Intro to Tensors Notes

Jacob Shin

Contents

Chapter 0 Tensor Definitions Page 3

Chapter 1 Forward and Backwards Transformations Page 5

1.1 Forward Transformation 5
1.2 Backwards Transformation 5
1.3 Generalizing to N-Dimensions 6

Chapter 2 Vector Definition Page 8

Chapter 3 Vector Transformation Rules Page 9

3.1 Generalization to n-dimensions 9
3.2 Notation 10

Chapter 4 What are covectors? Page 11

4.1 Visualizing Covectors 11

Chapter 5 Covector Components Page 12

Chapter 6 Convector Transformation Rules Page 14

Chapter 7 Linear Maps Page 16

Chapter 8 Linear Map Transformation Rules Page 18

Chapter 9 The Metric Tensor Page 21

9.1 Lengths using the Metric Tensor 21
9.2 Angles using the Metric Tensor 22

9.3 How Metric Tensor Components Transform with a Change in Coordinates 22
9.4 Confirming Length Stays Constant 22
9.5 Summary 23
9.6 Revisiting the Coordinate Tensor Definition 23

Chapter 10 Bilinear Forms Page 25
10.1 Metric Tensor Review 25
10.2 Bilinear Forms 25
10.3 Properties of the Metric Tensor 26

Chapter 11 Linear Maps are Vector-Covector Pairs Page 27

Chapter 12 Bilinear Forms as Covector-Covector Pairs Page 29
12.1 Benefits of Tensor Product Definition for Linear Maps 29
12.2 Bilinear Forms as Covector-Covector Pairs 30

Chapter 13 Tensor Product vs. Kronecker Product Page 32
13.1 Definition of Tensor Product 32
13.2 Kronecker Product 33
13.3 Relation between Tensor Product and Kronecker Product 33

Chapter 14 Tensors as General Vector/Covector Combinations Page 34
14.1 Review of Tensors Written using Tensor Product 34
14.2 New Tensors 34
What are the transformation rules? — 34 • What is the Multiplication Rule for $Q(D)$ — 35 • What are the array shapes for D and Q — 36

Chapter 15 Tensor Product Spaces Page 37

Chapter 16 Raising/Lowering Indexes Page 39

Chapter 0

Tensor Definitions

Based on the video series by eigenchris

Definition 0.0.1: Tensors as multidimensional arrays

Examples:

1. Rank 0 Tensor: **Scalar**

$$[4] \text{ or } 4$$

2. Rank 1 Tensor: **Vector**

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

3. Rank 2 Tensor: **2D-Matrix**

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

4. Rank 3 Tensor: **3D-Matrix** "Cube" of values

Although tensors can be *represented* as matrices/arrays, this definition doesn't describe what tensors *actually* are since this definition doesn't explain the geometric meaning of tensors.

Definition 0.0.2: Tensors as objects invariant under a change in coordinates

1. Tensors have **components** that change in a **predictable** way when the coordinates are changed
2. Vectors are **invariant**, but vector components are not
3. Example of something invariant under coordinate transformation: length
4. Converting tensor components from one coordinate system to another is called a **Forward Transformation**, while doing the reverse if **Backwards Transformation**

Definition 0.0.3: Tensors as a combination of vectors and convectors combined using the tensor product

Best definition, but a bit abstract.

Definition 0.0.4: Tensors as partial derivatives and gradients that transform with the Jacobean matrix

Chapter 1

Forward and Backwards Transformations

Old Basis:

$$\{\vec{e_1}, \vec{e_2}\}$$

New Basis:

$$\{\tilde{\vec{e}}_1, \tilde{\vec{e}}_2\}$$

1.1 Forward Transformation

Convert from the old basis to the new basis:

$$\tilde{\vec{e}}_1 = c_1\vec{e_1} + c_2\vec{e_2}$$
$$\tilde{\vec{e}}_2 = c_3\vec{e_1} + c_4\vec{e_2}$$

where $c_1, c_2, c_3,$ and c_4 are scalar constants.

We can rewrite the above linear equations in matrix form by defining a **forward transformation matrix**, F

$$F = \begin{bmatrix} c_1 & c_3 \\ c_2 & c_4 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\vec{e}}_1 & \tilde{\vec{e}}_2 \end{bmatrix} = \begin{bmatrix} \vec{e_1} & \vec{e_2} \end{bmatrix} F = \begin{bmatrix} \vec{e_1} & \vec{e_2} \end{bmatrix} \begin{bmatrix} c_1 & c_3 \\ c_2 & c_4 \end{bmatrix}$$

Note that the F matrix is flipped along the diagonal (transposed) from what we'd get if we multiplied the forward transformation matrix before the old basis.

$$\tilde{F} = F^T = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\vec{e}}_1 \\ \tilde{\vec{e}}_2 \end{bmatrix} = \tilde{F} \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \end{bmatrix}$$

1.2 Backwards Transformation

Similarly, we can convert from the new basis to the old basis with a **backwards transformation**

$$\vec{e_1} = a_1\tilde{\vec{e}}_1 + a_2\tilde{\vec{e}}_2$$
$$\vec{e_2} = a_3\tilde{\vec{e}}_1 + a_4\tilde{\vec{e}}_2$$

where $a_1, a_2, a_3,$ and a_4 are scalar constants.

Rewrite in terms of the backwards transformation matrix, B

$$B = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix}$$

$$\begin{bmatrix} \vec{e_1} & \vec{e_2} \end{bmatrix} = \begin{bmatrix} \tilde{\vec{e}}_1 & \tilde{\vec{e}}_2 \end{bmatrix} B = \begin{bmatrix} \tilde{\vec{e}}_1 & \tilde{\vec{e}}_2 \end{bmatrix} \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix}$$

If we multiply F and B we get the identity matrix:

$$BF = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Thus one is the inverse of the other:

$$B = F^{-1}$$

1.3 Generalizing to N-Dimensions

We can generalize to n dimensions

$$F = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ F_{n1} & F_{n2} & \dots & F_{nn} \end{bmatrix}$$

$$\begin{aligned} \tilde{\vec{e}}_1 &= F_{11}\vec{e_1} + F_{21}\vec{e_2} + \ldots + F_{n1}\vec{e_n} \\ \tilde{\vec{e}}_2 &= F_{12}\vec{e_1} + F_{22}\vec{e_2} + \ldots + F_{n2}\vec{e_n} \\ &\vdots \\ \tilde{\vec{e}}_n &= F_{1n}\vec{e_1} + F_{2n}\vec{e_2} + \ldots + F_{nn}\vec{e_n} \end{aligned}$$

This can be written more simply as:

$$\tilde{\vec{e}}_i = \sum_{j=1}^{n} F_{ji}\vec{e_j}$$

Similarly we have

$$\vec{e_i} = \sum_{j=1}^{n} B_{ji}\tilde{\vec{e}}_j$$

We still have that F and B are inverses

$$FB = \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & & & \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Proof:

$$\vec{e_i} = \sum_{j=1}^{n} B_{ji}\tilde{\vec{e}}_j$$

$$\vec{e_i} = \sum_{j=1}^{n} B_{ji}\left(\sum_{k=1}^{n} F_{kj}\vec{e_k}\right) = \sum_{j=1}^{n}\left(\sum_{k=1}^{n} F_{kj}B_{ji}\vec{e_k}\right) = \sum_{k=1}^{n}\left(\sum_{j=1}^{n} F_{kj}B_{ji}\vec{e_k}\right)$$

Note that we want $\vec{e_1} = \vec{e_1}$, $\vec{e_2} = \vec{e_2}$, etc. so that implies that

$$\sum_j F_{kj} B_{ji} = \delta_{ki} = \begin{cases} 0 & i \neq k \\ 1 & i = k \end{cases}$$

which means that the matrix FB has 1's on the diagonals and 0's elsewhere (identity matrix).

□

Chapter 2

Vector Definition

A vector is an example of a tensor.

Definition 2.0.1: Vector Definition

1. Naive definition: Array of numbers
   - The list of numbers are the *components* of the vector, not the vector itself. Remember that vectors are invariant under a coordinate transformation while the components are not.
2. Arrow in space
   - A bit better than the above definition
   - Not all vectors can be visualized as arrows (like functions!)
   - An arrow is just a special type of vector: A *Euclidean vector*
3. Member of a Vector Space

Definition 2.0.2: Vector Space

$$\left(\underbrace{V}_{\text{Set of vectors}} , \underbrace{S}_{\text{Set of Scalars}} , \underbrace{+}_{\text{Vector addition}} , \underbrace{\cdot}_{Vectorscalingoperator} \right)$$

Vectors can be

- added together +
- Multiplied by a scalar ·

For these notes, when we say *vector*, we mean a euclidean vector.

