The Einstein Summation Notation

Introduction to Cartesian Tensors

and Extensions to the Notation

Alan H. Barr
California Institute of Technology

Introduction

The Einstein summation notation is an algebraic short-hand which allows multidimensional Cartesian quantities to be expressed, manipulated, and simplified in a compact and unambiguous manner. With this notation, it is convenient to manipulate tensor expressions involving determinants, rotations, multidimensional derivatives, matrix inverses, cross products, and a variety of other multicomponent mathematical entities.

The classical form of the summation notation is incomplete, however, in that it does not express all component equations that can be expressed with conventional scalar notation and summation signs. To overcome this limitation, an explicit "no-sum" operator is introduced, which completes the notation in this regard. In addition, other special symbols (μ, κ, and ν) are introduced for conveniently manipulating quaternion rotations.

Order-independent Representation

The Einstein summation notation is an algebraic short-hand for

- expressing multicomponent Cartesian quantities,
- manipulating them,
- simplifying them, and
- expressing the expressions in a computer language short-hand

all in a compact and unambiguous manner.

The summation convention has a remarkable conciseness and ease of manipulation for multidimensional equations. Expressions involving the cross-product, determinant, matrix inverse, rotation matrices and the three dimensional axis of rotation can all be represented using a few subscripted terms. In addition, these terms explicitly delineate the numerical components of the equations so that the tensors can be manipulated by a computer. A C-preprocessor has been implemented for this purpose, which converts embedded Einstein summation notation expressions into C Language expressions.

Esn will help you write closed form expressions for such things as

1. the rotation axis of a 3D rotation matrix;
2. the representation of a matrix in another coordinate system;
3. multidimensional Taylor series and chain rule;
4. a fourth (4D) vector perpendicular to 3 others

In conventional matrix and vector notation, the multiplication order contains critical information for the calculation. In the Einstein subscript form, however, the terms in the equations can be arranged and factored in different orders without changing the algebraic result. The ability to move the factors around as desired is a particularly useful symbolic property for derivations and manipulations.

Thus, using the Einstein summation notation, seemingly different formulations are seen to represent the same equations, and complex vector expressions may be simplified to create intuitive vector representations. In addition, equations can be changed easily from one set of Cartesian basis vectors to another.

Overview

This document describes the summation convention for tensors and components in Cartesian coordinate systems, following some of the notational traditions of fluid dynamics, dyadics, and solid mechanics.[1]

The first section provides background information such as the definitions and rules for the notation, and motivates the utility of the methods. The second section defines special notational symbols in the convention, and some of the manipulation properties of these symbols. The third section derives the transformation rules for Cartesian tensors, while the fourth section focuses on 3-D rotation and its axis-angle representation. The next section provides a table of more general algebraic identities which are likely to be useful in performing multidimensional manipulations. The sixth section extends the notation for quaternions; the seventh focuses on angular velocity, while the last section demonstrates example uses of the manipulation properties. Some sample identities using the no-sum operator are also shown in the last section.

1 Mathematical Background: Tensors

A *tensor* is a multidimensional mathematical object, like a vector or a matrix, which properly transforms between different coordinate systems. It can be thought of as a "linear machine" for performing these operations. Different representations of a *Cartesian* tensor are obtained by transforming between Cartesian coordinate systems, while a generalized tensor is transformed between curvilinear or anisotropic coordinate systems, such as tangent vectors in spherical coordinates. The components of a Cartesian tensor are magnitudes projected onto orthogonal Cartesian basis vectors; three dimensional N-th order Cartesian tensors are represented in terms of 3^N components.

[1]For non-Cartesian vectors and tensors, generalized tensor conventions should be used, with both subscripts and superscripts, instead of just subscripts.

Just as the map is not the territory, and the C pointer is not the C structure it points to, the arrays of numbers are not the tensors! The numbers represent the tensors. You could think of the tensor as an abstract mathematical object, that is represented by different sets of numbers that depend on the particular coordinate system that you want to represent the tensor in.

For instance, a rotation can be represented with a matrix, and the *same* rotation can be represented with a quaternion. The **rotation itself** is the tensor, not its numerical representation in matrices or quaternions. The same is true for other tensors - the operator or thing itself is the tensor, and then we represent that with arays of numbers in a variety of ways.

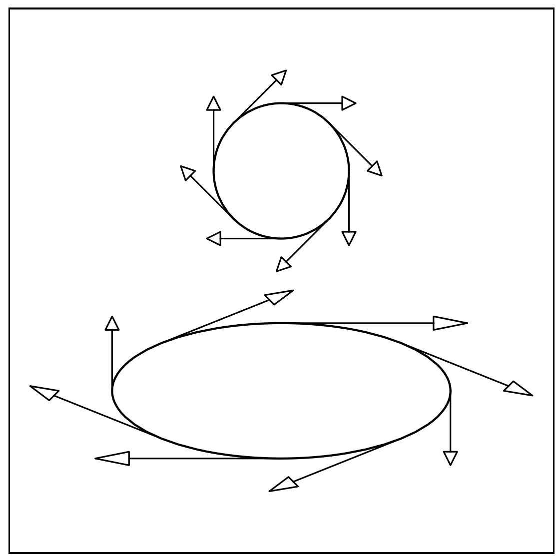

Figure 1: Tangent vectors are "contravariant" tensors. Note that the circle and its tangent vectors are transformed by the same matrix, to become a stretched curve/surface and a stretched set of tangent vectors.

Generalized tensor transformations are locally linear transformations that come in two types: *contravariant* tensors transform with the tangent vectors to a surface embedded in the transformed space, while *covariant* tensors transform with the normal vectors. Contravariant indices are generally indicated by superscripts, while covariant indices are indicated by subscripts. Good treatments of these conventions may be found in textbooks describing tensor analysis and three-dimensional mechanics (see [SEGEL], and [MIS-

NER,THORNE,WHEELER]). Generalized tensors are outside the scope of this document.

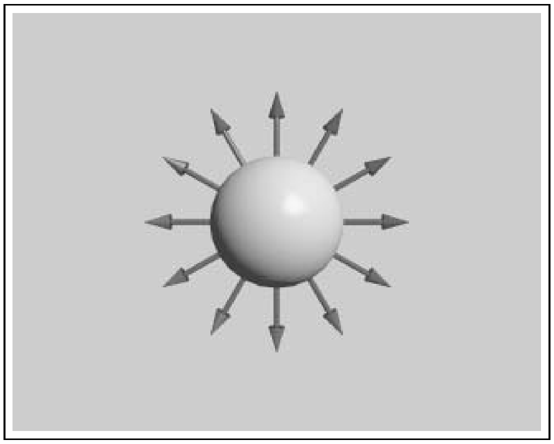

Figure 2: Normal vectors on the sphere

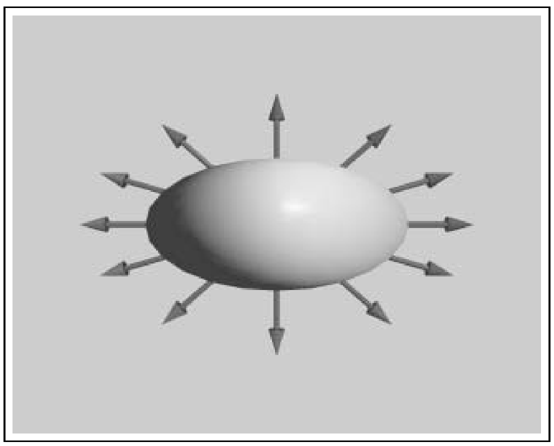

Figure 3: The original normal vectors, when transformed by the sphere-to-ellipsoid transformation matrix are not the normal vectors for the new surface!

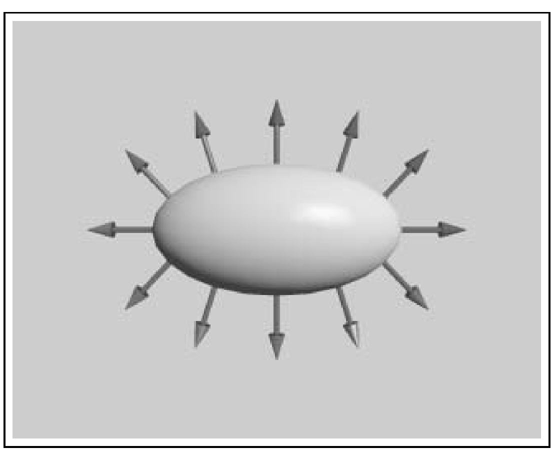

Figure 4: Normal vectors are covariant tensors; they are transformed by the inverse transpose of the sphere-to-ellipsoid transformation matrix.

