd permutation of 123 as is 132.

THE GRADIENT OPERATOR IS A VECTOR

Our notation becomes more powerful when we introduce the gradient operator, $\nabla_i \equiv \partial/\partial x_i$. You may have to shift mental gears a little to recognize that

$$(\nabla_1, \nabla_2, \nabla_3) \equiv (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3})$$

are components of a vector! However, ∇_i is more than a vector. It is a vector operator. It operates on something that is assumed to be to its right. Thus you don't have complete freedom to move it around relative to its neighbors in a term of an expression. Unless you delimit its action with parentheses, it is assumed to operate on everything to its right and unless that something is a constant, you cannot arbitrarily move it past ∇_i from right-to-left or left-to-right. Numbers, like a_i and b_i don't have that limitation: $a_i b_j \equiv b_j a_i$.

Anyway, here is how ∇_i becomes a vector. Go back to the original idea of a mathematical vector being an ordered set of three numbers that transform in a certain way:

$$a'_i = \Re_{ij} a_j.$$

For simplicity, think of a specific example. Think of a vector in two dimensions where,

$$\begin{pmatrix} a'_1 \\ a'_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

The matrix $\Re$ for the transformations that define a vector are orthogonal transformations that have the property that their inverses are their own transposes. Thus, if

$$\mathbf{a}' = \Re \mathbf{a}$$

then

$$\Re^T \mathbf{a}' = \Re^T \Re \mathbf{a} = \Re^{-1} \Re \mathbf{a} = \mathbf{a},$$

i.e.,

$$\mathbf{a} = \Re^T \mathbf{a}'$$

or, for a vector **r**,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}.$$

Now think of the chain rule for taking derivatives of a function, f,

$$\frac{\partial f}{\partial x_1'} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial x_1'} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial x_1'}$$

$$\frac{\partial f}{\partial x_2'} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial x_2'} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial x_2'}$$

We can use the matrix expression above for $\mathbf{a} = \Re^T \mathbf{a}'$ to get things like $\partial x_1/\partial x_1'$, etc. We can write,

$$\begin{pmatrix} \frac{\partial}{\partial x_1'} \\ \frac{\partial}{\partial x_2'} \end{pmatrix} f = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{pmatrix} f.$$

If you can bring yourself to “cancel” the f on both sides (the thing operated on), you will see that,

$$\begin{pmatrix} \frac{\partial}{\partial x_1'} \\ \frac{\partial}{\partial x_1'} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{pmatrix},$$

which is just the transformation property that defines a vector, $\nabla_i' = \Re_{ij}\nabla_j$. But, it is a vector *operator* since the function f is assumed sitting to the right of both sides.

SOME FAMILIAR FRIENDS

Here are some familiar things that we can write economically with our Einstein notation. Remember, unless otherwise indicated, it is assumed that there is a summation on repeated (dummy) indices.

$\mathbf{a} = a_\heartsuit \hat{\mathbf{e}}_\heartsuit$		(vector)
$\mathbf{a}\cdot\mathbf{b} = a_\heartsuit b_\heartsuit$		(dot product)
$\mathbf{c} = \mathbf{a}\times\mathbf{b}$	; $c_i = \varepsilon_{i\heartsuit\diamondsuit} a_\heartsuit b_\diamondsuit$ “i” is the connecting index)	(cross product)
$\det(A) = A_{1\heartsuit} A_{2\diamondsuit} A_{3\triangle} \varepsilon_{\heartsuit\diamondsuit\triangle}$		
$\det(A) = A_{\heartsuit 1} A_{\diamondsuit 2} A_{\triangle 3} \varepsilon_{\heartsuit\diamondsuit\triangle}$		(determinant of A)
$\nabla\cdot\mathbf{a} = \nabla_\heartsuit a_\heartsuit$		(divergence)
$\mathbf{c} = \nabla\times\mathbf{a}$	; $c_i = \varepsilon_{i\heartsuit\diamondsuit} \nabla_\heartsuit a_\diamondsuit$ “i” is the connecting index)	(curl)
$\nabla\psi = \hat{\mathbf{e}}_\heartsuit \nabla_\heartsuit \psi$		(gradient)
$\nabla^2\psi = \nabla_\heartsuit \nabla_\heartsuit \psi$		(Laplacian)

In these expressions the Levi-Civita tensor automatically keeps track of those confusing minus signs that permeate cross products and determinants. Since the expression for the determinant has three different sums over dummy indices, there are $3 \times 3 \times 3 = 27$ terms in the expression. In each term there is an ε with three indices to tell us whether that term has a positive or negative sign.

