**Tensors and stuff**

(note we’re using Einstein summation convention, implicitly; maybe see Appendix at bottom)

**4. Scalars, Vectors, and Tensors**

So now I’d like to discuss scalars, vectors, and tensors – namely what they are, and whether a set of quantities we’re considering qualifies as a scalar or the components of a vector or tensor. The definition of these quantites is tied up in how they transform when one changes coordinate systems. So suppose we have some coordinate system, ua and a metric gmn(ua). We may change coordinates to u´α and the metric will subsequently change form to g´μν(u´α) – and we know how these all relate. But fundamentally, we will have only changed our description of this space, and not the space itself.

Now say we have some entity which can be expressed as a function of the coordinates as: φ(xa). What is it in the new coordinate system, φ´(x´α)? That’s *not* automatically known. In other words, we cannot simply say that φ´(x´α) = φ(xa) [well, meaning they have the same values at a given point in space, regardless of how it is described. maybe better to write this as φ´(xa) = φ(xa)], which would make it a scalar if true. There must be some rule, beyond this tensor stuff, to tell us. Generally, in physics, one says that entities transform *covariantly*, meaning they are constructed the same *way*, from physical observables I suppose. This doesn’t necessarily mean that they have the same functional form, i.e., that if in a Cartesian system φC(x,y) = x2 + y2, then in a polar system φP(r,φ) = r2 + φ2. It just means that the functions are constructed from the same physical considerations. So for instance, consider D(xa,yb) which calculates the distance between any two points. In each coordinate system we’d express this as:



These two expressions are constructed the same way, from their respective metrics, but certainly don’t have the same functional form [but it just *so happens* that they do, if we’re considering just flat metrics in Special Relativity]. Another example would be a matter density function. We’d say that



Thing is, though, we wouldn’t be able to get *number of particles*, without using ρ(xa) in the first place. So it would kind of necessarily work out that ρ´(x´α) = ρ(xa) since dV´ = dV necessarily too [because ∫dV´ = ∫dV for any volume]. But this changes in special relativity for instance. In that case, supposing the primed reference frame to be moving at speed u w/r to the unprimed one, we’d have, based on physical considerations about length contraction, that ρ´(x´α) = γρ(xa), basically because dV´ = dV/γ [in SR, spatial volumes aren’t invariant, though space-time volumes are]. Continuing the SR frame of thought, we’d have the same principle in electrodynamics. Here we might be interested in a set of constructs, like an electric field, say.



The *construction* of fields is covariant. Namely, we use the same Maxwell’s equations to get the fields, using the observed densities and currents as input. But the functional *form* of the resultant construction is *not* covariant. It couldn’t be since for instance, in one frame charges could be stationary and B = 0, whereas in another frame charges would be moving and therefore B ≠ 0.

**4a. Scalars**

Consider a quantity q which is constructed from the coordinates ua according to some generic process to give us: q(ua). And likewise in coordinate system u´α it would be constructed by the same unspecified process to give us q´(u´α). The primes seem excessive (see red comment above in previous page), but it is usefull in the sense that it clearly separates the quantity from the coordinates. For instance, if we were to write q´(ua), this would still refer to q´. The fact that it is expressed in terms of the other system’s coordinates is not relevant per se´; we can express our quantity in terms of whatever system’s coordinates we choose. It won’t affect the value of the quantity at that physical point. Often we are interested in ascertaining whether such a quantity has geometrical significance that transcends the coordinate system it’s expressed in. Most often we want to know if the value of the quantity remains invariant under coordinate transformation, i.e., q´(u´α) = q(ua). If so, then the quantity is called a scalar. Note that by this we mean q constructed in one frame and evaluated at a given point P described as ua, has the same value as q constructed in another frame (and now denoted q´) and evaluated at the same given point P, which is described in this frame as u´α (though we can use whichever coordinates we want to evaluate the equality, namely we could check q´(ua) = q(ua), or q´(u´α) = q(u´α), etc.).



**Example**

Show that q = (ds)2 is a scalar. So we must form this expression in ua, and also in u´α and see if these are the same. So we have:



And in the other coordinate system it is:



and these are the same since the last expression



equals the first expression. So (dr)2 is a scalar – as we admitted before. And we expect that it is in fact, because (dr)2 is the distance between two points, and is independent of any coordinate system the points are expressed in.