Chapter 3

Vector Transformation Rules

To convert the vector components from the old basis $\{\vec{e_1}, \vec{e_2}\}$ to components in the new basis, $\left\{\tilde{\vec{e_1}}, \tilde{\vec{e_2}}\right\}$, we apply the *backwards* transformation on the old basis components:

Vector, v, in the old basis:

$$\vec{v} = v_1\vec{e_1} + v_2\vec{e_2} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{\vec{e_i}}$$

In the new basis

$$\vec{v} = \tilde{v_1}\tilde{\vec{e_1}} + \tilde{v_2}\tilde{\vec{e_2}} = \begin{bmatrix} \tilde{v_1} \\ \tilde{v_2} \end{bmatrix}_{\tilde{\vec{e}}_i}$$

where b_1, b_2, b_3 and $, b_4$ are scalars.

Converting vector components in the old basis to new

$$B \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{\vec{e_i}} = \begin{bmatrix} \tilde{v_1} \\ \tilde{v_2} \end{bmatrix}_{\tilde{\vec{e}}_i}$$

When the basis vectors get larger, the "measuring stick" gets bigger, which means the components of the vector get smaller so that the length remains the same.

Similarly, when the basis vectors are rotated clockwise, from the perspective of the vector, the components have to be rotated counterclockwise to maintain the same orientation in space.

Since the vector components transform *contrary* to the basis vectors, we say that the vector components are *contravariant*.

3.1 Generalization to n-dimensions

Proof:

$$\vec{v} = \sum_{i=1}^{n} v_i\vec{e_i} = \sum_{j=1}^{n} \tilde{v_j}\tilde{\vec{e_j}}$$

$$\widetilde{\vec{e_j}} = \sum_{i=1}^{n} F_{ij}\vec{e_i}$$

$$\vec{e_j} = \sum_{i=1}^{n} B_{ij}\widetilde{\vec{e_i}}$$

$$\vec{v} = \sum_{j=1}^{n} \tilde{v_j}\tilde{\vec{e_j}} = \sum_{j=1}^{n} \tilde{v_j}\left(\sum_{i=1}^{n} F_{ij}\vec{e_i}\right) = \sum_{i=1}^{n}\left(\sum_{j=1}^{n} F_{ij}\widetilde{v_j}\vec{e_i}\right)$$

$$\vec{v_i} = \sum_{j=1}^{n} F_{ij}\widetilde{\vec{v_j}}$$

This proves that converting the vector components in the new basis to the old basis requires the forward transformation □

3.2 Notation

Since the vector components are contravariant (do the opposite of the basis vectors), we write the index for the vector components as a superscript:

$$\vec{v} = \sum_{i=1}^{n} v^i \vec{e}_i = \sum_{i=1}^{n} \widetilde{v^i} \widetilde{\vec{e}_i}$$

Chapter 4

What are covectors?

Covectors are our second example of a tensor.

You can think of covectors as row vector. Row vectors are not simply just column vectors flipped on its side (only true in *orthonormal basis*).

Covectors/Row vectors can both be thought of as functions that act on a vector:

$$\alpha : V \rightarrow \mathbb{R}$$

Covectors exhibit linearity:

$$\alpha\left(\vec{v} + \vec{w}\right) = \alpha(\vec{v}) + \alpha\left(\vec{w}\right)$$

$$\alpha(n\vec{v}) = n\alpha(\vec{v})$$

Covectors are also elements of a vector space, called the dual vector space, V^*

$$(n\alpha)(\vec{v}) = n\alpha(\vec{v})$$

$$(\beta + \alpha)(\vec{v}) = \beta(\vec{v}) + \alpha(\vec{v})$$

where α is the covector

4.1 Visualizing Covectors

Take the 2-D case:

$$\alpha = \begin{bmatrix} 2 & 1 \end{bmatrix}$$

$$\alpha(\vec{v}) = \begin{bmatrix} 2 & 1 \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) = 2x + 1y$$

We can visualize the covector by looking at the isolines (i.e. where $\alpha = k$, with k as a constant)

$$2x + 1y = 0$$

$$2x + 1y = 1$$

$$2x + 1y = 2$$

$$\vdots$$

2D Covectors form a "stack" of lines. Graphically, the value of α acting on $\vec{v}$ is the number of isolines that $\vec{v}$ pierces

Chapter 5

Covector Components

Like vectors, covectors are invariant, but their components are not.

Covectors form a vector space, V^*, and are not part of V, so we need new basis vectors, ϵ^1 and ϵ^2: $V \to \mathbb{R}$. They are defined as follows:

$$\epsilon^1\left(\vec{e}_1\right) = 1$$
$$\epsilon^2\left(\vec{e}_1\right) = 0$$
$$\epsilon^1\left(\vec{e}_2\right) = 0$$
$$\epsilon^2\left(\vec{e}_2\right) = 1$$

or simply

$$\epsilon^i\left(\vec{e}_j\right) = \delta_{ij}$$

Let's see what the two covector bases looks like when acting on a vector, $\vec{v}$

$$\epsilon^1\left(\vec{v}\right) = \epsilon^1\left(v^1\vec{e}_1 + v^2\vec{e}_2\right)$$

Since covectors are linear:

$$= \epsilon^1\left(v^1\vec{e}_1 + v^2\vec{e}_2\right) = v^1\epsilon^1\left(\vec{e}_1\right) + v^2\epsilon^1\left(\vec{e}_2\right) = v^1 \cdot 1 + v^2 \cdot 0 = v^1$$

Similarly,

$$\epsilon^2\left(\vec{v}\right) = v^2$$

Now consider a general covector, α acting on $\vec{v}$

$$\alpha\left(\vec{v}\right) = \alpha\left(v^1\vec{e}_1 + v^2\vec{e}_2\right) = v^1\alpha(e_1) + v^2\alpha(e_2)$$

Using the definitions of $\epsilon^1(\vec{v}) = v^1$ and $\epsilon^2(\vec{v}) = v^2$ gives:

$$\alpha(\vec{v}) = \epsilon^1\left(\vec{v}\right) \cdot \alpha\left(e_1\right) + \epsilon^2\left(\vec{v}\right) \cdot \alpha\left(e_2\right)$$

Let $\alpha_1 = \alpha(\vec{e}_1)$ and $\alpha_2 = \alpha(\vec{e}_2)$

$$\alpha\left(\vec{v}\right) = \epsilon^1\left(\vec{v}\right)a_1 + \epsilon^2\left(\vec{v}\right)a_2 = \left(a_1\epsilon^1 + a_2\epsilon^2\right)\left(\vec{v}\right)$$
$$\implies \alpha = \alpha_1\epsilon^1 + \alpha_2\epsilon^2$$

We see that we can rewrite any generic covector, α, as a linear combination of ϵ^1 and ϵ^2. Thus the covectors ϵ^1 and ϵ^2 form a *dual basis*.

To summarize, with any set of basis vectors ($\vec{e}_1$ and $\vec{e}_2$), we can define basis convectors (ϵ^1 and ϵ^2). Then we can represent any convector as a linear combination of these basis convectors.

To convert between components of a convector from an old dual basis to a new dual basis, you use the *Forward matrix*

$$\begin{bmatrix}\alpha_1 & \alpha_2\end{bmatrix}_{\epsilon^i} F = \begin{bmatrix}\widetilde{\alpha_1} & \widetilde{\alpha_2}\end{bmatrix}_{\widetilde{\epsilon^i}}$$

Converting from the new basis to the old basis requires the *Backwards matrix*:

$$\begin{bmatrix}\widetilde{\alpha_1} & \widetilde{\alpha_2}\end{bmatrix}_{\widetilde{\epsilon^i}} B = \begin{bmatrix}\alpha_1 & \alpha_2\end{bmatrix}_{\epsilon^i}$$

Note this is the opposite compared to what happens for a change of basis for normal vector components.

This shows that the vector components are not always the same as convector components. See the eigenchris' video for examples.

Chapter 6

Convector Transformation Rules

We now know how covector components transform in 2D. This chapter will show how to go from an old dual vector basis to a new dual basis. Then we will use this result to prove how covector components transform in N-dimension.

We want to find the matrix, Q, where

$$\widetilde{\epsilon^1} = Q_{11}\epsilon^1 + Q_{12}\epsilon^2$$
$$\widetilde{\epsilon^2} = Q_{21}\epsilon^1 + Q_{22}\epsilon^2$$

Claim 6.0.1 Dual Basis Transformations

$$\widetilde{\epsilon^i} = \sum_{j=1}^{n} B_{ij}\epsilon^j \qquad \epsilon^i = \sum_{j=1}^{n} F_{ij}\widetilde{\epsilon^j}$$

Proof:

$$\widetilde{\vec{e}_j} = \sum_{i=1}^{n} F_{ij}\vec{e}_i \qquad \vec{e}_j = \sum_{i=1}^{n} B_{ij}\widetilde{\vec{e}_i}$$

$$\epsilon^i\left(\vec{e}_j\right) = \delta_{ij} \qquad \widetilde{\epsilon^i}\left(\widetilde{\vec{e}_j}\right) = \delta_{ij}$$

$$\widetilde{\epsilon^i} = \sum_{j=1}^{n} Q_{ij}\epsilon^j$$

$$\implies \widetilde{\epsilon^i}\left(\widetilde{\vec{e}_k}\right) = \sum_{j=1}^{n} Q_{ij}\epsilon^j\left(\widetilde{\vec{e}_k}\right) = \delta_{ij} = \sum_{j=1}^{n} Q_{ij}\epsilon^j\left(\sum_{l=1}^{n} F_{lk}\vec{e}_l\right)$$

$$\delta_{ij} = \sum_{j=1}^{n}\sum_{l=1}^{n} F_{lk}Q_{ij}\epsilon^j\left(\vec{e}_l\right) = \sum_{j=1}^{n}\sum_{l=1}^{n} F_{lk}Q_{ij}\delta_{lk}$$

$$\implies \delta_{ij} = F_{jk}Q_{ij}$$

$$\implies Q = F^{-1}$$

Since F only has one inverse this means

$$Q = B \implies Q_{ij} = B_{ij}$$

$$\widetilde{\epsilon^i} = \sum_{j=1}^{n} B_{ij}\epsilon^j$$

□

The proof for the inverse process is similar to the above. These results explain why we write the dual basis vectors as ϵ^i with a superscript, since they transform opposite the way euclidean basis vectors do.

