Tensor Notation (Basics)

Introduction

Tensor (or index, or indicial, or Einstein) notation has been introduced in the previous pages during the discussions of vectors and matrices. This page reviews the fundamentals introduced on those pages, while the next page goes into more depth on the usefulness and power of tensor notation.

This page repeats the tensor notation segments of earlier pages nearly verbatim. If you have already read them, then there is nothing new here. You can continue to the next page, which addresses more advanced tensor notation topics.

Summation Convention

Tensor notation introduces one simple operational rule. It is to automatically sum any index appearing twice from 1 to 3. As such, $a_i b_j$ is simply the product of two vector components, the i^{th} component of the **a** vector with the j^{th} component of the **b** vector. However, $a_i b_i$ is a completely different animal because the subscript i appears twice in the term. Therefore, $a_i b_i$ automatically expands to, and is shorthand for...

$$a_i b_i \equiv a_1 b_1 + a_2 b_2 + a_3 b_3$$

which is just the dot product of vectors **a** and **b**. Note that any letter could be used as the index (as i was in this case), in order to invoke the automatic summation, however, it is critical that the same letter be used in both subscripts in order to do so.

Kronecker Delta

Tensor notation introduces two new symbols into the mix, the Kronecker Delta, δ_{ij}, and the alternating or permutation tensor, ϵ_{ijk}. The Kronecker Delta, δ_{ij}, serves as the identity matrix, **I**, because it equals 1 when $i = j$ and 0 otherwise. So in a matrix indexed by i and j, the diagonal components equal 1 and the off-diagonal components equal 0.

Is It a Matrix or Not?

A note from the purests... The identity matrix is a matrix, but the Kronecker delta technically is not. δ_{ij} is a single scalar value that is either 1 or 0 depending on the values of i and j. Or think of it as an individual component of the identity

matrix, $\mathbf{I}$. This is also why tensor notation is not in bold, because it always refers to individual components of tensors, but never to a vector, matrix, or tensor as a whole.

Follow this link for an entertaining discussion between someone who gets it, and someone else who doesn't.

Kronecker Delta verses Dirac Delta

Don't confuse the Kronecker delta, δ_{ij}, with the Dirac delta, $\delta_{(t)}$. The Dirac delta is something totally different. It is often used in signal processing and equals 0 for all t except $t = 0$. At $t = 0$, it approaches ∞ such that

$$\int f(t)\delta_{(t)}dt = f(0)$$

Alternating Tensor

The alternating tensor, ϵ_{ijk}, is used in cross products as follows.

$$c_i = \epsilon_{ijk}a_j b_k \qquad \text{corresponds to} \qquad \mathbf{c} = \mathbf{a} \times \mathbf{b}$$

where $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$, while $\epsilon_{321} = \epsilon_{213} = \epsilon_{132} = -1$, and all other combinations equal zero. Summation of the j and k indices from 1 to 3 is implied because they are repeated as subscripts. In other words, it is shorthand for

$$\begin{aligned} c_i = \epsilon_{ijk}a_j b_k = {} & \epsilon_{i11}a_1 b_1 + \epsilon_{i12}a_1 b_2 + \epsilon_{i13}a_1 b_3 + \\ & \epsilon_{i21}a_2 b_1 + \epsilon_{i22}a_2 b_2 + \epsilon_{i23}a_2 b_3 + \\ & \epsilon_{i31}a_3 b_1 + \epsilon_{i32}a_3 b_2 + \epsilon_{i33}a_3 b_3 \end{aligned}$$

The equation is still general until a particular component is chosen for i to be evaluated.