For Cartesian transformations, such as pure rotation, tangent vectors transform via the same matrix as the normal vectors; there is no difference between the covariant and contravariant representations. Thus, subscript indices suffice for Cartesian tensors.

A zero-th order tensor is a scalar, whose single component is the same in all coordinate systems, while first order tensors are vectors, and second order tensors are represented with matrices. Higher order tensors arise frequently; intuitively, if a mathematical object is a "k-th order Cartesian tensor," in an N dimensional Cartesian coordinate system, then

1. The object is an entity which "lives" in the N dimensional Cartesian coordinate system.
2. The object can be represented with k subscripts and N^k components total.
3. The numerical representation of the object is typically different in different coordinate systems.
4. The representations of the object obey the Cartesian form of the transformation rule, to obtain the numerical representations of the same object in another coordinate system. That rule is described in section 3.

1.1 Example 1:

For example, let us consider the component-by-component description of the three-dimensional matrix equation

$$\underline{a} = \underline{b} + \underline{\underline{M}}\ \underline{c}$$

(we utilize one underscore to indicate a vector, two for a matrix, etc. We will occasionally use boldface letters to indicate vectors or matrices.)

The above set of equations is actually a set of three one-component equations, valid in any coordinate system in which we choose to represent the components of the vectors and matrices.

Using vertical column vectors, the equation is:

$$a_1$$

$$a_i = M_{ij}\, b_j + c_i$$

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

which is equivalent to:

$$\begin{aligned} a_1 &= b_1 + m_{11} c_1 + m_{12} c_2 + m_{13} c_3 \\ a_2 &= b_2 + m_{21} c_1 + m_{22} c_2 + m_{23} c_3 \\ a_3 &= b_3 + m_{31} c_1 + m_{32} c_2 + m_{33} c_3 \end{aligned}$$

The above equations condense into three instances of one equation,

$$a_i = b_i + \sum_{\boxed{j}=1}^{3} m_{i\boxed{j}}\, c_{\boxed{j}}\,, \quad i = 1, 2, 3.$$

The essence of the Einstein summation notation is to create a set of notational defaults so that the summation sign and the range of the subscripts do not need to be written explicitly in each expression. This notational compactness is intended to ease the expression and manipulation of complex equations.

With the following rules, the above example becomes:

$$a_i = b_i + m_{i\boxed{j}}\, c_{\boxed{j}}$$

1.2 Definition of Terms:

There are two types of subscript indices in the above equation. The subscript "i" is free to vary from one to three on both sides of the above equation, so it is called a **free index**. The free indices must match on both sides of an equation.

The other type of subscript index in Example 1 is the dummy subscript "j." This subscript is tied to the term inside the summation, and is called a **bound index**. Sometimes we will place "boxes" around the bound indices, to more readily indicate that they are bound. Dummy variables can be renamed as is convenient, as long as all instances of the dummy variable name are replaced by the new name, and the new names of the subscripts do not conflict with the names of other subscripts, or with variable names reserved for other purposes.

1.2.1 Independent Indices

A new type of subscript index is an **independent** index, denoted with brackets, which is neither free nor bound. It does not implicitly range from 1 to N (as the free and bound variables do) but instead has only the indicated value.

So, in the expression $a_{[n]}$, n is an independent index, and refers to the n-th component of array $\underline{a}$, using the current single value of n. The independent indices are a new type of summation-notation index provided by the author, useful for those algebraic expressions that do not loop or sum from 1 to N in a free or bound variable.

1.3 The Classical Rules for the N dimensional Cartesian Einstein Summation Notation

We remind the reader that an algebraic **term** is an algebraic entity separated from other terms by additions or subtractions, and is composed of a collection of multiplicative **factors**. Each factor may itself composed of a sum of terms. A **subexpression** of a given algebraic expression is a subtree combination of terms and/or factors of the algebraic expression,

The classical Einstein summation convention (without author-provided extensions) is governed by the following rules:

Scoping Rule: Given a valid summation-notation algebraic expression, the notation is still valid in each of the algebraic subexpressions. Thus, it is valid to **re-associate subexpressions**: If E, F, and G are valid ESN-expressions, then you can evaluate EF and then combine it with G or evaluate FG and then combine it with E:

$$\boxed{\text{EFG} = \text{(EF)G} = \text{E(FG)}}$$

Rule 1: A subscript index variable is **free** to vary from one to N if it occurs exactly **once** in a term.

Rule 2: A subscript index variable is **bound** to a term as the dummy index of a summation from one to N if it occurs exactly **twice** in the term. We will sometimes put boxes around bound variables for clarity.

Rule 3: A term is syntactically **wrong** if it contains **three** or more instances of the same subscript variable.

Rule 4: **Commas** in the subscript indicate that a **partial derivative** is to be taken. A (free or bound) subscript index after the comma indicates partial derivatives with respect to the default arguments of the function, (frequently spatial variables x_1, x_2, and x_3 or whatever x y z spatial coordinate system is being used). Partial derivatives with respect to a reserved variable (say t, the time parameter) are indicated by putting the reserved variable after the comma.

Comments: To be added together, expressions *must* be free in the same indices. Also, to count indices, imagine that parenthetical expressions are expanded: each free index occurs exactly once in every term, and every bound index occurs exactly twice in some of the terms. This approach

a) eliminates expressions with triple indices:

$$x_i\,(y_i\,z_j + x_j\,z_i)\;w_i$$

b) eliminates expressions with mismatching indices: $x_i\,(y_i\,z_j + x_j\,z_k)$

c) handles multiple-indexed terms correctly, such as M_{ii}

d) requires indexed constants for some operations: $x_i + 4$ is a bit sloppy, even if it's compact; $x_i + 4_i$ is more proper (although cumbersome).

With these rules, Example 1 becomes

$$a_i = b_i + m_{i\boxed{j}}\,c_{\boxed{j}}$$

The subscript "i" is a free index in each of the terms in the above equation, because it occurs only once in each term, while "j" is a bound index, because it appears twice in the last term. The boxes on the bound indices are not necessary; they are used just for emphasis.

At first, it is helpful to write out the interpretations of some "esn" expressions in full, complete with the summation signs, bound indices, and ranges on the free indices. This procedure can help clarify questions that arise, concerning the legality of a particular manipulation. In this way, you are brought back into familiar territory to see what the notation is "really" doing.

1.4 Why only two instances of a subscript index in a term?

There is an important reason why the number of instances of a subscript index in each algebraic term is restricted to two or fewer. The scoping rule shows why. We need to be able to interpret the meaning of each term and sub-term unambiguously, independently of the multiplicative association of the factors. For instance, if three instances of the subscript "j," were allowed in a term, the following expressions, being different only in the association of the factors, would be equivalent:

$$b_j\,c_j\,d_j \stackrel{?}{=} (b_{\boxed{j}}\,c_{\boxed{j}})d_j \stackrel{?}{=} b_j\,(c_{\boxed{j}}\,d_{\boxed{j}}).$$

However,

$$\sum_{j=1}^{3} b_j\,c_j\,d_j \neq (\sum_{\boxed{j}=1}^{3} b_{\boxed{j}}\,c_{\boxed{j}})\,d_j\,,\ etc.$$

The first expression is a number, while the second expression is a vector which equals a scalar ($\underline{b}\cdot\underline{c}$) times vector $\underline{d}$. The third expression is a different vector, which is the product of the vector $\underline{b}$ with the scalar $\underline{c}\cdot\underline{d}$, which are not remotely equivalent. Thus, we are limited to two instances of any particular subscript index in a term in order to retain the associative property of multiplication in our algebraic expressions. Unlike conventional matrix notation, Esn factors are both commutative and associative within a term. For beginners it is particularly important to check Esn expressions for validity with respect to rule 3 and the number of subscript instances in a term.