Using these results, we can find out how covector components transform in n-dimensions

Claim 6.0.2 Covector Component Transformations in n-dimensions

$$\widetilde{\alpha_j} = \sum_{i=1}^{n} F_{ij}\alpha_i \qquad \alpha_j = \sum_{i=1}^{n} B_{ij}\widetilde{\alpha_i}$$

Proof:

$$\widetilde{\epsilon^i} = \sum_{j=1}^{n} B_{ij}\epsilon^j \qquad \epsilon^i = \sum_{j=1}^{n} F_{ij}\widetilde{\epsilon^j}$$

$$\alpha = \sum_{i=1}^{n} \alpha_i \epsilon^i = \sum_{j=1}^{n} \widetilde{\alpha_j}\widetilde{\epsilon^j}$$

$$\alpha = \sum_{i=1}^{n} \alpha_i \left(\sum_{j=1}^{n} F_{ij}\widetilde{\epsilon^j} \right) = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} F_{ij}\alpha_i \right) \widetilde{\epsilon^j}$$

$$\implies \sum_{i=1}^{n} F_{ij}\alpha_i = \widetilde{\alpha_j}$$

□

As expected, the covector components transform in the same way as euclidean basis vectors, which is why a subscript is used (α_i).

Note:-

Summary of Transformation Rules

$$\widetilde{\vec{e}_j} = \sum_{i=1}^{n} F_{ij}\vec{e}_i \qquad \vec{e}_j = \sum_{i=1}^{n} B_{ij}\widetilde{\vec{e}_i} \qquad \text{Covariant}$$

$$\widetilde{\vec{v}^i} = \sum_{j=1}^{n} B_{ij}v^j \qquad \vec{v}^i = \sum_{j=1}^{n} F_{ij}\widetilde{v^j} \qquad \text{Contravariant}$$

$$\widetilde{\epsilon^i} = \sum_{j=1}^{n} B_{ij}\epsilon^j \qquad \epsilon^i = \sum_{j=1}^{n} F_{ij}\widetilde{\epsilon^j} \qquad \text{Contravariant}$$

$$\widetilde{\alpha_j} = \sum_{i=1}^{n} F_{ij}\alpha_i \qquad \alpha_j = \sum_{i=1}^{n} B_{ij}\widetilde{\alpha_i} \qquad \text{Covariant}$$

Chapter 7

Linear Maps

Definition 7.0.1: Array Definition

Matrices can be used to represent linear maps, just like how column vectors can be used to represent euclidean vectors and row vectors can be used to represent covectors.
Linear maps transform the input vectors, but they *do not* transform basis vectors.

Definition 7.0.2: Coordinate Definition

Linear maps are spatial transforms that

1. Keep gridlines parallel
2. Keep gridlines evenly spaced
3. Keep the origin stationary

Translations are not true linear maps under this definition.

Definition 7.0.3: Abstract Definition

1. A function that maps $V \to W$, where V and W are vector spaces
2. Adds inputs and outputs: $L\left(\vec{v} + \vec{w}\right) = L(\vec{v}) + L\left(\vec{w}\right)$
3. Scale the input and outputs: $L(n\vec{v}) = nL(\vec{v})$

Linear maps have linearity like covectors, except in this case, the output of the function is a vector and not a scalar. Linearity should not be confused with closure under multiplication by scalars and closure under addition of vectors. Linearity applies to *functions* and their inputs/outputs.

Example 7.0.1 (Where do we get Matrix Multiplication?)
Consider a linear map, $L : V \to V$, acting on a vector, $\vec{v}$

$$\vec{w} = L\left(\vec{v}\right) = L\left(v^1\vec{e}_1 + v^2\vec{e}_2\right) = v^1 L(\vec{e}_1) + v^2 L\left(\vec{e}_2\right)$$

Let us define the output of the linear map as follows:

$$L(\vec{e}_1) = L_1^1\vec{e}_1 + L_2^1\vec{e}_2$$

$$L(\vec{e}_2) = L_1^2\vec{e}_1 + L_2^2\vec{e}_2$$

we can do this since we know that the output vector of the linear map is an element of the vector space, V, which means we can write the output vector as a linear combination of the basis vectors, $\vec{e}_1$ and $\vec{e}_2$.

$$\vec{w} = v^1\left(L_1^1\vec{e}_1 + L_1^2\vec{e}_2\right) + v^2\left(L_2^1\vec{e}_1 + L_2^2\vec{e}_2\right)$$
$$\vec{w} = \vec{e}_1\underbrace{\left(L_1^1v^1 + L_2^1v^2\right)}_{w^1} + \vec{e}_2\underbrace{\left(L_1^2v^1 + L_2^2v^2\right)}_{w^2}$$

Thus we have

$$w^1 = L_1^1v^1 + L_2^1v^2$$
$$w^2 = L_1^2v^1 + L_2^2v^2$$

Notice this is the formula we get when multiplying the matrix, L with the vector, v, in 2D. This is where we get the way we multiply matrices.

In n-dimensions we have

$$\vec{w} = \sum_{i=1}^{n} w^i\vec{e}_i$$
$$L\left(\vec{e}_i\right) = \sum_{j=1}^{n} L_i^j\vec{e}_j$$

Thus the matrix multiplication rule we get is:

$$w^i = \sum_{j=1}^{n} L_j^i v^j$$

Chapter 8

Linear Map Transformation Rules

We want to be able to do a change of basis on $\vec{e}_i$ to get $\widetilde{\vec{e}}_i$ and compute the linear map on those vectors:

In the original basis

$$L\left(\vec{e}_i\right) = \sum_{k=1}^{n} L_i^k \vec{e}_k$$

In the new basis

$$L\left(\widetilde{\vec{e}}_i\right) = \sum_{q=1}^{n} \widetilde{L}_i^q \widetilde{\vec{e}}_q$$

We want to find what these $\widetilde{L}_i^q$ coefficients are

Claim 8.0.1 Linear Map Transformation

$$\widetilde{L}_i^q = \sum_{j=1}^{n} \sum_{k=1}^{n} B_k^q L_j^k F_i^j$$

Proof: We have the basis vector transformation rules:

$$\widetilde{\vec{e}}_i = \sum_{j=1}^{n} F_i^j \vec{e}_j \qquad \vec{e}_k = \sum_{l=1}^{n} B_k^l \widetilde{\vec{e}}_l$$

$$L\left(\widetilde{\vec{e}}_i\right) = \sum_{q=1}^{n} \widetilde{L}_i^q \widetilde{\vec{e}}_q$$

$$L\left(\widetilde{\vec{e}}_i\right) = L\left(\sum_{j=1}^{n} F_i^j \vec{e}_j\right)$$

$$\implies \sum_{q=1}^{n} \widetilde{L}_i^q \widetilde{\vec{e}}_q = \sum_{j=1}^{n} F_i^j L\left(\vec{e}_j\right)$$

Use the definition of L applied on $\vec{e}_i$

$$\sum_{q=1}^{n} \widetilde{L}_i^q \widetilde{\vec{e}}_q = \sum_{j=1}^{n} F_i^j \sum_{k=1}^{n} L_j^k \vec{e}_k$$

Now get the RHS in terms of the new basis vectors, $\widetilde{\vec{e}}_k$ using the backwards transformation:

$$\sum_{q=1}^{n} \widetilde{L}_i^q \widetilde{\vec{e}}_q = \sum_{j=1}^{n} F_i^j \sum_{k=1}^{n} L_j^k \sum_{l=1}^{n} B_k^l \widetilde{\vec{e}}_l$$

Rearranging gives

$$\sum_{q=1}^{n} \widetilde{L_i^q} \widetilde{\vec{e}_q} = \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} F_i^j L_j^k B_k^l \widetilde{\vec{e}_l}$$

Rewriting $q = l$

$$\sum_{q=1}^{n} \widetilde{L_i^q} \widetilde{\vec{e}_q} = \sum_{q=1}^{n} \left(\sum_{j=1}^{n} \sum_{k=1}^{n} F_i^j L_j^k B_k^q \right) \widetilde{\vec{e}_q}$$

Thus we arrive at the final expression for the new linear map matrix:

$$\widetilde{L_i^q} = \sum_{j=1}^{n} \sum_{k=1}^{n} F_i^j L_j^k B_k^q$$

□

In einstein notation, we can rewrite this as:

$$\widetilde{L_i^q} = F_i^{\boldsymbol{j}} L_{\boldsymbol{j}}^{\boldsymbol{k}} B_{\boldsymbol{k}}^q$$

The bolded letters are the indices we sum over. Whenever we have the superscript followed by the same subscript (or vice versa), we sum over that letter.