**Example**

Show that the differential volume element is a scalar. So recall that a volume element is given by:



Volume elements are invariant under a coordinate transformation, as should be the case since the volume in any given space should be the same regardless of how you describe it. We can show this via:



We’ll note that we used the Jacobian which relates a product of differentials in one coordinate system, to a product of differentials in another:



So incidentally, if we have the measure in one coordinate system, we can express it an another via:



**Example**

Consider the coordinate transformation below. Is q(x,y) = x2 + y2 a scalar, presuming one calculates q by the same functional form?



Well let’s check. In the (u,v) coordinate system we’d have q(u,v) = u2 + v2. Let’s check that q(x,y) = q(u,v)…



So this entity is a scalar. And of course it is, because x2 + y2 is the absolute distance of the particle (squared) from the origin. And the coordinate transformation is just a rotation which won’t change how that distance is described.

**4b. Vectors**

Now consider a *set* of quantities va(ua), or va(ua) [ua, again, stands for set of unprimed coordinates, and index a on v is not tied to index on u per se´]. As with the previous case, we typically know a priori how this set of quantities transforms from one coordinate system to another, namely in coordinate system ua they would be constructed via some process, and in coordinate system u´α they would be constructed according to the same process (though not same functional form, mind you): v´α(u´α), v´α(u´α). Again we are interested in ascertaining whether such a set of quantities has geometrical significance that transcends the coordinate system it’s expressed in. Most often we want to know if the set of quantities form the components of a *vector*, **T**. So what we’d do is express the components in each coordinate system and see if they’re related as explicated next. So a vector is a quantity with magnitude and direction which remains invariant under coordinate transformations. Basically, a vector is a *scalar* combination of the three basis vectors.



where Ta are called the covariant components of the vector, and Ta are called the contravariant components of the vector. Note that under this definition, the individual basis vectors by themselves are *not* ‘vectors’ strictly speaking, as they *do* change under coordinate transformations. FWIW, we can get the components of the tensor via:



Now since **T** is by assumption a scalar, this means that **T**, expressed in two different coordinate systems, must be invariant. And so for instance,



And so we must have the following transformation law for vector components:



For example, what this means is that the prospective vector components T´α, constructed in coordinate system u´α and evaluated at point P is equal to XaαTa, where Ta are the prospective tensor components constructed in coordinate system ua and evaluated at the same point P. Of course the X terms are also evaluated at the same point P as well. Now expressing it in terms of its contravariant components in ua we have:





That these two components are related via simple change of base is apparent as well if we use the resolution of identity technique. Consider:



Parenthetically, observe that we can get the covariant vector components from the contravariant vector components and vice versa using the metric. For instance,



and,



and so we have that:



We can calculate magnitudes of vectors, with the metric. For instance,



and this may be equivalently written,



We’ll observe, later, that this is a scalar, and so invariant in all reference frames, as it should be.

**Example**

Show that the gradient, ∂φ/∂ua, where φ is a scalar, forms the covariant components of a vector. To do this we want to express how ∂φ/∂ua relates to itself expressed in another coordinate system, and see how they’re related.



So we see that the gradient forms the covariant components of a vector because these components follow the covariant transformation law. And this makes sense because the gradient vector is:



is the vector which points in the direction of the steepest descent of a function. And this direction would be the same regardless of the coordinate system it is expressed in.

**Example**

Show that dua form the contravariant components of a vector. To do this we express it in terms of itself in another coordinate system,



So it does indeed correspond to the contravariant components of a vector. And this makes sense as well since we can write,



So we see that dua form the contravariant components of the differential displacement between two points, which is certainly a coordinate system-invariant entity.

**Example**

Consider two coordinate systems, the usual Cartesian one ua = (x,y), and another u´α = (u,v) related to first via a rotation. Verify that (x y) forms the components of a contravariant vector under a covariant functional form transformation.



So what we do is form this entity in the Cartesian system and see how it relates to the same in uα system. So we write:



And it is the case that:



which is the way contravariant vector components transform. And this makes sense because the vector **r** is just:



which is invariant w/r to coordinate system.