Cross Products Using Tensor Notation

Set $i = 3$ to obtain the z$^{\text{th}}$ component of a cross product.

$$\begin{aligned} c_3 = \epsilon_{3jk}a_j b_k = {} & \epsilon_{311}a_1 b_1 + \epsilon_{312}a_1 b_2 + \epsilon_{313}a_1 b_3 + \\ & \epsilon_{321}a_2 b_1 + \epsilon_{322}a_2 b_2 + \epsilon_{323}a_2 b_3 + \\ & \epsilon_{331}a_3 b_1 + \epsilon_{332}a_3 b_2 + \epsilon_{333}a_3 b_3 \end{aligned}$$

All subscripts are now specified, and this permits evaluation of all alternating tensors. All of them will equal zero except two. This leaves

$$c_3 = \epsilon_{3jk} a_j b_k = a_1 b_2 - a_2 b_1$$

which is consistent with the determinant result (as it had better be). Results for the xth and yth components are obtained by setting i equal to 1 and 2, respectively.

Vector and Tensor Addition

Vector and tensor addition is written in tensor notation simply as

$$c_i = a_i + b_i \qquad \text{and} \qquad c_{ij} = a_{ij} + b_{ij}$$

Vector Dot Products

A dot product of vectors **a** and **b** is written in tensor notation simply as $a_i b_i$. The summation from 1 to 3 is implied because the subscript (i in this case) appears twice (on a and b). In other words:

$$a_i b_i \equiv a_1 b_1 + a_2 b_2 + a_3 b_3$$

Note that any letter could be used as the index (as i was in this case), however, it is critical that the same letter be used on both subscripts in order to invoke the automatic summation.

Vector Cross Products

The alternating tensor, ϵ_{ijk}, is used in cross products as follows. (Yes, this does repeat the alternating tensor section above.)

$$c_i = \epsilon_{ijk} a_j b_k$$

where $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$, while $\epsilon_{321} = \epsilon_{213} = \epsilon_{132} = -1$, and all other combinations equal zero. Summation of the j and k indices from 1 to 3 is implied because they are repeated as subscripts in the above equation. In other words, it is shorthand for

$$\begin{aligned} c_i = \epsilon_{ijk} a_j b_k = {} & \epsilon_{i11} a_1 b_1 + \epsilon_{i12} a_1 b_2 + \epsilon_{i13} a_1 b_3 + \\ & \epsilon_{i21} a_2 b_1 + \epsilon_{i22} a_2 b_2 + \epsilon_{i23} a_2 b_3 + \\ & \epsilon_{i31} a_3 b_1 + \epsilon_{i32} a_3 b_2 + \epsilon_{i33} a_3 b_3 \end{aligned}$$

The equation is still general until a particular component is chosen for i to be evaluated.

Cross Products Using Tensor Notation

Set $i = 3$ to obtain the zth component of a cross product.

$$\begin{aligned} c_3 = \epsilon_{3jk} a_j b_k = {} & \epsilon_{311} a_1 b_1 + \epsilon_{312} a_1 b_2 + \epsilon_{313} a_1 b_3 + \\ & \epsilon_{321} a_2 b_1 + \epsilon_{322} a_2 b_2 + \epsilon_{323} a_2 b_3 + \\ & \epsilon_{331} a_3 b_1 + \epsilon_{332} a_3 b_2 + \epsilon_{333} a_3 b_3 \end{aligned}$$

All subscripts are now specified, and this permits evaluation of all alternating tensors. All of them will equal zero except two. This leaves

$$c_3 = \epsilon_{3jk} a_j b_k = a_1 b_2 - a_2 b_1$$

which is consistent with the determinant result (as it had better be). Results for the xth and yth components are obtained by setting i equal to 1 and 2, respectively.

The following example of area calculation of a triangle illustrates an important property of tensor notation, namely that the indices dictate the summation and order of multiplication, not the order in which the terms are written.

The area of a triangle bounded on two sides by vectors $\mathbf{a}$ and $\mathbf{b}$ is

$$Area = \frac{1}{2} |\mathbf{a} \times \mathbf{b}|$$

In tensor notation, this is written in two steps as

$$c_i = \epsilon_{ijk} a_j b_k \qquad \text{and} \qquad Area = \frac{1}{2} \sqrt{c_i c_i}$$

or in a single equation as

$$Area = \frac{1}{2} \sqrt{\epsilon_{ijk} a_j b_k \epsilon_{imn} a_m b_n}$$

Note that each index appears twice in the above equation because, by convention, it is not permitted to appear more than 2 times.