2 A Few Esn Symbols

The **Kronecker delta**, or identity matrix, is represented with the symbol δ with subscripts for the rows and columns of the matrix.

$$\delta_{ij} = \begin{cases} 0, i \neq j \\ 1, i = j \end{cases}$$

The **logical "1" symbol** is an extended symbol related to the Kronecker delta.

$$1_{logical\ expression} = \begin{cases} 1, \textit{expression} \text{ is true} \\ 0, \text{otherwise} \end{cases}$$

So $1_{i=j} = \delta_{ij}$.

The order-three **permutation symbol** ϵ_{ijk} is used to manipulate cross products and three dimensional matrix determinants, and is defined as follows:

$$\epsilon_{ijk} = \begin{cases} 0, & \text{if any pair of subscripts is equal} \\ 1, & (i,j,k) \text{ is an even perm. of } (1,2,3) \\ -1, & (i,j,k) \text{ is an odd perm. of } (1,2,3) \end{cases}$$

An even permutation of a list is a permutation created using an even number of interchange operations of the elements. An odd permutation requires an odd number of interchanges. The six permutations of (1,2,3) may be obtained with six interchange operations of elements:

$$(1,2,3) \rightarrow (2,1,3) \rightarrow (2,3,1)$$
$$\rightarrow (3,2,1) \rightarrow (3,1,2) \rightarrow (1,3,2).$$

Thus, the even permutations are (1,2,3), (2,3,1), and (3,1,2), while the odd permutations are (2,1,3), (3,2,1), and (1,3,2).

Higher dimensional forms of the ϵ operator have additional subscripts, but otherwise are similarly defined. The order N permutation symbol is $\epsilon_{i_1, i_2, \ldots i_N}$; the N subscripts are compared to even and odd permutations of the integers $(1, 2, \ldots, N)$ instead of the integers $(1, 2, 3)$.

Note that

$$\epsilon_{ijk\ell m} = 1_{\{i,j,k,\ell,m\} \text{ is even}} - 1_{\{i,j,k,\ell,m\} \text{ is odd}}.$$

2.1 Preliminary simplification rules

The **free-index operator** of a vector, matrix or tensor converts a conventional vector expression into the Einstein summation form. It is indicated by parentheses with subscripts on the right. Vectors require one subscript, as in

$$(\underline{b})_i = b_i$$

which is verbosely read as "the i-th component of vector b (in the default Cartesian coordinate system) is the number b sub i."

Two subscripts are needed for matrices:

$$(\underline{\underline{M}})_{ij} = M_{ij}.$$

can be read as "the ij-th component of matrix M in the default coordinate system is the number M sub i j."

To **add two vectors**, you add the components:

$$(\underline{a} + \underline{b})_i = a_i + b_i$$

This can be read as "the i-th component of the sum of vector a and vector b is the number a sub i plus b sub i."[2]

To perform **matrix-vector multiplications**, the second subscript of the matrix must be bound to the index of the vector. Thus, to multiply matrix $\underline{\underline{M}}$ by vector $\underline{b}$, new bound subscripts are created, as in:

$$(\underline{\underline{M}}\,\underline{b})_i = M_{i\boxed{j}}\, b_{\boxed{j}}$$

which converts to

$$\begin{aligned} (\underline{\underline{M}}\,\underline{b})_1 &= M_{11} b_1 + M_{12} b_2 + M_{13} b_3 \\ (\underline{\underline{M}}\,\underline{b})_2 &= M_{21} b_1 + M_{22} b_2 + M_{23} b_3 \\ (\underline{\underline{M}}\,\underline{b})_3 &= M_{31} b_1 + M_{32} b_2 + M_{33} b_3 \end{aligned}$$

To perform **matrix-matrix multiplications**, the second subscript of the first matrix must be bound to the first subscript of the second matrix. Thus, to multiply matrix $\underline{\underline{A}}$ by matrix $\underline{\underline{B}}$, you create indices like:

$$(\underline{\underline{A}}\,\underline{\underline{B}})_{ij} = A_{i\boxed{k}}\, B_{\boxed{k}j}$$

which converts to:

[2] In the esn C preprocessor, the default ranges are different:

```
#{ c_i = a_i + b_i
```

} (assuming the arrays have been declared over the same range). In C, this expands to:

```
c[0] = a[0] + b[0];
c[1] = a[1] + b[1];
c[2] = a[2] + b[2];
```

$$\begin{aligned}
(\underline{\underline{A}}\,\underline{\underline{B}})_{11} &= A_{11}\,B_{11} + A_{12}\,B_{21} + A_{13}\,B_{31}\\
(\underline{\underline{A}}\,\underline{\underline{B}})_{21} &= A_{21}\,B_{11} + A_{22}\,B_{21} + A_{23}\,B_{31}\\
(\underline{\underline{A}}\,\underline{\underline{B}})_{31} &= A_{31}\,B_{11} + A_{32}\,B_{21} + A_{33}\,B_{31}
\end{aligned}$$

$$\begin{aligned}
(\underline{\underline{A}}\,\underline{\underline{B}})_{12} &= A_{11}\,B_{12} + A_{12}\,B_{22} + A_{13}\,B_{32}\\
(\underline{\underline{A}}\,\underline{\underline{B}})_{22} &= A_{21}\,B_{12} + A_{22}\,B_{22} + A_{23}\,B_{32}\\
(\underline{\underline{A}}\,\underline{\underline{B}})_{32} &= A_{31}\,B_{12} + A_{32}\,B_{22} + A_{33}\,B_{32}
\end{aligned}$$

$$\begin{aligned}
(\underline{\underline{A}}\,\underline{\underline{B}})_{13} &= A_{11}\,B_{13} + A_{12}\,B_{23} + A_{13}\,B_{33}\\
(\underline{\underline{A}}\,\underline{\underline{B}})_{23} &= A_{21}\,B_{13} + A_{22}\,B_{23} + A_{23}\,B_{33}\\
(\underline{\underline{A}}\,\underline{\underline{B}})_{33} &= A_{31}\,B_{13} + A_{32}\,B_{23} + A_{33}\,B_{33}
\end{aligned}$$

2.1.1 Rules involving ϵ_{ijk} and δ_{ij}

The delta rule

When a Kronecker delta subscript is bound in a term, the expression simplifies by

(1) eliminating the Kronecker delta symbol, and

(2) replacing the bound subscript in the rest of the term by the other subscript of the Kronecker delta. For instance, $\delta_{i\boxed{j}}\,M_{\boxed{j}k}$ becomes M_{ik}, and $\delta_{ij}\,a_i\,M_{jk}$ becomes $a_{\boxed{j}}\,M_{\boxed{j}k}$ or $a_i\,M_{ik}$. Note that in standard notation, this is equal to $(\underline{\underline{M}}^T\;\underline{a})_k$.

Rules for the order-N permutation symbol

Interchanges of subscripts flip the sign of the permutation symbol:

$$\epsilon_{i_1\,i_2\,\ldots\,i_N} = -\epsilon_{i_2\,i_1\,\ldots\,i_N}$$

Repeated indices eliminate the permutation symbol (since lists with repeated elements are not permutations).

$$\epsilon_{iijk\ldots\ell} = 0$$

This is related to the behavior of determinants, where interchanges of columns change the sign, or repeated columns send the determinant to zero. Note repeat of index i.

The order-three permutation symbol subscript rule

For the special case of an order-three permutation symbol, the following identity holds.

$$\epsilon_{ijk} = \epsilon_{jki}$$

The epsilon-delta rule

The order-3 ϵ-δ rule allows the following subscript simplification (when the first subscript of two permutation symbols match:

$$\epsilon_{\boxed{i}jk}\,\epsilon_{\boxed{i}pq} = \delta_{jp}\,\delta_{kq} - \delta_{jq}\,\delta_{kp}$$

In other words, the combination of two permutation symbols with the same first subscript is equal to an expression involving Kronecker deltas and the four free indices j, p, k, and q. In practice, the subscripts in the permutation symbols are permuted via the above relations, in order to apply the ϵ-δ rule.

A deriviation to verify the identity is provided in Appendix A.

3 Transformation Rules for Cartesian Tensors

We express vectors, matrices, and other tensors in different Cartesian coordinate systems, without changing which tensor we are representing. The numerical representation (of the same tensors) will generally be different, in the different coordinate systems.