**4c. Tensors**

Now consider a *set* of quantities Mab(ua) or Mab(ua) or Mab(ua), or Mab(ua) for instance. As with the previous case, we typically know a priori how this set of quantities transforms from one coordinate system to another, namely in coordinate system ua they would be constructed via some process, and in coordinate system uα they would be constructed according to the same process (not same functional form): M´αβ(u´α) or M´αβ(u´α) or M´αβ(u´α), or M´αβ(u´α). Just to be clear, other transformation laws are possible, but in physics, covariance, is the most common. And like before, we are interested in ascertaining whether such a set of quantities has geometrical significance that transcends the coordinate system it’s expressed in. Most often we want to know if the set of quantities form the components of a *tensor*, **T**. So what we’d do is express the components in each coordinate system and see if they’re related as explicated next. So a tensor is a quantity with magnitude and direction(s) which remains invariant under coordinate transformations. Basically, a tensor is a *scalar* combination of products of two or more basis vectors. I’ll just illustrate two:



where Tab are called the covariant components of the tensor, Tab the contravariant components of the tensor, and Tab, Tab the mixed components. We might note that we can write the metric tensor itself as:



And of course gab = gab = δab = δab, which are all the same thing. We may obtain the components of the tensor from the inner product as was done for vectors. For instance,



Anyway, since **T** is by assumption a scalar, this means that **T**, expressed in two different coordinate systems, must be invariant. And so for instance,



And so we must have the following transformation law for covariant tensor components:



For example, what the first one means is that the prospective tensor components T´αβ, constructed in coordinate system u´α and evaluated at point P is equal to XaαXbβTab, where Tab are the prospective tensor components constructed in coordinate system ua and evaluated at the same point P. Of course the X terms are also evaluated at the same point P as well.

Again, we can use the resolution of identity deal to come to the same conclusion:



And like with the vector, we can get the components of the tensor in different bases using the metric – as you can prove at your leisure using the same methodology as above.



**4d. Tensor Properties**

There are some worthwhile identities to consider. Many involve constructing scalars/vectors/tensors from other scalars/vectors/tensors. For instance,

**Vector contraction**

We can produce scalars out of vector components via a process called vector contraction. For instance, suppose we have two vectors components Sa and Tb, then the contraction of S and T is given by SaTa = SaTa. This contraction forms a scalar. To prove it observe that:



So there we go. Actually, are we sure ∂xb/∂xa = δba in general – isn’t it only when coordinate system is orthogonal? Well maybe not, because even in the non-orthogonal coordinate system I played with above, one can vary u and v independently cause when you go along the v axis, u isn’t changing. It just seems like it because you’re mistakenly viewing the projection orthogonally.

**Tensor contraction**

Similar to with vectors, tensor component contraction will result in other tensor components/vector components/scalars. Consider the contraction of a tensor and vector.



This is a contravariant component vector, since:



So as you can see it does transform as a contravariant vector component. Generally any tensor contraction will form a lower order tensor.

**Vector product**

There is a sort of outer product identity too. Let’s show that if Sa and Tb are contravariant vector components, then SaTb form the contravariant components of a tensor. So we have:



which is the transformation law of the contravariant components of a tensor. This makes sense because the outer product of two vectors **S** and **T** is.



and so SaTb certainly form the contravariant components of a tensor. But be careful here. We cannot go backwards and presume that any old combination SaTb**e**a**e**b is a tensor. We’d have to know how SaTb transforms. But if we know that ST is a tensor, then we can conclude that SaTb do transform as components of one. Another note of caution…the outer product example must be interpreted correctly when applied to operators. So note that Ta ∂φ/∂ub outerproduct would be a tensor (presuming φ a scalar). We check the transformation law:



So this always forms a tensor. But if operator applied to the tensor, then not so. Check if ∂Ta/∂ub forms the mixed components of a tensor? We check the transformation law:



So this would form a tensor, only for a linear coordinate transformation. Thus this entity would be called a pseudo-tensor since it is a tensor only for a limited set of coordinate transformations. This makes sense because when we try to form the entity with these components we get:



so we see that the thing under consideration, the coefficient of the first term, is only part of a tensor, not a tensor itself. So note that not everything of the form τab**e**a**e**b, etc., is a tensor.

**Vector division**

This is kind of misnomer, but the idea is this. Suppose we have a quantity qα and an arbitrary vector Ta. Then if qaTa is a scalar, then qa are the contravariant components of a vector. We can prove this as follows. We have:



Now since Tb is arbitrary, we must have that:



which is the vector transformation law. So our theorem is proved.

**Tensor division**

Just like we can prove the vector nature of a set of components by contraction, we can do the same with tensors. For instance if we contract qab with some completely arbitrary vector Sa and get another vector Tb, then qab forms the contravariant components of a tensor.