Similarly, using Einstein notation we have:

$$\widetilde{\vec{e}_i} = F_i^{\boldsymbol{j}} \vec{e}_{\boldsymbol{j}} \qquad \vec{e}_k = B_k^{\boldsymbol{l}} \vec{e}_{\boldsymbol{l}}$$

Example 8.0.1 (Multiplying by Identity Matrix with Einstein Notation)

If we have a matrix M

$$MI = M$$

$$\implies (MI)_i^k = \sum_{j=1}^{n} M_i^j \delta_j^k = M_i^{\boldsymbol{j}} \delta_{\boldsymbol{j}}^k = M_i^k$$

From this example, we can see that δ_j^k acts to cancel consecutive summation indices.

Proof: Now we will prove how to go from the new basis to the old basis for linear maps:

We have the following equality:

$$\delta_y^x = B_y^z F_z^x$$

From the definition of the first transformation matrix we got:

$$\widetilde{L_i^l} = B_k^l L_j^k F_i^j$$

Multiply both sides with F and B

$$F_l^s \widetilde{L_i^l} B_t^i = F_l^s B_k^l L_j^k F_i^j B_t^i$$

$$F_l^s \widetilde{L_i^l} B_t^i = \delta_k^s L_j^k \delta_t^j$$

$$F_l^s \widetilde{L_i^l} B_t^i = L_t^s$$

□

Definition 8.0.1: Contravariant (1, 0)-Tensors

$$\widetilde{\epsilon^i} = B_j^i \epsilon^j \qquad \epsilon^i = F_j^i \widetilde{\epsilon^j}$$

$$\widetilde{v^i} = B_j^i v^j \qquad v^i = F_j^i \widetilde{v^j}$$

Definition 8.0.2: Covariant (1, 0)-Tensors

$$\widetilde{\vec{e}}_j = F^i_j \vec{e}_i \qquad \vec{e}_j = B^i_j \widetilde{\vec{e}}_i$$

$$\widetilde{\alpha}_j = F^i_j \alpha_i \qquad \alpha_j = B^i_j \widetilde{\alpha}_i$$

Definition 8.0.3: (1, 1) Tensor

$$\widetilde{L}^i_j = B^i_k L^k_l F^l_j \qquad L^i_j = F^i_k \widetilde{L}^k_l B^l_j$$

Since this tensor uses both the forward and backwards matrices, it is called a $(1,1)$-tensor.

Chapter 9

The Metric Tensor

9.1 Lengths using the Metric Tensor

When getting the length of a vector, we usually just use the pythagorean theorem. However, the pythagorean theoerem assumes an *orthonormal basis*, since assume we can make a right triangle with the basis vectors and the vector we're getting the length of.

The pythagorean theorem only works for basis vectors of length 1 that are orthgonal.

The real way to compute length is with the dot product:

$$\|v\|^2 = \vec{v} \cdot \vec{v} = \left(v^1\right)^2 \left(\vec{e}_1 \cdot \vec{e}_1\right) + 2v^1v^2 \left(\vec{e}_1 \cdot \vec{e}_2\right) + \left(v^2\right)^2 \left(\vec{e}_2 \cdot \vec{e}_2\right)$$

Note that in the case of an orthonormal basis,

$$\delta_{ij} = \vec{e}_i \cdot \vec{e}_j$$

which means we just get the pythagorean theorem.

Example 9.1.1 (Metric Tensor)

Consider a vector, v, in an old basis, $\vec{e}_i$, and in a new basis, $\widetilde{\vec{e}_i}$. Let the old basis be orthonormal and the following be true for the new basis:

$$\widetilde{\vec{e}_1} \cdot \widetilde{\vec{e}_1} = 5, \qquad \widetilde{\vec{e}_1} \cdot \widetilde{\vec{e}_2} = -\frac{3}{4}, \qquad \widetilde{\vec{e}_1} \cdot \widetilde{\vec{e}_2} = \frac{5}{16}$$

In the old basis the length squared is

$$\|\vec{v}\| = \left(v^1\right)^2 + \left(v^2\right)^2$$

and in the new basis:

$$\|\vec{v}\| = 5\left(\widetilde{v^1}\right)^2 + 2\left(-\frac{3}{4}\right)\widetilde{v^1}\widetilde{v^2} + \frac{5}{16}\left(\widetilde{v^2}\right)^2$$

These two equations can be rewritten in matrix form:

$$\|v\|^2 = \begin{bmatrix} v^1 & v^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}$$

$$\|v\|^2 = \begin{bmatrix} v^1 & v^2 \end{bmatrix} \begin{bmatrix} 5 & -\frac{3}{4} \\ -\frac{3}{4} & \frac{5}{16} \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}$$

The matrices in the middle are called the **metric tensor**, denoted by g

$$g_{\vec{e}_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{\vec{e}_i}$$

$$g_{\widetilde{\vec{e}_i}} = \begin{bmatrix} 5 & -\frac{3}{4} \\ -\frac{3}{4} & \frac{5}{16} \end{bmatrix}_{\widetilde{\vec{e}_i}}$$

Note these are the same metric tensor, but represented with a different basis

The length of a vector can then be rewritten in terms of the *metric tensor*:

$$\|\vec{v}\|^2 = v^i v^j \left(\vec{e}_i \cdot \vec{e}_j\right) = v^i v^j g_{ij} = v^i g_{ij} v^j$$

Note the use of einstein notation with an implied summation over i and j in the above equation

$$g_{ij} = \vec{e}_i \cdot \vec{e}_j$$

9.2 Angles using the Metric Tensor

The metric tensor can also be used to compute the angle between two vectors, $\vec{w}$ and $\vec{v}$. Create a unit basis vectors that is in the dirction of $\vec{v}$ and another unit basis vector in the direction of $\vec{w}$:

$$\begin{gathered}
\vec{e_1} \cdot \vec{e_1} = 1 \\
\vec{e_2} \cdot \vec{e_2} = 1 \\
\vec{e_1} \cdot \vec{e_2} = \cos\theta \\
\implies \vec{v} \cdot \vec{w} = \|\vec{v}\|\|\vec{w}\| \cos\theta \\
\frac{\left(\vec{v} \cdot \vec{w}\right)}{\|\vec{v}\|\|\vec{w}\|} = \cos\theta
\end{gathered}$$

$$\begin{gathered}
\vec{v} \cdot \vec{w} = \left(v^1 \vec{e_1} + v^2 \vec{e_2}\right) \cdot \left(w^1 \vec{e_1} + w^2 \vec{e_2}\right) \\
\vec{v} \cdot \vec{w} = v^1 w^1 \left(\vec{e_1} \cdot \vec{e_1}\right) + v^1 w^2 \left(\vec{e_1} \cdot \vec{e_2}\right) + v^2 w^1 \left(\vec{e_2} \cdot \vec{e_1}\right) + v^2 w^2 \left(\vec{e_2} \cdot \vec{e_2}\right) \\
\vec{v} \cdot \vec{w} = v^1 w^1 g_{11} + v^1 w^2 g_{12} + v^2 w^1 g_{21} + v^2 w^2 g_{22} \\
\vec{v} \cdot \vec{w} = v^i w^i g_{ij}
\end{gathered}$$

9.3 How Metric Tensor Components Transform with a Change in Coordinates

Converting from old to new

$$\widetilde{\vec{e}_j} = F^i_j \vec{e}_i \qquad \vec{e}_j = B^i_j \widetilde{\vec{e}_i}$$

$$\begin{gathered}
\widetilde{g_{ij}} = \widetilde{\vec{e}_i} \cdot \widetilde{\vec{e}_j} = F^k_i \vec{e}_k \cdot F^l_j \vec{e}_l \\
\widetilde{g_{ij}} = F^k_i F^l_j \left(\vec{e}_k \cdot \vec{e}_l\right) \\
\widetilde{g_{ij}} = F^k_i F^l_j g_{kl} = F^k_i g_{kl} F^l_j
\end{gathered}$$

Similarly from new to old:

$$g_{kl} = B^i_k B^j_l \widetilde{g_{ij}}$$

9.4 Confirming Length Stays Constant

We have the transformation rules for vectors

$$\widetilde{v^i} = B^i_j v^j \qquad v^i = F^i_j \widetilde{v^j}$$

and the transformation rules for the metric tensor

$$\widetilde{g_{ij}} = F^k_i F^l_j g_{kl} \qquad g_{kl} = B^i_k B^j_l \widetilde{g_{ij}}$$

$$\|v\|^2 = \widetilde{v^i} \widetilde{v^j} \widetilde{g_{ij}}$$

$$\begin{gathered}
\|v\|^2 = B^i_a v^a B^j_b v^b \left(F^k_i F^l_j g_{kl}\right) = v^a v^b \delta^k_a \delta^l_b g_{kl} \\
\|v\|^2 = v^a v^b g_{ab}
\end{gathered}$$

9.5 Summary

Definition 9.5.1: Contravariant (1, 0)-Tensors

$$\widetilde{\epsilon^i} = B^i_j \epsilon^j \qquad \epsilon^i = F^i_j \widetilde{\epsilon^j}$$

$$\widetilde{v^i} = B^i_j v^j \qquad v^i = F^i_j \widetilde{v^j}$$

Definition 9.5.2: Covariant (1, 0)-Tensors

$$\widetilde{\vec{e}_j} = F^i_j \vec{e}_i \qquad \vec{e}_j = B^i_j \widetilde{\vec{e}_i}$$

$$\widetilde{\alpha_j} = F^i_j \alpha_i \qquad \alpha_j = B^i_j \widetilde{\alpha_i}$$

Definition 9.5.3: (1, 1) Tensor

$$\widetilde{L^i_j} = B^i_k L^k_l F^l_j \qquad L^i_j = F^i_k \widetilde{L^k_l} B^l_j$$

Since this tensor uses both the forward and backwards matrices, it is called a $(1,1)$-tensor.