Order of Factors in Tensor Notation

Tensor notation allows for increased flexibility of the order in which factors are written than is permitted in vector notation. For example, $\mathbf{a} \times \mathbf{b}$ is not equal to $\mathbf{b} \times \mathbf{a}$, although they are closely related. In contrast $\epsilon_{ijk} a_j b_k$ equals $\epsilon_{ijk} b_k a_j$ equals $a_j b_k \epsilon_{ijk}$ because the order of operation is dictated by the indices rather than the order the factors are written in. So in the above discussion, $\epsilon_{ijk} a_j b_k \epsilon_{imn} a_m b_n$ could also be written as $\epsilon_{ijk} \epsilon_{imn} a_j b_k a_m b_n$. It is simply a matter of personal preference.

Vector Diadic Products

Tensor notation of a diadic product could not be simpler.

$$c_{ij} = a_i b_j$$

Diadic Product Example

If $\mathbf{a} = (3,7,2)$, and $\mathbf{b} = (1,2,3)$, then the diadic product of the two is

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} 3*1 & 3*2 & 3*3 \\ 7*1 & 7*2 & 7*3 \\ 2*1 & 2*2 & 2*3 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 6 & 9 \\ 7 & 14 & 21 \\ 2 & 4 & 6 \end{bmatrix}$$

Tensor notation dictates that the value of any c_{ij} component is simply $a_i b_j$. For example, selecting $i = 2$ and $j = 3$ gives

$$c_{23} = a_2 b_3 = 7*3 = 21$$

Matrix Transpose

The transpose of A_{ij} is A_{ji}.

Determinant

The calculation of a determinant can be written in tensor notation in a couple different ways

$$\det(\mathbf{A}) = \epsilon_{ijk} A_{i1} A_{j2} A_{k3} = \frac{1}{6} \epsilon_{ijk} \epsilon_{rst} A_{ir} A_{js} A_{kt}$$

Matrix Inverse

The inverse of A_{ij} is written as A_{ij}^{-1}. Note that this is probably not rigorously correct since, as discussed earlier, neither A_{ij} nor A_{ij}^{-1} are technically matrices themselves. They are only components of a matrix. Oh well...

The inverse can be calculated using

$$A_{ij}^{-1} = \frac{1}{2\det(\mathbf{A})}\epsilon_{jmn}\,\epsilon_{ipq}A_{mp}A_{nq}$$

Matrix Multiplication

The dot product of two matrices multiplies each row of the first by each column of the second. Products are often written with a dot in matrix notation as $\mathbf{A}\cdot\mathbf{B}$, but sometimes written without the dot as $\mathbf{AB}$. Multiplication rules are in fact best explained through tensor notation.

$$C_{ij} = A_{ik}B_{kj}$$

(Note that no dot is used in tensor notation.) The k in both factors automatically implies

$$C_{ij} = A_{i1}B_{1j} + A_{i2}B_{2j} + A_{i3}B_{3j}$$

which is the ith row of the first matrix multiplied by the jth column of the second matrix. If, for example, you want to compute C_{23}, then $i = 2$ and $j = 3$, and

$$C_{23} = A_{21}B_{13} + A_{22}B_{23} + A_{23}B_{33}$$

Matrix Multiplication Example

If $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 2 & 2 \\ 2 & 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$ then $A_{ik}B_{kj}$ implies

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 2 & 2 \\ 2 & 3 & 4 \end{bmatrix}\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 14 & 32 & 50 \\ 14 & 38 & 62 \\ 20 & 47 & 74 \end{bmatrix}$$