THE RULES

So, the rules are:

1. Repeated indices mean "sum". The repeated index is a dummy and may have *any* symbol as an index that is otherwise unused in a particular term. You could even use a $\heartsuit$. And, you can change a dummy index in a given term of an expression at will if you would like it to be something else. $a_\heartsuit b_\heartsuit \equiv a_\diamondsuit b_\diamondsuit \equiv a_j b_j$.

2. No more than two identical dummy indices are allowed in any term. $a_i b_i$ is OK but $a_i b_i c_i$ is not OK unless specifically noted and given meaning.

3. The operator $\nabla_i \equiv \partial/\partial x_i$ operates on everything to its right. It is a *vector* operator.

Example: $\nabla_i a_j b_k = a_j \nabla_i b_k + b_k \nabla_i a_j$

4. The operator ∇_i operating on a Cartesian coordinate, x_j yields 1 if $i = j$, but yields 0 if $i \neq j$.

$$\begin{aligned} &\nabla_i x_j = \delta_{ij}, \text{ where} \\ &\delta_{ij} = 0 \quad \text{if i} \neq \text{j} \\ &\quad\; = 1 \quad \text{if i=j} \qquad\qquad \text{(Kronecker delta)} \end{aligned}$$

5. Products of Kronecker deltas may be reduced if an index is repeated (summation).

$$\begin{aligned} &\delta_{i\heartsuit}\delta_{k\heartsuit} = \delta_{ik} \\ &\textit{but} \quad \delta_{\heartsuit\diamondsuit}\delta_{\heartsuit\diamondsuit} = \delta_{\diamondsuit\diamondsuit} = 3 \qquad \text{(trace of } \delta_{ij}) \end{aligned}$$

6. Relationships among elements of the Levi-Civita tensor.

$$\begin{aligned} &\varepsilon_{ijk} = -\varepsilon_{jik} = -\varepsilon_{ikj} = -\varepsilon_{kji} \qquad \text{(odd permutations)} \\ &\varepsilon_{ijk} = \varepsilon_{kij} = \varepsilon_{jki} \qquad \text{(even permutations)} \end{aligned}$$

7. Products of Levi-Civita tensors may be reduced if an index is repeated (summation) with the following pattern:

$$\varepsilon_{\heartsuit jk}\varepsilon_{\heartsuit lm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl} \qquad \text{("j","k","l","m" are connecting indices.)}$$

EXAMPLES

Example:

$$\begin{aligned}(\mathbf{a}\times\nabla)\cdot\mathbf{r} &= \varepsilon_{\heartsuit\diamondsuit\triangle}a_{\diamondsuit}\nabla_{\triangle}x_{\heartsuit}\\ &= \varepsilon_{\heartsuit\diamondsuit\triangle}a_{\diamondsuit}\delta_{\triangle\heartsuit}\\ &= \varepsilon_{\heartsuit\diamondsuit\heartsuit}a_{\diamondsuit}\\ &= 0\end{aligned}$$

We could give different names to the dummy indices and write the same thing,

$$\begin{aligned}(\mathbf{a}\times\nabla)\cdot\mathbf{r} &= \varepsilon_{ijk}a_j\nabla_k x_i\\ &= \varepsilon_{ijk}a_j\delta_{ki}\\ &= \varepsilon_{iji}a_j\\ &= 0\end{aligned}$$

The dummy index i is the dummy index for the sum of the dot product and $\varepsilon_{iji}=0$ because the elements of the Levi-Civita tensor are 0 if two indices are the same.