Definition 9.5.4: (2, 0)-Tensor

$$\widetilde{g_{ij}} = F^k_i F^l_j g_{lk}$$

$$g_{lk} = B^i_k B^j_l \widetilde{g_{ij}}$$

Called the $(2,0)$-tensor since the metric tensor uses two covariant rules

9.6 Revisiting the Coordinate Tensor Definition

Earlier we gave a coordinate definition of the tensor:

Definition 9.6.1: Tensors as objects invariant under a change in coordinates

1. Tensors have **components** that change in a **predictable** way when the coordinates are changed
2. Vectors are **invariant**, but vector components are not
3. Example of something invariant under coordinate transformation: length
4. Converting tensor components from one coordinate system to another is called a **Forward Transformation**, while doing the reverse if **Backwards Transformation**

Now we can explain the **predictable** way that tensor components transform. Suppose we have a tensor, T

$$T^{ijk\ldots}_{rst\ldots}$$

$ijk\ldots$ are the contravariant components while $rst\ldots$ are the covariant parts of the tensor, T
The **predictabe** way tensors transform is defined by the following:

$$\widetilde{T^{abc}_{xyz}} = \left(B^a_i B^b_j B^c_k \ldots\right) T^{ijk}_{rst} \left(F^r_x F^s_y F^t_z\right)$$

$$T^{ijk}_{rst} = \left(F^i_a F^j_b F^k_c \ldots\right) \widetilde{T^{abc}_{xyz}} \left(B^x_r B^y_s B^z_t \ldots\right)$$

When a tensor has m contravariant components and n covariant components, the tensor is called a (m, n)-tensor.

Chapter 10

Bilinear Forms

Another type of tensor is a bilinear form. A metric tensor is a special type of bilinear form.

10.1 Metric Tensor Review

Note that since $g_{ij} = \vec{e}_i \cdot \vec{e}_j$, and the dot product is commutative, this means the metric tensor is symmetric.

You can think of the metric tensor, g, as a function $g : V \times V \to \mathbb{R}$

$$g\left(\vec{v}, \vec{w}\right) \mapsto v^i w^i g_{ij}$$

Multiplying by a means you can move the a to one of the inputs, but not both:

$$ag(\vec{v}, \vec{w}) = g\left(\vec{v}\right), a\vec{w}) = g(a\vec{v}, \vec{w})$$

$$ag(\vec{v}, \vec{w}) \neq g(a\vec{v}, a\vec{w})$$

Adding a vector to one of the inputs:

$$\left(\begin{bmatrix} v^1 & v^2 \end{bmatrix} + \begin{bmatrix} u^1 & u^2 \end{bmatrix}\right) \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$$

$$\begin{bmatrix} v^1 & v^2 \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} w^1 \\ w^2 \end{bmatrix} + \begin{bmatrix} u^1 & u^2 \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$$

$$\left(v^i + u^i\right) w^j g_{ij} = v^i w^j g_{ij} + u^i w^j g_{ij}$$

$$\implies g\left(\vec{v} + \vec{u}, \vec{w}\right) = g(\vec{v}, \vec{w}) + g(\vec{u}, \vec{w})$$

$$g\left(\vec{v}, \vec{w} + \vec{t}\right) = g\left(\vec{v}, \vec{w}\right) + g\left(\vec{v}, \vec{t}\right)$$

$$g\left(\vec{v} + \vec{u}, \vec{w} + \vec{t}\right) = g\left(\vec{v}, \vec{w}\right) + g\left(\vec{u}, \vec{w}\right) + g\left(\vec{v}, \vec{w}\right) + g\left(\vec{v}, \vec{t}\right)$$

10.2 Bilinear Forms

Like the metric tensor, bilinear forms are $(0, 2)$-tensors with the same properties as metric tensors:

$$\mathcal{B} : V \times V \to \mathbb{R}$$
$$\mathcal{B} \mapsto v^i w^j B_{ij}$$

$$a\mathcal{B}\left(\vec{v}, \vec{w}\right) = \mathcal{B}\left(\vec{v}, a\vec{w}\right) = \mathcal{B}\left(a\vec{v}, \vec{w}\right)$$
$$\mathcal{B}\left(\vec{v} + \vec{u}, \vec{w}\right) = \mathcal{B}\left(\vec{v}, \vec{w}\right) + \mathcal{B}\left(\vec{u}, \vec{w}\right)$$

$$\mathcal{B}\left(\vec{v}, \vec{w} + \vec{t}\right) = \mathcal{B}\left(\vec{v}, \vec{w}\right) + \mathcal{B}\left(\vec{v}, \vec{w}\vec{t}\right)$$

with the same transformation rules as the metric tensor:

$$\widetilde{\mathcal{B}_{ij}} = F_i^k F_j^l \mathcal{B}_{lk}$$

$$\mathcal{B}_{lk} = B_k^i B_l^j \widetilde{\mathcal{B}_{ij}}$$

A *form* is just a function that takes vectors are inputs and output a scalar:

$$V \times V \times V \ldots \times V \to \mathbb{R}$$

Covectors are somtimes called 1-Forms since they are linear forms. If you look at one input of the bilinear form, it behaves like a covector (scaling and addition work the same way).

10.3 Properties of the Metric Tensor

The following two additional criterion are what makes a metric tensor a special type of bilinear form:

$$g\left(\vec{v}, \vec{w}\right) = v^i w^j g_{ij} = v^i w^j g_{ji} = g\left(\vec{w}, \vec{v}\right)$$

$$g\left(\vec{v}, \vec{v}\right) = \|v\|^2 \geq 0$$

Chapter 11

Linear Maps are Vector-Covector Pairs

Recall the abstract definition of the tensor from earlier:

Definition 11.0.1: Tensors as a combination of vectors and convectors combined using the tensor product

Best definition, but a bit abstract.

This implies that the $(0,2)$-tensors (e.g. bilinear forms) and $(1,1)$-tensors (e.g. linear maps) can be built from just $(0,1)$ and $(1,0)$ tensors (covectors and contravariant tensors, respectively).

If you multiply a column vector with a row vector, you get scalar. However, if you multiply a row vector (i.e. covector) with a column vector (i.e. vector), you get a matrix (i.e. a linear map).

$$\begin{bmatrix} 3 \\ -4 \end{bmatrix} \begin{bmatrix} 2 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 3 \\ -8 & -4 \end{bmatrix}$$

If we try the reverse operation by breaking an arbitrary matrix into a row vector and column vector, then we find that only some matrices can be broken down.

Pure matrices can be written as the product of column vector and row vector components while **impure** matrices cannot.

But with pure matrices, the linear transformation just scales the input vector by a constant, which isn't very interesting.

$$L^i_j = v^i \alpha_j$$

$$\begin{bmatrix} 4 & 400 \\ 8 & 800 \end{bmatrix}$$

So we need a different way of constructing a linear map since multiplying the column vector by a row vector only allows us to create uninteresting linear transformations.

Consider the following products between basis and dual basis vectors.

$$\vec{e}_1 \epsilon^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\vec{e}_1 \epsilon^2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\vec{e}_2 \epsilon^1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\vec{e}_2 \epsilon^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

If we take a linear combination all these products, we can represent any matrix. So these products $\left\{\vec{e}_1\epsilon^1, \vec{e}_2\epsilon^1, \vec{e}_2\epsilon^1, \vec{e}_2\epsilon^2\right\}$ form a basis of all matrices that are linear maps from $V \to V$

$$a\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} + b\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} + c\begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix} + d\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix} = \begin{bmatrix}a & b\\ c & d\end{bmatrix}$$

Any general linear map, L can be written

$$L = L^i_j \vec{e}_i \epsilon^j$$

Proof: Consider an input vector, $\vec{v}$, and output vector, $\vec{w}$

$$\vec{w} = L\left(\vec{v}\right)$$

$$\vec{w} = L^i_j \vec{e}_i \epsilon^j\left(\vec{v}\right) = L^i_j \vec{e}_i \epsilon^j\left(v^k \vec{e}_k\right)$$

$$\vec{w} = L^i_j \vec{e}_i v^k \epsilon^j\left(\vec{e}_k\right)$$

$$\epsilon_j\left(\vec{e}_k\right) = \delta^j_k$$

$$\implies \vec{w} = L^i_j \vec{e}_i v^k \delta^j_k$$

$$\vec{w} = L^i_j \vec{e}_i v^j$$

$$\vec{w} = L^i_j v^j \vec{e}_i$$

This is the same result as before where

$$w^i = L^i_j v^j$$

□

The basis we had before, $\left\{\vec{e}_1\epsilon^1, \vec{e}_2\epsilon^1, \vec{e}_2\epsilon^1, \vec{e}_2\epsilon^2\right\}$, is not the only basis we can have. We can have a different basis that can form the set of all linear maps when we take a linear combination.