Tensor Notation and Computer Programming

Another advantage of tensor notation is that it spells for you how to write the computer code to do it. Note how the subscripts in the FORTRAN example below exactly match the tensor notation for $C_{ij} = A_{ik}B_{kj}$. This is true for all tensor notation operations, not just this matrix dot product.

```
subroutine aa_dot_bb(n,a,b,c)
dimension a(n,n), b(n,n), c(n,n)
do i = 1,n
   do j = 1,n
      c(i,j) = 0
      do k = 1,n
         c(i,j) = c(i,j) + a(i,k) * b(k,j)
      end do
   end do
end do
return
end
```

Double Dot Products

The double dot product of two matrices produces a scalar result. It is written in matrix notation as $\mathbf{A} : \mathbf{B}$. Once again, its calculation is best explained with tensor notation.

$$\mathbf{A} : \mathbf{B} = A_{ij} B_{ij}$$

Since the i and j subscripts appear in both factors, they are both summed to give

$$\begin{aligned} \mathbf{A} : \mathbf{B} = A_{ij} B_{ij} = \quad & A_{11} * B_{11} + A_{12} * B_{12} + A_{13} * B_{13} + \\ & A_{21} * B_{21} + A_{22} * B_{22} + A_{23} * B_{23} + \\ & A_{31} * B_{31} + A_{32} * B_{32} + A_{33} * B_{33} \end{aligned}$$

Double Dot Product Example

If $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 2 & 2 \\ 2 & 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$ then

$$\begin{aligned} A_{ij} B_{ij} &= 1*1 + 2*4 + 3*7 + \\ &\quad 4*2 + 2*5 + 2*8 + \\ &\quad 2*3 + 3*6 + 4*9 \\ &= 124 \end{aligned}$$

Differentiation With Respect To Time

Differentiation with respect to time can be written in several forms.

$$\text{velocity} = \frac{d\mathbf{x}}{dt} = \left(\frac{dx_1}{dt}, \frac{dx_2}{dt}, \frac{dx_3}{dt} \right) = \dot{\mathbf{x}} = \dot{x}_i = x_{i,t}$$

$$\text{acceleration} = \frac{d\mathbf{v}}{dt} = \left(\frac{dv_1}{dt}, \frac{dv_2}{dt}, \frac{dv_3}{dt}\right) = \dot{\mathbf{v}} = \dot{v}_i = v_{i,t}$$

$$\text{acceleration} = \frac{d^2\mathbf{x}}{dt^2} = \left(\frac{d^2x_1}{dt^2}, \frac{d^2x_2}{dt^2}, \frac{d^2x_3}{dt^2}\right) = \ddot{\mathbf{x}} = \ddot{x}_i = x_{i,tt}$$

One can use the derivative with respect to t, or the dot, which is probably the most popular, or the comma notation, which is a popular subset of tensor notation. Note that the notation $x_{i,tt}$ somewhat violates the tensor notation rule of double-indices automatically summing from 1 to 3. This is because time does not have 3 dimensions as space does, so it is understood that no summation is performed.

Differentiation With Respect To Spatial Coordinates

Differentiation of a scalar function $f(\mathbf{x})$ with respect to x_j is

$$\frac{\partial f}{\partial x_j} \quad \text{or} \quad f_{,j}$$

Differentiation of a vector, $\mathbf{v}$, is

$$\frac{\partial \mathbf{v}}{\partial x_j} \quad \text{or} \quad \left(\frac{\partial v_x}{\partial x_j}, \frac{\partial v_y}{\partial x_j}, \frac{\partial v_z}{\partial x_j}\right) \quad \text{or} \quad v_{i,j}$$

Differentiation of a tensor, $\boldsymbol{\sigma}$, is

$$\frac{\partial \boldsymbol{\sigma}}{\partial x_k} \quad \text{or} \quad \sigma_{ij,k}$$

As with vectors, every component of a tensor is differentiated.

Divergence

The divergence of a vector is a scalar result. It is written as $v_{i,i}$ and computed as

$$v_{i,i} = \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3}$$

$$= \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

As stated above, the divergence is written in tensor notation as $v_{i,i}$. It is very important that both subscripts are the same because this dictates that they are automatically summed from 1 to 3. They can in fact be any letter one desires, so long as they are both the same letter.