Here is another example that illustrates a trick of the trade. You probably know that the divergence of the curl vanishes identically. Here's how you might show that. The divergence of a curl has no components, so there is no connecting index. All indices are dummies,

$$\nabla\cdot(\nabla\times\mathbf{a})=\nabla_i\varepsilon_{ijk}\nabla_j a_k.$$

Here's the trick. Rewrite the right-hand side,

$$\frac{1}{2}((\nabla_i\varepsilon_{ijk}\nabla_j a_k)+(\nabla_i\varepsilon_{ijk}\nabla_j a_k)).$$

Concentrate on the second term by renaming the dummy indices. Then, i becomes j and j becomes i, i.e.,

$$\nabla_i\varepsilon_{ijk}\nabla_j a_k\equiv\nabla_j\varepsilon_{jik}\nabla_i a_k$$

Remember, those are dummy indices. They can have any name you want to give them including $\heartsuit$ and $\diamondsuit$. Use the property of the odd permutation of an element of the Levi-Civita tensor, $\varepsilon_{jik}=-\varepsilon_{ijk}$. Then the second term of the right-hand-side is $-\nabla_j\varepsilon_{ijk}\nabla_i a_k$. Finally, the order of differentiation commutes, $\nabla_j\nabla_i\equiv\nabla_i\nabla_j$ since the ε_{ijk} are constants and can be temporarily moved out of the way. So, when the smoke clears,

$$\nabla\cdot(\nabla\times\mathbf{a})=\frac{1}{2}((\nabla_i\varepsilon_{ijk}\nabla_j a_k)-(\nabla_i\varepsilon_{ijk}\nabla_j a_k))\equiv 0! \quad (QED)$$

It may seem a little like magic, but what you were actually doing was reordering the pieces in that second term until they were piece-for-piece identical to the pieces in the first term, but with an opposite sign so that you could see explicitly that they add to zero. You could have also written out the whole nine terms on a big sheet of paper and hunted down the cancelling pairs, but that would not have been elegant or fun!

Ever tried to show (or remember while you were castaway on a desert island) that $\mathbf{A}\times(\mathbf{B}\times\mathbf{C}) = \mathbf{B}(\mathbf{A}\cdot\mathbf{C}) - \mathbf{C}(\mathbf{A}\cdot\mathbf{B})$? This expression has three components, so we will need a connecting index. Let’s choose “i”.

$$[\mathbf{A}\times(\mathbf{B}\times\mathbf{C})]_i = (\varepsilon_{ijk}a_j)(\varepsilon_{klm}b_l c_m) = \varepsilon_{ijk}\varepsilon_{klm}a_j b_l c_m.$$

Use the property that, for even permulations, $\varepsilon_{ijk} = \varepsilon_{kij}$ to put the product of the two Levi-Civita tensors into the pattern (dummy index in first slot for both) to allow reducing to Kronecker deltas.

$$[\mathbf{A}\times(\mathbf{B}\times\mathbf{C})]_i = (\delta_{il}\delta_{jm} - \delta_{im}\delta_{jl})a_j b_l c_m = \delta_{il}\delta_{jm}a_j b_l c_m - \delta_{im}\delta_{jl}a_j b_l c_m.$$

Clear the Kronecker deltas with patterns like $\delta_{il}b_l = b_i$ to remove *dummy* indices. Be sure that you keep your connecting index in each term!

$$[\mathbf{A}\times(\mathbf{B}\times\mathbf{C})]_i = b_i(a_m c_m) - c_i(a_l b_l).$$

You should be able to recognize that this is an expression for the “i”th component of

$$\mathbf{A}\times(\mathbf{B}\times\mathbf{C}) = \mathbf{B}(\mathbf{A}\cdot\mathbf{C}) - \mathbf{C}(\mathbf{A}\cdot\mathbf{B}). \quad (QED)$$

With a little practice, you can learn to manipulate vector identities in a very efficient and economical way.