Additionally, the following term is a *tensor product* between a vector and covector:

$$\vec{e}_i \epsilon^j$$

It can be written using the tensor product symbol, $\otimes$

$$\vec{e}_i \otimes \epsilon^j$$

Chapter 12

Bilinear Forms as Covector-Covector Pairs

Note:-

Note on non-standard notation.
The tensor product between covector-covector pairs will be written as

$$\epsilon_i \epsilon_j = \epsilon^i \otimes \epsilon^j$$

and the tensor product between vector-covector:

$$\vec{e}_i \epsilon^j = \vec{e}_i \otimes \epsilon^j$$

12.1 Benefits of Tensor Product Definition for Linear Maps

One of the benefits of looking at tensors as a product between covectors and vectors is that we can rederive the transformation rules easily for linear maps:

$$L = L^k_l \vec{e}_k \epsilon^l$$

$$L = L^k_l \left(B^i_k \widetilde{\vec{e}_i} \right) \left(F^l_j \widetilde{\epsilon^j} \right)$$

$$L = \left(B^i_k L^k_l F^l_j \right) \widetilde{\vec{e}_i} \widetilde{\epsilon^j}$$

$$\widetilde{L^i_j} = B^i_k L^k_l F^l_j$$

We also showed that we derive the output vector components that result from a linear map:

$$\vec{w} = L\left(\vec{v}\right)$$

$$\vec{w} = L^i_j \vec{e}_i \epsilon^j \left(v^k \vec{e}_k \right)$$

$$\vec{w} = L^i_j \vec{e}_i v^k \epsilon^j \left(\vec{e}_k \right)$$

$$\vec{w} = L^i_j v^k \vec{e}_i \delta^j_k$$

$$\vec{w} = L^i_j v^j \vec{e}_i$$

Finally, a tensor product allows us to get the dimensions of "array multiplication." Previously we showed that the product of a column vector and row vector forms a matrix/linear map:

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix} = \begin{bmatrix} v^1 \alpha_1 & v^1 \alpha_2 \\ v^2 \alpha_1 & v^2 \alpha_2 \end{bmatrix}$$

But doing the tensor product gives us a different, but equivalent way of construcing the linear map:

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \alpha_1 & \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \alpha_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} v^1\alpha_1 \\ v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} v^1\alpha_2 \\ v^2\alpha_2 \end{bmatrix} \end{bmatrix}$$

The result of the tensor product is a row vector of column vectors, which is basically a matrix.

12.2 Bilinear Forms as Covector-Covector Pairs

We will represent a bilinear form as a *linear combination* of covector-covector pairs:

$$\mathcal{B} = \mathcal{B}_{ij}\epsilon^i\epsilon^j = \mathcal{B}_{ij}\left(\epsilon^i \otimes \epsilon^j\right)$$

We choose covector-covector pairs since each covector takes in a vector input, so two covectors will take in two inputs just like a bilinear form.

Let's confirm the transformation rules for this new representation of a bilinear form as a linear combination of covector-covector pairs:

$$\mathcal{B} = \mathcal{B}_{kl}\epsilon^k\epsilon^l$$

$$\mathcal{B} = \mathcal{B}_{lk}\left(F_i^k\widetilde{\epsilon^i}\right)\left(F_j^l\widetilde{\epsilon^j}\right)$$

$$\mathcal{B} = \underbrace{\left(F_i^kF_j^l\mathcal{B}_{lk}\right)}_{\widetilde{\mathcal{B}_{ij}}}\widetilde{\epsilon^i}\widetilde{\epsilon^j}$$

$$\widetilde{\mathcal{B}_{ij}} = F_i^kF_j^l\mathcal{B}_{lk}$$

We can also rederive the formula for computing the bilinear form on two input vectors:

$$\mathcal{B} = \mathcal{B}_{ij}\epsilon^i\epsilon^j$$

$$s = \mathcal{B}\left(\vec{v},\vec{w}\right) = \mathcal{B}_{ij}\epsilon^i\epsilon^j\left(\vec{v},\vec{w}\right) = \mathcal{B}_{ij}\epsilon^i\epsilon^j\left(v^k\vec{e}_k, w^l\vec{e}_l\right)$$

$$s = \mathcal{B}_{ij}\epsilon^i\left(v^k\vec{e}_k\right)\epsilon^j\left(w^l\vec{e}_l\right)$$

$$s = \mathcal{B}_{ij}v^kw^l\delta_k^i\delta_l^k = \mathcal{B}_{ij}v^iw^k$$

Doing the tensor product between two covectors gives us the "shape" of the bilinear form:

$$\begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix} \otimes \begin{bmatrix} \beta_1 & \beta_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix}\beta_1 & \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix}\beta_2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} \alpha_1\beta_1 & \alpha_2\beta_1 \end{bmatrix} & \begin{bmatrix} \alpha_1\beta_2 & \alpha_2\beta_2 \end{bmatrix} \end{bmatrix}$$

The tensor product between two covectors gives a row of rows. This might contradict what we got earlier for a bilinear form, which was a matrix. But recall that we had to write the two vector inputs in an awkward manner by writing one of the vector inputs as a row vector, when usually we wrote vectors as column vectors:

$$\mathcal{B}\left(\vec{v},\vec{w}\right) = \begin{bmatrix} v^1 & v^2 \end{bmatrix}\begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} \\ \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix}\begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$$

but this alternative representation of bilinear forms as a row of rows allows us to write both vector inputs as column vectors:

$$\mathcal{B}\left(\vec{v},\vec{w}\right) = \begin{bmatrix} \begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} \end{bmatrix} & \begin{bmatrix} \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix} \end{bmatrix}\begin{bmatrix} v^1 \\ v^2 \end{bmatrix}\begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$$

$$= \left(\begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} \end{bmatrix}v^1 + \begin{bmatrix} \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix}v^2\right)\begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$$

$$
\begin{aligned}
&= \begin{bmatrix} \left(\mathcal{B}_{11}v^1 + \mathcal{B}_{21}v^2\right) & \left(\mathcal{B}_{12}v^1 + \mathcal{B}_{22}v^2\right) \end{bmatrix} \begin{bmatrix} w^1 \\ w^2 \end{bmatrix} \\
&= \left(\mathcal{B}_{11}v^1 + \mathcal{B}_{21}v^2\right) w^1 + \left(\mathcal{B}_{12}v^1 + \mathcal{B}_{22}v^2\right) w^2 \\
&= w^1\mathcal{B}_{11}v^1 + w^1\mathcal{B}_{21}v^2 + w^2\mathcal{B}_{12}v^1 + w^2\mathcal{B}_{22}v^2 \\
&= B_{ij}v^i w^j
\end{aligned}
$$

Chapter 13

Tensor Product vs. Kronecker Product

Both the tensor product and kronecker product are denoted with $\otimes$

13.1 Definition of Tensor Product

Basis for vector space, V

$$\vec{e}_1, \vec{e}_2 \in V$$

Basis for dual vector space, $V*$

$$\epsilon^1, \epsilon_2 \in V$$

$$\epsilon^i\left(\vec{e}_j\right) = \delta_{ij}$$

The tensor product takes two tensors and produces a new tensor.

$$\vec{e}_i \otimes \epsilon^j$$

In the above case the tensor product takes a vector and covector to produce a linear map:

Proof:

$$\left(\vec{e}_i \otimes \epsilon^j\right)\left(\vec{v}\right)$$

$$= \vec{e}_i \otimes \left(\epsilon^j\left(\vec{v}\right)\right)$$

$$= \vec{e}_i \otimes \left(\epsilon^j\left(v^k\vec{e}_k\right)\right)$$

$$= \vec{e}_i v^k \otimes \left(\epsilon^j\left(\vec{e}_k\right)\right)$$

$$= \vec{e}_i v^k \delta^j_k$$

$$= \vec{e}_i v^j$$

Thus the tensor product between $\vec{e}_i$ and ϵ^j is a linear map since it can take vector input and produces a vector output

□

13.2 Kronecker Product

The kronecker product operates on two arrays/matrices, in this case a column vector and row vector:

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \alpha_1 & \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \alpha_2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} v^1\alpha_1 \\ v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} v^1\alpha_2 \\ v^2\alpha_2 \end{bmatrix} \end{bmatrix}$$

The kronecker product between the above and a column vector, $\vec{w}$:

$$= \begin{bmatrix} \begin{bmatrix} v^1\alpha_1 \\ v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} v^1\alpha_2 \\ v^2\alpha_2 \end{bmatrix} \end{bmatrix} \otimes \begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} v^1\alpha_1 \\ \\ v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} v^1\alpha_2 \\ \\ v^2\alpha_2 \end{bmatrix} \end{bmatrix} w^1 \\ \begin{bmatrix} \begin{bmatrix} v^1\alpha_1 \\ \\ v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} v^1\alpha_2 \\ \\ v^2\alpha_2 \end{bmatrix} \end{bmatrix} w^2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} w^1v^1\alpha_1 \\ \\ w^1v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} w^1v^1\alpha_2 \\ \\ w^1v^2\alpha_2 \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} \begin{bmatrix} w^2v^1\alpha_1 \\ \\ w^2v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} w^2v^1\alpha_2 \\ \\ w^2v^2\alpha_2 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

13.3 Relation between Tensor Product and Kronecker Product

Tensor Products act on abstract, algebraic tensors while the kronecker product acts on arrays (which in turn might represent tensors).