We now derive the transformation rules for Cartesian tensors in the Einstein summation notation, to change the representation from one Cartesian coordinate system to another. Some people in fact use these transformation rules as the definition of a Cartesian tensor – any mathemtaical object whose representation transforms like the Cartesian tensors do **is** a Cartesian tensor.

Consider a three dimensional space, with **right-handed**[3] orthonormal[4] basis vectors $\underline{e}_1$, $\underline{e}_2$, and $\underline{e}_3$. In other words, each basis vector has

[3]The author **strongly** recommends avoiding the computer graphics convention of left handed coordinates, and recommends performing all physically based and geometric calculations in right handed coordinates. That way, you can use the last 300 years of mathematics texts as references.

[4]mutually perpendicular unit vectors

unit length, is perpendicular to the other basis vectors, and the 3D vectors are named subject to the right hand rule.

Right handed unit basis vectors satisify:

$$\underline{e}_1 \times \underline{e}_2 = \underline{e}_3 ,$$
$$\underline{e}_2 \times \underline{e}_3 = \underline{e}_1 , \text{ and}$$
$$\underline{e}_3 \times \underline{e}_1 = \underline{e}_2 .$$

In addition, we note that

$$\underline{e}_i \cdot \underline{e}_j = \delta_{ij} .$$

This result occurs because the dot products of perpendicular vectors is zero, and the dot products of identical unit vectors is 1.

We also consider another set of right-handed orthonormal basis vectors for the same three dimensional space, vectors $\hat{\underline{e}}_1$, $\hat{\underline{e}}_2$, and $\hat{\underline{e}}_3$, which will also have the same properties.

$$\hat{\underline{e}}_i \cdot \hat{\underline{e}}_j = \delta_{ij} , \text{ etc.}$$

3.1 Transformations of vectors

Consider a vector $\underline{a}$, and express it with respect to both bases, with the (vertical column) array of numbers a_1, a_2, a_3 in one coordinate system, and with the array of different numbers $\hat{a}_1, \hat{a}_2, \hat{a}_3$ in the other.

$$\text{vector } \underline{a} = a_1\ \underline{e}_1 + a_2\ \underline{e}_2 + a_3\ \underline{e}_3 = a_i\ \underline{e}_i$$

and also for the same vector,

$$\underline{a} = \hat{a}_1\ \hat{\underline{e}}_1 + \hat{a}_2\ \hat{\underline{e}}_2 + \hat{a}_3\ \hat{\underline{e}}_3 = \hat{a}_i\ \hat{\underline{e}}_i$$

Since the two expressions represent the same vector, they are equal:

$$a_i\ \underline{e}_i = \hat{a}_i\ \hat{\underline{e}}_i$$

We derive the relation between $\hat{a}_i$ and a_i, by dotting both sides of the above equation with basis vector $\underline{e}_k$.

Thus, by commuting and reassociating, and the fact that the unitbasis vectors are mutually perpendicular, we obtain:

$$(a_i\ \underline{e}_i) \cdot \underline{e}_k = (\hat{a}_i\ \hat{\underline{e}}_i) \cdot \underline{e}_k, \text{ or}$$
$$a_i\,(\underline{e}_i \cdot \underline{e}_k) = \hat{a}_i\,(\hat{\underline{e}}_i \cdot \underline{e}_k), \text{ or}$$
$$a_i\,\delta_{ik} = \hat{a}_i\,(\hat{\underline{e}}_i \cdot \underline{e}_k), \text{ or}$$
$$a_k = (\underline{e}_k \cdot \hat{\underline{e}}_i)\hat{a}_i$$

(Feel free to convert the above expression into conventional notation if the validity of the rearrangement of terms is unclear).

Thus, a matrix $\underline{\underline{T}}$ re-expresses the numerical representation of a vector relative to a new basis via

$$\boxed{a_i = T_{ij}\ \hat{a}_j}$$

where

$$T_{ij} = (\underline{e}_i \cdot \hat{\underline{e}}_j).$$

Note that the transpose of T is its inverse, and det $T = 1$, so T is a rotation matrix.

3.2 Transformations of matrices

The nine basis vectors of (3 dimensional) matrices (2nd order tensors) are given by the nine outer products of the previous basis vectors:

$$\xi_{ij} = \underline{e}_i\,\underline{e}_j.$$

Thus, the nine quantities which form the nine dimensional basis "vectors" of a three-by-three matrix are the outer products of the original basis vectors:

$$\underline{\underline{M}} = M_{ij}\ \underline{e}_i\,\underline{e}_j$$

For instance, with the standard bases, where

$$\underline{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

$$\underline{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

etc., note that

$$\underline{e}_1\,\underline{e}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\underline{e}_1\,\underline{e}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

etc.

By dotting twice by $\underline{e}_p$ and $\underline{e}_q$, a similar derivation shows us that the transformation rule for a matrix is:

$$\boxed{M_{ij} = T_{ip}\ T_{jq}\ \hat{M}_{pq}}$$

3.3 Transformations of N-th order Tensors

You would not be surprised then, to imagine basis vectors of 3rd order tensors as being given by

$$\underline{\underline{\underline{\xi}}}_{ijk} = \underline{e}_i\, \underline{e}_j\, \underline{e}_k$$

with a transformation rule of

$$\boxed{A_{ijk} = (T_{ip})\,(T_{jq})\,(T_{kr})\,\hat{A}_{pqr}}$$

Also, not surprisingly, given an N^{th} order tensor quantity $\mathbf{X}$, it will have analogous basis vectors and will transform with N copies of the T matrix, via:

$$\boxed{X_{i_1 i_2 \ldots i_N} = (T_{i_1 j_1}) \ldots (T_{i_N j_N})\, \hat{X}_{j_1 \ldots j_N}}$$

4 Axis-Angle Representations of 3D Rotation

One very useful application of Esn is in the representation and manipulation of rotations. In three dimensions, rotation can take place only around one vector axis.

4.1 The Projection Operator

The first operator needed to derive the axis-angle matrix formulation is the projection operator. To project out vector $\underline{b}$ from vector $\underline{a}$, let

$$\underline{a} \setminus \underline{b} = \underline{a} - \alpha\, \underline{b}$$

such that the result is perpendicular to $\underline{b}$.

The projection operation $\underline{a} \setminus \underline{b}$" can be read as vector a "without" vector b. Note that

$$\alpha = \frac{\underline{a} \cdot \underline{b}}{\underline{b} \cdot \underline{b}}.$$

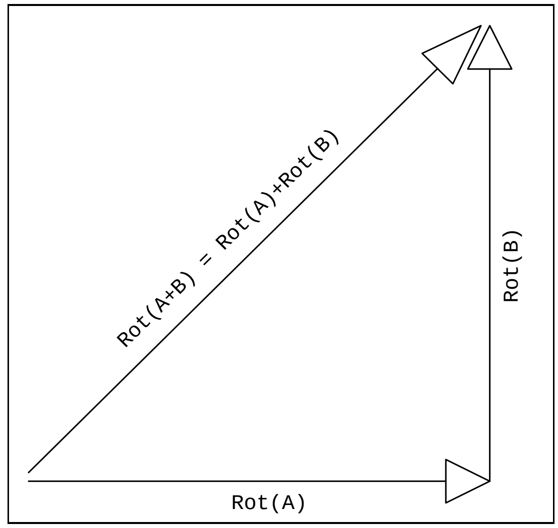

Figure 5. Visual demonstration that rotation is a linear operator. Just rotate the document and observe that the relationship holds: Rot(A+B) = Rot(A) + Rot(B) no matter which rotation operates on the vectors.

4.2 Rotation is a Linear Operator

We will be exploiting linearity properties for the derivation:

$$Rot(\underline{a} + \underline{b}) = Rot(\underline{a}) + Rot(\underline{b})$$

and

$$Rot(\sigma\; \underline{a}) = \sigma\; Rot(\underline{a}).$$

See Figure 5.