**Tensor identities**

Consider a tensor identity true in one frame of reference:



then it must be true in all frames of references. For instance, just multiply both sides by XaαXbβ and we get:



Now this wouldn’t have worked if we had an inhomogeneous term in the equation, like,



‘cause then fab would change in every reference frame. This feature is nice since we can search for identities in easy coordinate systems (like flat ones), and then generalize to all coordinate systems, as appropriate.

**4e. Special tensor δab**

Let’s discuss the special tensor (components) δab. Its transformation law is that it is diagonal in every coordinate system I’d say. So in u´α coordinates it’s also δαβ. First let’s verify that it is a tensor. Well then, forming it in both coordinate systems uα and ua we have:



so it checks out. Actually, do we know for sure that ∂ua/∂ub = δab? What if coordinate system isn’t orthogonal? I think it’s okay.

**4f. Special pseudo-tensor εabc**

Let’s introduce the anti-symmetric tensor components (really pseudo-tensor) εabc. It is defined as:



Is this also a tensor component, meaning in every frame of reference, it’s equal to that stuff above? This would require that the following is also true then.



But turns out this isn’t generally true as there is the following identity.



(to be clear, εαβγ is also 1 for even permutations of αβγ, -1 for odd permutations, and 0 otherwise, just as εabc is 1 for even permutations of abc, -1 for odd permutations, and 0 otherwise). So εαβγ would form the components of a tensor only under coordinate transformations with unit determinant (rotations would be one such, but not inversions say, since they’d have a -1 determinant, or stretches/compressions as they’d have a non-unit determinant). There are special relationships between εabc and δab. Some are as follows.



a special case is:



and another special case is:



We can use the anti-symmetric (pseudo) tensor to construct (pseudo) vectors, and to generally express and simplify cross-products. For instance, you can verify that:



and similarly,



(we’ll generalize this to cases where the basis vectors can depend on position later). We can simplify more complicated vector expressions rather quickly with this notation. So for instance we can write:



and we can prove that it is zero since:



and we can simplify expressions like,



**Example**

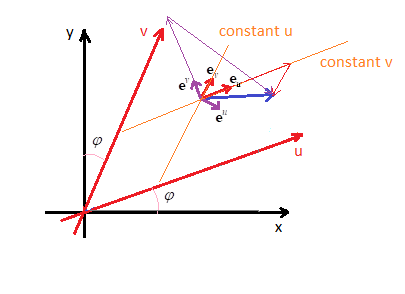
Say we have two vectors **A** = ai**e**i, and **B** = bj**e**j. And we want the cross product: **C** = **A**×**B**, defined as:



Is this invariant?



So yeah, forming the cross product from two tensors will only produce a tensor if the transformation has a unit determinant. But if I did the dot product between them, this would be invariant. Wanna be more concrete. Let’s go back to a previous example in the basis vectors file and introduce a coordinate system tilted inwards w/r to our original, and we’ll say the z axis is unaffected.



(I’m showing a blue vector, with its components (red) in the new covariant basis, and its components (purple) in the new contravariant basis. We said that the relationship between ‘unit’ vectors was (note they are not unit magnitude):



and the relationship between coordinates was:



So the basis vectors of the (u,v) system were:



Now for simplicity, let’s consider two vectors that just happened to be aligned with the axes of the new coordinate system,



Their cross product is in the original Cartesian system is:



Now let’s consider their components in the new system. Can plainly see that:



But just to verify,



so that checks out for **A**. Anyway, now forming the cross product, we have:



So these clearly don’t match.

**Example**

Let’s do the same thing, but with a more complicated coordinate transformation. Let’s go back to a previous example where the u and v axes made angles φu, and φv w/r to the x-axis, and were stretched out a distance du and dv respectively. The coordinate transformation was:



And recall the covariant, contravariant basis vectors were:



Now I’ll take the opposite simple tack from the last example, and say we have vectors that are aligned with our original coordinate axes.



Their cross product is:



And their components in the new system are:



and,



Their cross product in the u-v-w system is:



And this also doesn’t match, in general. But if:



then it will. And we can see that this the same as requiring the determinant of the coordinate transformation matrix X be 1. For instance, if φv – φu = π/2, and du = dv = 1, so that u-v-w is just a rotation of the x-y-z coordinate system, then we can see that we do get 1. But say we just inverted the x axis so that φu = π, and du = dv = 1. Then we still don’t get this equality. So in general, coordinate inversions don’t work either.