Divergence Example

If $\mathbf{v} = (3x^2 - 2y, z^2 + x, y^3 - z)$, then the dot product is

$$v_{i,i} = \frac{\partial}{\partial x}(3x^2 - 2y) + \frac{\partial}{\partial y}(z^2 + x) + \frac{\partial}{\partial z}(y^3 - z) = 6x - 1$$

Curl

The curl of a vector is written in tensor notation as $\epsilon_{ijk} v_{k,j}$. It is critical to recognize that the vector is written as $v_{k,j}$ here, not $v_{j,k}$. This is because the curl is $\nabla \times \mathbf{v}$, not $\mathbf{v} \times \nabla$.

As with cross products, the fact that j and k both occur twice in $\epsilon_{ijk} v_{k,j}$ dictates that both are automatically summed from 1 to 3. The term expands to

$$\begin{aligned} \epsilon_{ijk} v_{k,j} = {} & \epsilon_{i11} v_{1,1} + \epsilon_{i12} v_{2,1} + \epsilon_{i13} v_{3,1} + \\ & \epsilon_{i21} v_{1,2} + \epsilon_{i22} v_{2,2} + \epsilon_{i23} v_{3,2} + \\ & \epsilon_{i31} v_{1,3} + \epsilon_{i32} v_{2,3} + \epsilon_{i33} v_{3,3} \end{aligned}$$

Curls Using Tensor Notation

To obtain the yth component of a curl, set i equal to 2 in the above equation.

$$\begin{aligned} \epsilon_{2jk} v_{k,j} = {} & \epsilon_{211} v_{1,1} + \epsilon_{212} v_{2,1} + \epsilon_{213} v_{3,1} + \\ & \epsilon_{221} v_{1,2} + \epsilon_{222} v_{2,2} + \epsilon_{223} v_{3,2} + \\ & \epsilon_{231} v_{1,3} + \epsilon_{232} v_{2,3} + \epsilon_{233} v_{3,3} \end{aligned}$$

All subscripts are now specified, and this permits evaluation of all alternating tensors. All of them will equal zero except two, leaving

$$\epsilon_{2jk} v_{k,j} = v_{1,3} - v_{3,1} = \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}$$

which is again consistent with the determinant result (as it must be). Results for the xth and zth components are obtained by setting i equal to 1 and 3, respectively.

Laplacian

The Laplacian is the divergence of the gradient of a function. It often arises in 2nd order partial differential equations and is written in matrix notation as $\nabla^2 f(\mathbf{x})$ and in tensor notation as $f_{,ii}$. Its definition is

$$f_{,ii} \equiv \frac{\partial^2 f(\mathbf{x})}{\partial x^2} + \frac{\partial^2 f(\mathbf{x})}{\partial y^2} + \frac{\partial^2 f(\mathbf{x})}{\partial z^2}$$

Laplacian Example

Determine the Laplacian of $f(\mathbf{x}) = 2x^3 y - z\sin(y)$.

Start by calculating the gradient of $f(\mathbf{x})$.

$$f_{,i} = \left(6x^2 y, 2x^3 - z\cos(y), -\sin(y)\right)$$

And the divergence of the gradient (which is the Laplacian after all) is

$$f_{,ii} = \nabla \cdot \nabla f(\mathbf{x}) = 12xy + z\sin(y)$$

Derivatives of Products

The product rule applies to the derivatives of vector (and tensor) products just as it does for scalar products. Examples include the gradient of a dot product

$$(a_i b_i)_{,j} = a_{i,j} b_i + a_i b_{i,j}$$

the derivative of a cross product

$$(\epsilon_{ijk} a_j b_k)_{,m} = \epsilon_{ijk} a_{j,m} b_k + \epsilon_{ijk} a_j b_{k,m}$$

and the derivative of a diadic product.

$$(a_i b_j)_{,k} = a_{i,k} b_j + a_i b_{j,k}$$