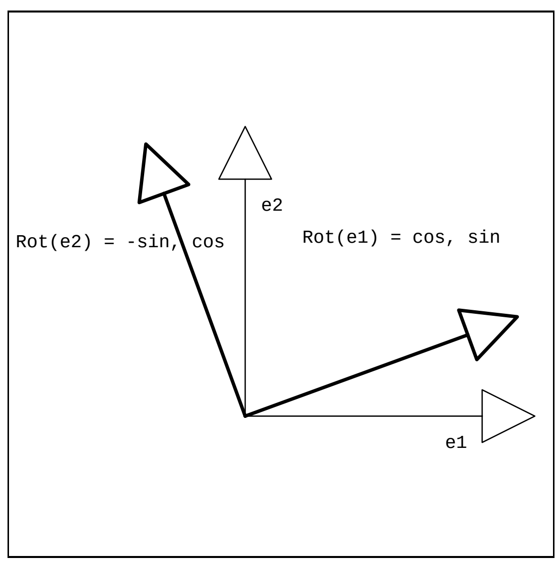

Figure 6. Two-d rotation of basis vectors e_1 and e_2 by angle θ are displayed. $rot(\underline{e}_1) = \cos\theta \underline{e}_1 + \sin\theta \underline{e}_2$, while $rot(\underline{e}_2) = -\sin\theta \underline{e}_1 + \cos\theta \underline{e}_2$.

4.3 Two-D Rotation.

Two dimensional rotation is easily derived, using linearity. We express the vector in terms of its basis vector components, and then apply the rotation operator.

Since

$$\underline{a} = a_1 \underline{e}_1 + a_2 \underline{e}_2$$

Therefore

$$rot(\underline{a}) = rot(a_1 \underline{e}_1 + a_2 \underline{e}_2).$$

Using linearity,

$$rot(\underline{a}) = a_1\, rot(\underline{e}_1) + a_2\, rot(\underline{e}_2)$$

From Figure 6 we can see that that

$$rot(\underline{e}_1) = \cos\theta \underline{e}_1 + \sin\theta \underline{e}_2$$

and

$$rot(\underline{e}_2) = -\sin\theta \underline{e}_1 + \cos\theta \underline{e}_2 .$$

Thus,

$$\begin{aligned} rot(\underline{a}) \;=\; & (\cos\theta\ a_1 - \sin\theta\ a_2)\ \underline{e}_1 \\ + & (\sin\theta\ a_1 + \cos\theta\ a_2)\ \underline{e}_2 \end{aligned}$$

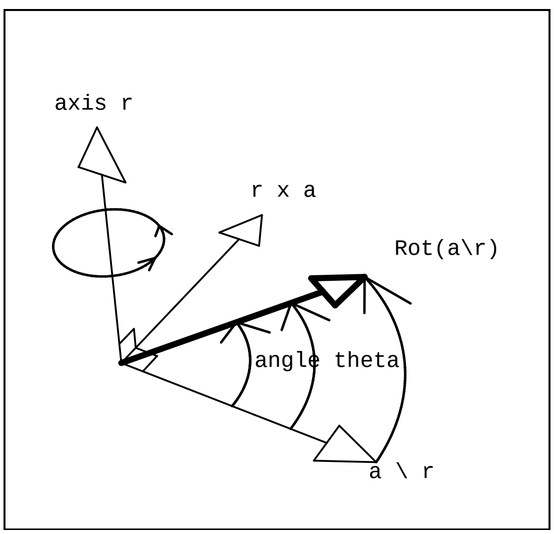

Figure 7. The rotation is around unit vector axis $\underline{r}$, by right handed angle θ. Note that the vectors $\underline{r}$, $\underline{a}\setminus\underline{r}$ and $\underline{r}\times\underline{a}$ form an orthogonal triple.

4.4 An Orthogonal triple

Given a unit vector $\underline{r}$ around which the rotation will take place (the axis) we can make three basis orthogonal basis vectors.

Let the third basis vector, $\underline{e}_3 = r$.

Let the first basis vector, $\underline{e}_1$, be the unit vector in the direction of $a \setminus r$.

Note that by the definition of projection, $\underline{e}_3$ is perpendicular to $\underline{e}_1$.

Finally, let $\underline{e}_2 = \underline{e}_3 \times \underline{e}_1$. It's in the direction of $r\times(a\setminus r)$ which is in the same direction as $r\times a$.

Note that

$$\underline{e}_1 \times \underline{e}_2 = \underline{e}_3,$$

$$\underline{e}_2 \times \underline{e}_3 = \underline{e}_1, \text{ and}$$

$$\underline{e}_3 \times \underline{e}_1 = \underline{e}_2 .$$

We have a right-handed system of basis vectors.

4.5 Deriving the axis-angle formulation

First, note that rotation of vectors parallel to $\underline{r}$ around itself remain unchanged.

We're now ready to derive $rot(\underline{a})$ around r by θ.

By definition,

$$\underline{a} = \underline{a} \setminus \underline{r} + \alpha\underline{r}$$

Thus,

$$\begin{aligned} rot(\underline{a}) &= rot(\underline{a} \setminus \underline{r} + \alpha\ \underline{r}) \\ &= rot(\underline{a} \setminus \underline{r}) + rot(\alpha\underline{r}) \\ &= rot(|\underline{a} \setminus \underline{r}|\ \underline{e}_1) + \alpha\ \underline{r} \\ &= |\underline{a} \setminus \underline{r}|\ rot(\underline{e}_1) + \alpha\ \underline{r} \end{aligned}$$

$$= |\underline{a} \setminus \underline{r}|\ (\cos\theta\ \underline{e}_1 + \sin\theta\ \underline{e}_2) + \alpha\underline{r}$$

$$= \cos\theta\ (\underline{a} \setminus \underline{r}) + \sin\theta\ (\underline{r} \times \underline{a}) + (\underline{a} \cdot \underline{r})\underline{r}$$

5 Summary of Manipulation Identities

In this section, a series of algebraic identities are listed with potential applications in multidimensional mathematical modeling.

List of Einstein Summation Identities:

(1) The **free-index operator** of a vector, matrix or tensor converts a conventional vector expression into the Einstein form. It is indicated by parentheses with subscripts, requiring one subscript for vectors, as in $(\underline{b})_i = b_i$, and two subscripts for matrices, as in $(\underline{\underline{M}})_{ij} = M_{ij}$. You can think of $()_i$ as an operator which dots the argument by the i-th basis vector.

Sometimes we will use the free index operator to select column vectors of a matrix, such as the following:

$$(\underline{\underline{E}})_i = \underline{e}_i$$

In this case $\underline{e}_1$ is the first column vector of the matrix $\underline{\underline{E}}$, $\underline{e}_2$ is the second column, etc. [5]

(2) The **dot product** of two vectors $\underline{a}$ and $\underline{b}$ is expressed via: $\underline{a} \cdot \underline{b} = a_i b_i = a_{\boxed{i}} b_{\boxed{i}}$.

(2a) The **outer product** of two vectors $\underline{a}$ and $\underline{b}$ is expressed via: $(\underline{a}\,\underline{b})_{ij} = a_i b_j$.

(3) The **vector cross product** of two vectors $\underline{a}$ and $\underline{b}$ is expressed via:

$$(\underline{a} \times \underline{b})_i = \epsilon_{ijk} a_j b_k ,$$

or, putting boxes on the matching sets,

$$(\underline{a} \times \underline{b})_i = \epsilon_{i\boxed{j}\boxed{k}} a_{\boxed{j}} b_{\boxed{k}} ,$$

or

$$(\underline{a} \times \underline{b})_1 = a_2 b_3 - a_3 b_2$$
$$(\underline{a} \times \underline{b})_2 = a_3 b_1 - a_1 b_3$$
$$(\underline{a} \times \underline{b})_3 = a_1 b_2 - a_2 b_1$$

(in which the free index of the output cross product vector becomes the first subscript of the permutation symbol, while two new bound indices are created). If vector $\underline{a}$ points East on a sphere, and vector $\underline{b}$ points North, the cross-product vector in right-handed coordinates points Up, i.e., out of the sphere, while in left-handed coordinates, the cross-product points Down into the sphere center.

(4) $\delta_{ij} \delta_{jk} = \delta_{i\boxed{j}} \delta_{\boxed{j}k} = \delta_{ik} = \delta_{ki}$.

[5] Note that $\underline{e}_i \neq (\underline{e})_i$. The right hand side is an i-th scalar, while the left side is an i-th vector.