Tensor product:

$$\vec{v} \otimes \alpha = \left(v^i\vec{e}_i\right) \otimes \left(\alpha_j e^j\right)$$

$$= v^i\alpha_j \left(\vec{e}_i \otimes \vec{e}_j\right)$$

Remember that the quantity, $\vec{e}_i \otimes \vec{e}_j$ represents a sort of unit linear map, so the scalar quantity, $v^i\alpha_j$ represents the entries of some matrix.

Doing the kronecker product gives:

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} v^1\alpha_1 \\ v^2\alpha_1 \end{bmatrix} & \begin{bmatrix} v^1\alpha_2 \\ v^2\alpha_2 \end{bmatrix} \end{bmatrix}$$

the kronecker product basically gives an array of arrays which is basically a matrix. We see that components, $v^i\alpha_j$ are the entries that the kronecker product gives. Thus the tensor product and kronecker product are similar, but one acts on the abstract tensor while the other operates on the arrays.

Chapter 14

Tensors as General Vector/Covector Combinations

14.1 Review of Tensors Written using Tensor Product

Linear Maps

$$L = L^i_j \vec{e}_i \epsilon^j = L^i_j \vec{e}_i \otimes \epsilon^j$$

Bilinear Forms:

$$\mathcal{B} = \mathcal{B}_{ij} \epsilon^i \epsilon^j = \mathcal{B}_{ij} \epsilon^i \otimes \epsilon^j$$

From these definitions, we can rederive the transformation rules, multiplication formulas, and array shapes.

14.2 New Tensors

Let's define some new tensors:

$$D = D^{ab} \vec{e}_a \vec{e}_b \qquad (2,0) - \text{tensor}$$
$$Q = Q^i_{jk} \vec{e}_i \epsilon^j \epsilon^k \qquad (1,2) - \text{tensor}$$

14.2.1 What are the transformation rules?

For the D tensor

$$D = D^{ab} \vec{e}_a \vec{e}_b$$
$$D = D^{ab} \left(B^i_a \widetilde{\vec{e}_i} \right) \left(B^j_b \widetilde{\vec{e}_j} \right)$$
$$D = \left(D^{ab} B^i_a B^j_b \right) \widetilde{\vec{e}_i} \widetilde{\vec{e}_j}$$
$$\widetilde{D^{ij}} = D^{ab} B^i_a B^j_b$$

Similarly,

$$D^{ab} = \widetilde{D^{ij}} F^a_i F^b_j$$

For the Q tensor

$$Q = Q^a_{bc} \vec{e}_a \epsilon^b \epsilon^c$$
$$Q = Q^a_{bc} \left(B^i_a \widetilde{\vec{e}_i} \right) \left(F^b_j \widetilde{\epsilon^j} \right) \left(F^c_k \epsilon^k \right)$$
$$Q = \left(Q^a_{bc} B^i_a F^b_j F^c_k \right) \widetilde{\vec{e}_i} \widetilde{\epsilon^j} \epsilon^k$$
$$\widetilde{Q^i_{jk}} = Q^a_{bc} B^i_a F^b_j F^c_k$$

Similarly,

$$Q^a_{bc} = \widetilde{Q^i_{jk}} F^a_i B^j_b B^k_c$$

14.2.2 What is the Multiplication Rule for $Q(D)$

Writing $Q(D)$ is ambiguous, since there are many ways of applying the covectors to the input vectors:

$$Q(D) = Q^i_{jk}\vec{e}_i\epsilon^j\epsilon^k\left(D^{ab}\vec{e}_a\vec{e}_b\right)$$

Let the output of $Q(D)$ be a vector, $\vec{w}$:

$$\vec{w} = Q(D) = w^i\vec{e}_i$$

Case $w^i = Q^i_{jk}D^{jk}$**:**

$$\begin{aligned} Q(D) &= Q^i_{jk}D^{ab}\vec{e}_i\ \epsilon^j\left(\vec{e}_a\right)\ \epsilon^k\left(\vec{e}_b\right) \\ &= Q^i_{jk}D^{ab}\vec{e}_i\delta^j_a\delta^k_b \\ &= Q^i_{jk}D^{jk}\vec{e}_i \end{aligned}$$

Case $w^i = Q^i_{jk}D^{kj}$**:**

$$\begin{aligned} Q(D) &= Q^i_{jk}D^{ab}\vec{e}_i\ \epsilon^j\left(\vec{e}_b\right)\ \epsilon^k\left(\vec{e}_a\right) \\ &= Q^i_{jk}D^{ab}\vec{e}_i\delta^j_b\delta^k_a \\ &= Q^i_{jk}D^{kj}\vec{e}_i \end{aligned}$$

Case $w^i = Q^i_{jk}D^{kb}\vec{e}_b\epsilon^j$**:**

Apply only one of the covectors to one of the input vectors:

$$Q(D) = Q^i_{jk}D^{ab}\vec{e}_i\epsilon^j\vec{e}_b\ \epsilon^k\left(\vec{e}_a\right)$$

$$Q(D) = Q^i_{jk}D^{ab}\vec{e}_i\epsilon^j\vec{e}_b\delta^k_a$$

$$Q(D) = Q^i_{jk}D^{kb}\vec{e}_i\epsilon^j\vec{e}_b$$

Case $w^i = Q^i_{jk}D^{aj}\vec{e}_a\epsilon^k$**:**

Apply only one of the covectors to one of the input vectors:

$$Q(D) = Q^i_{jk}D^{ab}\vec{e}_i\epsilon^k\vec{e}_a\ \epsilon^j\left(\vec{e}_b\right)$$

$$Q(D) = Q^i_{jk}D^{ab}\vec{e}_i\epsilon^k\vec{e}_a\delta^j_b$$

$$Q(D) = Q^i_{jk}D^{aj}\vec{e}_i\epsilon^k\vec{e}_a$$

There are still more valid ways to do $Q(D)$ as well. Note that in the case of the linear map, there was only one covector acting on one vector input, so there was only one way to write out the multiplication, but this is not the case with $Q(D)$.

14.2.3 What are the array shapes for D and Q

Shape of Q

$$\begin{bmatrix} v^1 \\ v^1 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix} \otimes \begin{bmatrix} \beta_1 & \beta_2 \end{bmatrix}$$

$$= \left[\begin{bmatrix} v^1\alpha_1 \\ v^2\alpha_1 \end{bmatrix} \quad \begin{bmatrix} v^1\alpha_2 \\ v^2\alpha_2 \end{bmatrix} \right] \otimes \begin{bmatrix} \beta_1 & \beta_2 \end{bmatrix}$$

$$= \left[\beta_1 \left[\begin{bmatrix} v^1\alpha_1 \\ v^2\alpha_1 \end{bmatrix} \quad \begin{bmatrix} v^1\alpha_2 \\ v^2\alpha_2 \end{bmatrix} \right] \quad \beta_2 \left[\begin{bmatrix} v^1\alpha_1 \\ v^2\alpha_1 \end{bmatrix} \quad \begin{bmatrix} v^1\alpha_2 \\ v^2\alpha_2 \end{bmatrix} \right] \right]$$

$$= \left[\left[\begin{bmatrix} \beta_1 v^1\alpha_1 \\ \beta_1 v^2\alpha_1 \end{bmatrix} \quad \begin{bmatrix} \beta_1 v^1\alpha_2 \\ \beta_1 v^2\alpha_2 \end{bmatrix} \right] \quad \left[\begin{bmatrix} \beta_2 v^1\alpha_1 \\ \beta_2 v^2\alpha_1 \end{bmatrix} \quad \begin{bmatrix} \beta_2 v^1\alpha_2 \\ \beta_2 v^2\alpha_2 \end{bmatrix} \right] \right]_{\vec{e}_i \epsilon^j \epsilon^k}$$

This is a row of row of column vectors. This can be visualized as a 3d cube, but this causes us to lose information about what type of tensor we have. Since we have 2 row aspects and 1 column aspect, we have a $(2, 0)$-tensor

Shape of D(Q)

As shown previously, there is not just one way to interpret $D(Q)$, so there isn't a general way to find the shape of $D(Q)$ with array multiplication. With higher types of tensors, array definition is less useful and it's better to just use einstein notation.