(5) $\delta_{ii} = \delta_{\boxed{i}\boxed{i}} = 3$ in three dimensional space. $\delta_{ii} = N$ in N dimensional spaces.

(6) $\delta_{i\boxed{j}} \delta_{i\boxed{j}} = \delta_{ii} = N$.

(7) $\epsilon_{ijk} = -\epsilon_{jik}$.

(8) $\epsilon_{ijk} = \epsilon_{jki}$.

(9) The ϵ-δ rule allows the following subscript simplification:

$$\epsilon_{\boxed{i}jk} \epsilon_{\boxed{i}pq} = \delta_{jp} \delta_{kq} - \delta_{jq} \delta_{kp}$$

(10) If S_{ij} is **Symmetric**, i.e., if $S_{ij} = S_{ji}$, then

$$\epsilon_{qij} S_{ij} = 0.$$

(11) If A_{ij} is **Antisymmetric**, i.e., if $A_{ij} = -A_{ji}$, then $A_{ij} - A_{ji} = 2A_{ij}$. In addition, since $M_{ij} - M_{ji}$ is always antisymmetric,

$$\epsilon_{qij} (M_{ij} - M_{ji}) = 2\epsilon_{qij} M_{ij}$$

(12) **Partial derivatives** are taken with respect to the default argument variables when the subscript index follows a comma: For example,

$$(\nabla F)_i = F_{,i} = \frac{\partial F}{\partial x_i}.$$

Argument evaluation takes place *after* the partial derivative evaluation:

$$F_{i,j}(\underline{x}) = \left.\frac{\partial F_i(\underline{\alpha})}{\partial \alpha_j}\right|_{\underline{\alpha} = \underline{x}}$$

(13) $x_{i,j} = \delta_{ij}$, where $\underline{x}$ is the default spatial coordinate vector.

(14) Partial derivatives may also be taken with respect to reserved symbols, set aside in advance.

$$F_{,t} = \frac{\partial F}{\partial t}.$$

(15) $(\nabla^2 \underline{F})_i = F_{i,jj}$

(16) The **determinant** of an NxN matrix $\underline{\underline{M}}$ may be expressed via:

$$detM = \epsilon_{i_1 i_2 \ldots i_N} M_{1i_1} M_{2i_2} \ldots M_{Ni_N}$$

The order N cross product, is produced by leaving out one of the column vectors in the above expression, to produce a vector perpendicular to $N-1$ other vectors.

For three-by-three matrices,

$$\det(\underline{\underline{M}}) = \epsilon_{ijk} M_{1i} M_{2j} M_{3k} = \epsilon_{ijk} M_{i1} M_{j2} M_{k3} .$$

(17) Another identity involving the determinant:

$$\epsilon_{qnp} \det(\underline{\underline{M}}) = \epsilon_{ijk} M_{qi} M_{nj} M_{pk}$$

(18) The first column of a matrix $\underline{\underline{M}}$ is designated M_{i1}, while the second and third columns are indicated by M_{i2} and M_{i3}. The three rows of a three dimensional matrix are indicated by M_{1i}, M_{2i}, and M_{3i}.

(19) The **transpose operator** is achieved by switching subscripts:

$$(\underline{\underline{M}}^T)_{ij} = M_{ji} .$$

(20) A matrix times its inverse is the identity matrix:

$$M_{i\boxed{j}} (\underline{\underline{M}}^{-1})_{\boxed{j}k} = \delta_{ik} .$$

(21) The SinAxis operator. The instantaneous **rotation axis** $\underline{r}$ and counter-clockwise angle of rotation θ of a three by three rotation matrix $\underline{\underline{R}}$ is governed by the following relation (a minus sign is necessary for the left-handed version):

$$(\mathrm{SinAxis}(R))_i = r_i \ \sin\theta = \frac{1}{2}\epsilon_{ijk} R_{kj} .$$

This identity seems easier to derive through the "esn" form than through the matrix notation form of the identity. It expands to

$$r_1 \sin\theta = 0.5(R_{32} - R_{23})$$
$$r_2 \sin\theta = 0.5(R_{13} - R_{31})$$
$$r_3 \sin\theta = 0.5(R_{21} - R_{12})$$

(22) The three by three right-handed **rotation matrix** $\underline{\underline{R}}$ corresponding to the instantaneous unit rotation axis $\underline{r}$ and counter-clockwise angle of rotation θ is given by:

$$R_{ij} = r_i r_j + \cos\theta(\delta_{ij} - r_i r_j) - \sin\theta\ \epsilon_{ijk} r_k .$$

Expanded, the above relation becomes:

$$\begin{aligned}
M_{11} &= r_1 r_1 + \cos\theta(1 - r_1 r_1) \\
M_{21} &= r_2 r_1 - \cos\theta r_2 r_1 + r_3 \sin\theta \\
M_{31} &= r_3 r_1 - \cos\theta r_3 r_1 - r_2 \sin\theta \\
\\
M_{12} &= r_1 r_2 - \cos\theta r_1 r_2 - r_3 \sin\theta \\
M_{22} &= r_2 r_2 + \cos\theta(1 - r_2 r_2) \\
M_{32} &= r_3 r_2 - \cos\theta r_3 r_2 + r_1 \sin\theta \\
\\
M_{13} &= r_1 r_3 - \cos\theta r_1 r_3 + r_2 \sin\theta \\
M_{23} &= r_2 r_3 - \cos\theta r_2 r_3 - r_1 \sin\theta \\
M_{33} &= r_3 r_3 + \cos\theta(1 - r_3 r_3)
\end{aligned}$$

Relation 22 yields a right-handed rotation around the axis $\underline{r}$ by angle θ. It can be verified by multiplying by vector $\underline{a}_i$ and comparing the result to the axis-angle formula in section 4.5.

(22a) The **axis-axis** representation of a rotation rotates unit vector axis $\underline{a}$ to unit vector axis $\underline{b}$ by setting

$$\underline{r} = \frac{\underline{a} \times \underline{b}}{|\underline{a} \times \underline{b}|}$$

and letting

$$\theta = \cos^{-1}(a_i b_i)$$

(23) The **inverse** of a rotation matrix $\underline{\underline{R}}$:

$$(\underline{\underline{R}}^{-1})_{ij} = R_{ji}$$

i.e.,

$$\underline{\underline{R}}^{-1} = \underline{\underline{R}}^T$$

(24) When $\underline{\underline{R}}$ is a rotation matrix, $R_{ij} \ R_{kj} = \delta_{ik}$. Likewise, $R_{ji} \ R_{jk} = \delta_{ik}$.

(25a) The **matrix inverse**[6] of a general N by N matrix $\underline{\underline{M}}$:

$$(M^{-1})_{ji} = \frac{\epsilon_{i i_2 i_3 \ldots i_N} \epsilon_{j j_2 j_3 \ldots j_N} M_{i_2 j_2} \ldots M_{i_N j_N}}{(N-1)!\ det M}$$

[6] The author hesitates to expand this daunting expression out!

Please note that the numerator of the above expression involves an enormous number of individual components when written out in a conventional form. The above expression exhibits an incredible economy.

(25b) The **three by three matrix inverse** is given by:

$$(\underline{\underline{M}}^{-1})_{qi} = \frac{\epsilon_{ijk}\,\epsilon_{qnp}\,M_{jn}\,M_{kp}}{2\det M}.$$

The algebraically simplified terms of the above expression are given by:

$$\begin{aligned}
(\underline{\underline{M}}^{-1})_{11} &= (M_{33}\,M_{22} - M_{23}\,M_{32})/\det M\\
(\underline{\underline{M}}^{-1})_{21} &= (M_{23}\,M_{31} - M_{33}\,M_{21})/\det M\\
(\underline{\underline{M}}^{-1})_{31} &= (M_{32}\,M_{21} - M_{22}\,M_{31})/\det M
\end{aligned}$$

$$\begin{aligned}
(\underline{\underline{M}}^{-1})_{12} &= (M_{13}\,M_{32} - M_{33}\,M_{12})/\det M\\
(\underline{\underline{M}}^{-1})_{22} &= (M_{33}\,M_{11} - M_{13}\,M_{31})/\det M\\
(\underline{\underline{M}}^{-1})_{32} &= (M_{12}\,M_{31} - M_{32}\,M_{11})/\det M
\end{aligned}$$

$$\begin{aligned}
(\underline{\underline{M}}^{-1})_{13} &= (M_{23}\,M_{12} - M_{13}\,M_{22})/\det M\\
(\underline{\underline{M}}^{-1})_{23} &= (M_{13}\,M_{21} - M_{23}\,M_{11})/\det M\\
(\underline{\underline{M}}^{-1})_{33} &= (M_{22}\,M_{11} - M_{12}\,M_{21})/\det M
\end{aligned}$$

The factor of 2 canceled out.

To verify the above relationship, we can perform the following computation:

$$\delta_{qr} = (\underline{\underline{M}}^{-1}\underline{\underline{M}})_{qr} = (\underline{\underline{M}}^{-1})_{qi}\,M_{ir}$$

Thus, the above terms simplify to:

$$M_{ir}\,\frac{\epsilon_{ijk}\,\epsilon_{qnp}\,M_{jn}\,M_{kp}}{2\det M} = \frac{\epsilon_{qnp}\,(\epsilon_{ijk}\,M_{ir}\,M_{jn}\,M_{kp})}{2\det M}$$

From identity 17, the determinant cancels out and the above simplifies to:

$$\frac{\epsilon_{qnp}\,\epsilon_{rnp}}{2}$$

which, by the epsilon-delta rule, becomes

$$\begin{aligned}
\frac{\epsilon_{pqn}\,\epsilon_{prn}}{2} &= \frac{\delta_{qr}\,\delta_{nn} - \delta_{qn}\,\delta_{nr}}{2}\\
&= \frac{3\delta_{qr} - \delta_{qr}}{2} = \delta_{qr}
\end{aligned}$$

which verifies the relation.

(26) From the preceding result, we can see that:

$$\epsilon_{ijk}\,M_{jr}\,M_{ks} = 2\det M(M^{-1})_{qi}\,\epsilon_{qrs}.$$

(27) The **Multidimensional Chain Rule** includes the effects of the summations automatically. Additional subscript indices are created as necessary. For instance,

$$(\underline{F}(\underline{G}(\underline{x}(t))))_{i,t} = F_{i,j}\,(\underline{G}(\underline{x}(t)))G_{j,k}\,(\,\underline{x}(t))x_{k,t}.$$

This is equivalent to

$$\frac{\partial(\underline{F}(\underline{G}(\underline{x}(t))))}{\partial t} = \sum_{j=1}^{3}\sum_{k=1}^{3} F_{i,j}\,G_{j,k}\,\frac{\partial x_k\,(t)}{\partial t}$$

where

$$F_{i,j} = \left.\frac{\partial F_i\,(\alpha)}{\partial \alpha_j}\right|_{\underline{\alpha} = \underline{G}(\,\underline{x}(t))}$$

and

$$G_{j,k} = \left.\frac{\partial G_j\,(\beta)}{\partial \beta_k}\right|_{\underline{\beta} = \underline{x}(t)}$$

(28) **Multidimensional Taylor Series**:

$$F_i\,(\underline{x}) = F_i\,(\underline{x}_0) + F_{i,j}\,(\underline{x}_0)(x_j - x_{0j})$$

$$+\frac{1}{2!}\,F_{i,jk}\,(\underline{x}_0)(x_j - x_{0j})(x_k - x_{0k})$$

$$+\frac{1}{3!}\,F_{i,jkp}\,(\underline{x}_0)(x_j - x_{0j})(x_k - x_{0k})(x_p - x_{0p}) + \cdots$$

(29) **Multidimensional Newton's Method** can be derived from the linear terms of the multidimensional Taylor series, in which we are solving $\underline{F}(\underline{x}) = \underline{0}$, and letting $J_{ij} = F_{i,j}$:

$$x_j = x_{0j} - (\underline{\underline{J}}^{-1})_{ji}\,F_i\,(\underline{x}_0).$$

(30) **Orthogonal Decomposition** of a vector $\underline{v}$ in terms of orthonormal (orthogonal unit) vectors $\underline{e}_1$, $\underline{e}_2$, and $\underline{e}_3$:

$$\underline{v} = (\underline{v}\cdot\underline{e}_j)\underline{e}_j.$$

(31) **Change of Basis**: we want to express $v_j\,\underline{e}_j$ in terms of new components times orthonormal basis vectors $\hat{\underline{e}}_j$. A matrix $\underline{\underline{T}}$ re-expresses the numerical representation of a vector relative to the new basis via

$$v_j = T_{ij}\,\hat{v}_j$$

where

$$T_{ij} = (\,\underline{e}_i\cdot\hat{\underline{e}}_j).$$

(32) **The delta rule**:

$$\delta_{ij}\ \textit{esn-expression}_i = \textit{esn-expression}_j\ .$$

(33) **The generalized stokes theorem**:

$$\int_{\partial R} n_j\ \textit{esn-expression}_i\ d\,\sigma = \int_R \textit{esn-expression}_{i,\,j}\ dv$$

Likewise,

$$\int_{\partial R} n_i\ \textit{esn-expression}_i\ d\,\sigma = \int_R \textit{esn-expression}_{i,\,i}\ dv$$

(34) **Multiplying by matrix inverse**: if

$$M_{ij}\, x_j = \textit{esn-expression}_i$$

then

$$(M^{-1})_{pi}\, M_{ij}\, x_j = (M^{-1})_{pi}\ \textit{esn-expression}_i$$

or, simplifying and renaming p back to j,

$$x_j = (M^{-1})_{ji}\ \textit{esn-expression}_i$$

Note the transpose relationship of matrices $\underline{\underline{M}}$ and $\underline{\underline{M}}^{(-1)}$ in the first and last equations.

(35) **Conversion of Rotation matrix R_{ij} to the axis-angle formulation**:

We first note that any collection of continuously varying axes $\underline{r}$ and angles θ produces a continuous rotation matrix function, via identity 22. However, it is not true that this relation is completely invertible, due to an ambiguity of sign: the same matrix is produced by different axis-angle pairs. For instance, the matrix produced by $\underline{r}$ and θ is the same matrix produced by $-\underline{r}$ and $-\theta$. In fact, if $\theta = 0$ then there is no net rotation, and any unit vector $\underline{r}$ produces the identity matrix (null rotation).

Since

$$|\sin\theta| = \frac{|\epsilon_{ijk}\, R_{ij}\,|}{2}, \text{ and}$$

$$\cos\theta = \frac{R_{ii} - 1}{2}.$$

the angle θ is given by

$$\theta = Atan(|\sin\theta|, \cos\theta).$$

If $\theta = 0$, any axis suffices, and we are finished.

Otherwise, if $\theta \neq 0$, we need to know the value of $\underline{r}$. In that case, if $\theta \neq \pi$, then

$$r_k = \frac{-\epsilon_{ijk}\, R_{ij}}{2\sin\theta},$$

Otherwise, if $\theta = \pi$, then since the original rotation matrix

$$R_{ij} = 2r_i\, r_j - \delta_{ij}\ ,$$

then

$$r_i\, r_j = \frac{R_{ij} + \delta_{ij}}{2}$$

Letting

$$M_{ij} = \frac{R_{ij} + \delta_{ij}}{2},$$

we can solve for r_i, by taking the ratio of non-diagonal and square roots of nonzero diagonal terms, via

$$r_i = \frac{M_{i[j]}}{\sqrt{M_{[j][j]}}}$$

Note that we are using $[j]$ to be an independent index as defined near the bottom of page 3. We choose the value of j to be such that $M_{[j][j]}$ is its largest value.

6 Extensions to the tensor notation for Multi-component equations

In this section, we introduce the no-sum notation and special symbols for quaternions.

6.1 The "no-sum" operator

The classical summation convention is incomplete as it usually stands, in the sense that not all multi-component equations (i.e, those involving summation signs and subscripted variables) can be represented. In the scalar convention, the default is not to sum indices, so that summation signs must be written explicitly if they are desired. Using the classical summation convention, summation is the default, and there is no convenient way not to sum over a repeated index (other than perhaps an awkward comment in the margin, directing the reader that the equation is written with no implicit summation).