Chapter 15

Tensor Product Spaces

Let α and β be covectors. The tensor product has the following rules:

Scaling:

$$n\left(\vec{v}\alpha\right) = \left(n\vec{v}\alpha\right) = \vec{v}\left(n\alpha\right)$$

Adding:

$$\vec{v}\left(\alpha+\beta\right) = \vec{v}\alpha + \vec{v}\beta$$

$$\left(\vec{v}+\vec{w}\right)\alpha = \vec{v}\alpha + \vec{w}\alpha$$

Or for general tensor products between vectors, the above 3 rules can be written as:

$$n\left(\vec{a}\vec{b}\vec{c}\vec{d}\right) = \vec{a}\left(n\vec{b}\right)\vec{c}\vec{d} = \vec{a}\vec{b}\left(n\vec{c}\right)\vec{d} = \vec{a}\vec{b}\vec{c}\left(n\vec{d}\right)$$

$$\vec{a}\left(\vec{b}+\vec{c}\right) = \vec{a}\vec{b} + \vec{a}\vec{c}$$

$$\left(\vec{a}+\vec{b}\right)\vec{c} = \vec{a}\vec{c} + \vec{b}\vec{c}$$

where $\vec{v}\vec{w} = \vec{v}\otimes\vec{w}$

Since we have these scaling and addition rules for tensors, this means that tensors form a vector space.

Review of vector spaces we know:

$$\vec{v}, \vec{w}, \vec{e}_1, \vec{e}_2 \in V$$

$$\alpha, \beta, \epsilon^1, \epsilon^2 \in V^*$$

This implies that

$$\vec{v}\alpha, \vec{w}\beta, L^i_j\vec{e}_i\epsilon^j \in V\otimes V^*$$

$$\vec{a}\vec{b}\vec{c} \in V\otimes V\otimes V$$

Note that this symbol, $\otimes$, is different that the kronecker product and the tensor product between vectors. The above use of $\otimes$ is the tensor product of *Vector Spaces* and not ordinary vectors/covectors.

Table 15.1: Types of tensor elements that are an element of $V\otimes V^*$

$L^i_j\vec{e}_i\epsilon^j \in V\otimes V^*$	(1, 1)-Tensors
$L^i_j v^j = w^i$	$V\to V$
$L^i_j\alpha_i = \beta_j$	$V^*\to V^*$
$L^i_j v^j\alpha_i = s$	$V\times V^*\to\mathbb{R}$
$L^i_j\alpha_i v^j = s$	$V^*\times V\to\mathbb{R}$

Table 15.2: Types of tensor elements that are an element of $V^* \otimes V^*$. Note that the last two entries will have different vector spaces, V, for the input and different dual spaces, V^* since they sum over different indices.

$\mathcal{B}_{ij}\epsilon^i\epsilon^i \in V^* \otimes V^*$	(0, 2)-Tensors
$\mathcal{B}_{ij}v^iw^i = s$	$V \times V \to \mathbb{R}$
$\mathcal{B}_{ij}v^i = \alpha_j$	$V \to V^*$
$\mathcal{B}_{ij}v^j = \beta_i$	$V \to V^*$

Table 15.3: Types of tensor elements that are an element of $V^* \otimes V \otimes V^* \otimes V^*$. T is a Multilinear Map

$T \in V^* \otimes V \otimes V^* \otimes V^*$	$T = T_{i\ kl}^{\ j} v^i \alpha_j w^k u^l$
$T = T_{i\ kl}^{\ j} v^i \alpha_j w^k u^l$	$V \times V^* \times V \times V \to \mathbb{R}$
$T = T_{i\ kl}^{\ j} U^{ikl} \beta_j$	$(V \otimes V \otimes V) \times V^* \to \mathbb{R}$
$T = T_{i\ kl}^{\ j} \alpha_j D^{kl}$	$V^* \times (V \otimes V) \to V^*$
$T = T_{i\ kl}^{\ j} L_j^i$	$(V \otimes V^*) \to (V^* \otimes V^*)$

Definition 15.0.1: Multilinear Map

A function that is linear when all inputs except one are held constant. All tensors are multilinear maps.

$$T(x_1, x_2, \ldots, nx_i, \ldots, x_n) = nT(x_1, x_2, \ldots, x_i, \ldots, x_n)$$
$$T(x_1, x_2, \ldots, x_i + y_i, \ldots, x_n) = T(x_1, x_2, \ldots, x_i, \ldots, x_n) + T(x_1, x_2, \ldots, y_i, \ldots, x_n)$$

Chapter 16

Raising/Lowering Indexes

We want a way to raise or lower indexes:

$$T_i \leftrightarrow T^i$$

We want a correspondance between vectors in V and V^* (homomorphism). At first we might pair $\vec{e}_i$ with ϵ^i

$$\vec{v} = v^1\vec{e}_1 + v^2\vec{e^2}\dots$$
$$\alpha = v^1\epsilon^1 + v^2\epsilon_2\dots$$

However, this doesn't work when we do a change in basis. When we scale the basis vector by a constant, the covector basis scales in the opposite way, because basis vectors are covariant while basis covectors are contravariant.

Introducing a basis was what made this not work, so the correct solution will not use a basis. We can take the dot product between a vector in V and another vector to get an element of V^*

$$\vec{v}\cdot\_ \in V^*$$

This is true since the above expression takes in one vector and outputs a scalar, just like a covector. The dot product is also linear:

$$\vec{v}\cdot\left(n\vec{a}\right) = n(\vec{v}\cdot\vec{a})$$
$$\vec{v}\cdot\left(\vec{a}+\vec{b}\right) = \vec{v}\cdot\vec{a} + \vec{v}\cdot\vec{b}$$

If $\vec{v}\cdot\_ \in V^*$, then we can represent it as a linear combination of dual basis vectors:

$$\vec{v}\cdot\_ = x_i\epsilon^i$$

Remember that

$$\vec{v}\cdot\vec{w} = g(\vec{v},\vec{w}) = g_{ij}v^iw^i$$

Similarly,

$$\vec{v}\cdot\_ = g(\vec{v},\_) = g_{ik}\epsilon^i\epsilon^k\left(v^j\vec{e}_j\right)$$
$$= g_{ik}v^j\epsilon^i\delta^k_j = g_{ij}v^j\epsilon^i$$

This means

$$x_i = g_{ij}v^j$$

Note that the above equation allows us to convert between subscripts and superscripts using the metric tensor.

Summary

$$\vec{v} = v^j\vec{e}_j = \widetilde{v^j}\widetilde{\vec{e}_j} \in V$$
$$\vec{v}\cdot\_ = x_i\epsilon^i = \widetilde{x_i}\widetilde{\epsilon_i} \in V^*$$

$$x_i = g_{ij}v^j \qquad \widetilde{x_i} = \widetilde{g_{ij}}\widetilde{v^j}$$

$v^i \neq x_i$ in general. They are only equal when $g_{ij} = \delta_{ij}$, which is when we have an orthonormal coordinate system.

We usually think of the metric tensor as $g : V \times V \to \mathbb{R}$, but we have used it here as $g : V \to V^*$. Now we need to do the reverse and go from a covector to a vector.

$$g \in V^* \otimes V^*$$
$$\mathfrak{g} \in V \otimes V$$

We can use the "inverse" metric tensor, $\mathfrak{g} : V^* \to V$, to find a homomorphism between a covector and a vector.

$$\mathfrak{g}^{ki} g_{ij} = \delta^k_j$$

$$x_i = g_{ij}v^j$$
$$\mathfrak{g}^{ki} x_i = \mathfrak{g}^{ki} g_{ij} v^j$$
$$\mathfrak{g}^{ki} x_i = \delta^k_j v^j$$
$$\mathfrak{g}^{ki} x_i = v^k$$

We can use the metric tensor (sometimes called the *covariant metric tensor*) to lower indexes, while the inverse metric tensor (*contravariant metric tensor*) can be used to lower raise indexes.

Example 16.0.1 (Raising/Lowering)

$$Q \in V \otimes V^* \otimes V^*$$
$$Q = Q^i{}_{jk} \vec{e}_i \epsilon^j \epsilon^k$$
$$Q' = Q^i{}_{jk} \mathfrak{g}^{jx} \vec{e}_i \epsilon^j \epsilon^k = Q^{ix}{}_k \vec{e}_i \epsilon^x \epsilon^k$$
$$Q' \in V \otimes V \otimes V^*$$

When we convert between a vector, $\vec{v}$, to a covector, x we have

$$x = g(\vec{v}, \_)$$

$$x_i = g_{ij}v^j$$

We can also use a different notation, $\flat$:

$$x = g(\vec{v}, \_) = \flat\vec{v}$$

Conversely, converting from a covector to a vector requires:

$$\vec{v} = \mathfrak{g}(x)$$

$$v^k = \mathfrak{g}_{ki} x_i$$

$$\vec{v} = \mathfrak{g}(x) = \sharp x$$