Thus, an extension of the notation is proposed in which an explicit "no-sum" or "free-index" operator prevents the summation over a particular index within a term. The no-sum operator is represented via $\sum_i$ or (which is easier to write) a subscripted prefix parenthesis $(_i \quad)$. This modification extends the types of formulations we can represent, and augments algebraic manipulative skills. Expressions involving the no-sum operator are found in calculations which take place in a particular coordinate system; without the extension, all of the terms are tensors, and transform correctly from one coordinate system to another. For instance, a diagonal matrix

$$\underline{\underline{M}} = \begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{pmatrix}$$

is not diagonal in all coordinate systems. Nonetheless, it can be convenient to do the calculation in the coordinate system in which M is diagonal.

$$M_{ij} = \sum_i (a_i \delta_{ij}) = (_i \ a_i \delta_{ij})$$

Some of the identities for this augmented notation are found later in this document.

The ease of manipulation and the compactness of representation are the main advantages of the summation convention. Generally, an expression is converted from conventional matrix and vector notation into the summation notation, simplified in the summation form, and then converted back into matrix and vector notation for interpretation. Sometimes, however, there is no convenient way to express the result in conventional matrix notation.

6.2 Quaternions

The other proposed extensions to the notation aid in manipulating quaternions. For convenience, a few new special symbols are defined to take quaternion inverses, quaternion products, products of quaternions and vectors, and conversions of quaternions to rotation matrices. A few identities involving these symbols are also presented in this section.

6.2.1 Properties of Quaternions

A *quaternion* is a four dimensional mathematical object which is a linear combination of four independent basis vectors: $\mathbf{1}$, $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$, satisfying the following relations:

$$\begin{aligned} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 &= -1 \\ \mathbf{ijk} &= -1 \end{aligned}$$

By pre- and post-multiplying by any of $\mathbf{i}$, $\mathbf{j}$, or $\mathbf{k}$, it is straightforward to show that

$$\begin{aligned} \mathbf{ij} &= \mathbf{k} \\ \mathbf{jk} &= \mathbf{i} \\ \mathbf{ki} &= \mathbf{j} \\ \mathbf{ji} &= -\mathbf{k} \\ \mathbf{kj} &= -\mathbf{i} \\ \mathbf{ik} &= -\mathbf{j} \end{aligned}$$

Thus, a quaternion

$$\underline{q} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}$$

can represented as

$$\underline{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k},$$

and quaternion multiplication takes place by applying the above identities, to obtain

$$\begin{aligned} \underline{p}\,\underline{q} &= (p_0 + p_1 \mathbf{i} + p_2 \mathbf{j} + p_3 \mathbf{k})(q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}) \\ &= p_0 (q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}) \\ &\quad + p_1 \mathbf{i}(q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}) \\ &\quad + p_2 \mathbf{j}(q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}) \\ &\quad + p_3 \mathbf{k}(q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}) \\ &= p_0 q_0 + p_0 q_1 \mathbf{i} + p_0 q_2 \mathbf{j} + p_0 q_3 \mathbf{k} \\ &\quad + p_1 \mathbf{i} q_0 + p_1 q_1 (\mathbf{ii}) + p_1 q_2 (\mathbf{ij}) + p_1 q_3 (\mathbf{ik}) \\ &\quad + p_2 \mathbf{j} q_0 + p_2 q_1 (\mathbf{ji}) + p_2 q_2 (\mathbf{jj}) + p_2 q_3 (\mathbf{jk}) \\ &\quad + p_3 \mathbf{k} q_0 + p_3 q_1 (\mathbf{ki}) + p_3 q_2 (\mathbf{kj}) + p_3 q_3 (\mathbf{kk}) \\ &= p_0 q_0 - p_1 q_1 p_2 q_2 - p_3 q_3 \\ &\quad + \mathbf{i}(p_1 q_0 + p_0 q_1 + p_2 q_3 - p_3 q_2) \\ &\quad + \mathbf{j}(p_2 q_0 + p_0 q_2 + p_3 q_1 - p_1 q_3) \\ &\quad + \mathbf{k}(p_3 q_0 + p_0 q_3 + p_1 q_2 - p_2 q_1) \end{aligned}$$

6.2.2 The geometric interpretation of a quaternion

A quaternion can be represented as a 4-D composite vector, consisting of a scalar part s and a three dimensional vector portion $\underline{v}$:

$$\underline{q} = \begin{pmatrix} s \\ \underline{v} \end{pmatrix}.$$

A quaternion is intimately related to the axis-angle representation for a three dimensional rotation (see Figure 1). The vector portion $\underline{v}$ of a unit quaternion is the rotation axis, scaled by the sine of half of the rotation angle. The scalar portion, s is the cosine of half the rotation angle.

$$\underline{q} = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2)\underline{r} \end{pmatrix},$$

where $\underline{r}$ is a unit vector, and $q \cdot q = 1$

Thus, the conversion of a unit quaternion to and from a rotation matrix can be derived using the axis-angle formulas 22 and 35. Remember that the quaternion angle is half of the axis-angle angle.

Quaternion Product:

A more compact form for quaternion multiplication is

$$\begin{pmatrix} s_1 \\ \underline{v}_1 \end{pmatrix} \begin{pmatrix} s_2 \\ \underline{v}_2 \end{pmatrix} = \begin{pmatrix} s_1 s_2 - \underline{v}_1 \cdot \underline{v}_2 \\ s_1 \underline{v}_2 + s_2 \underline{v}_1 + \underline{v}_1 \times \underline{v}_2 \end{pmatrix}.$$

Note that

$$\begin{pmatrix} s_1 \\ \underline{v}_1 \end{pmatrix} \begin{pmatrix} s_2 \\ \underline{v}_2 \end{pmatrix} \neq \begin{pmatrix} s_2 \\ \underline{v}_2 \end{pmatrix} \begin{pmatrix} s_1 \\ \underline{v}_1 \end{pmatrix}$$

With this notation, quaternion-vector, vector-quaternion, and vector-vector multiplication rules become clear: to represent a vector, we set the scalar part of the quaternion to zero, and use the above relation to perform the multiplications.

Quaternion Inverse:

The **inverse** of a quaternion $\underline{q}$ is another quaternion $\underline{q}^{-1}$ such that $\underline{q}\,\underline{q}^{-1} = 1$. It is easily verified that

$$\underline{q}^{-1} = \begin{pmatrix} s \\ -\underline{v} \end{pmatrix} / (s^2 + \underline{v} \cdot \underline{v}).$$

Rotating a vector with quaternions is achieved by pre-multiplying the vector with the quaternion, and postmultiplying by the quaternion inverse. We use this property to derive the conversion formula from quaternions to rotation matrices.

Quaternion Rotation:

$$(Rot(\underline{v}))_i = (\underline{q}\underline{v}\underline{q}^{-1})_i, i = 1, 2, 3.$$

With this identity, it is possible to verify the geometric interpretation of quaternions and their relationship to the axis-angle formulation for rotation.

The most straightforward way to verify this relation is to expand the vector $\underline{v}$ into components parallel to $\underline{r}$ and those perpendicular to it, plug into the rotation formula, and compare to the equation in section 4.5.

i.e., to evaluate

$$(Rot(\underline{v})) = (\underline{q}((\underline{v} \setminus r) + \alpha r)\underline{q}^{-1})$$

6.3 Extended Symbols for Quaternion Manipulations

The author has developed a few special symbols to help write and manipulate quaternion quantities more conveniently. We define all quaternion components as going from 0 to 3, with the vector part still going from 1 to 3, with zero for the scalar component.

If we are not using the full range of a variable (going, say from 1 to 3 when the original goes from 0 to 3), we need to explicitly denote that.

We hereby extend our Kronecker delta to allow zero in the subscripts, so

$$\delta_{00} = 1$$

and

$$\delta_{0i} = 0, \quad i \neq 0.$$

We create a new permutation symbol, ϵ^0 which allows zero in the subscripts. It will be $+1$ for even permutations of $(0, 1, \ldots, 3)$, and -1 for odd permutations.

The quaternion **inverter** structure constant ν_{ij} allows us to compute quaternion inverses:

$$\boxed{q_I^{-1} = \nu_{ij}\; q_j / (q_K\, q_K).}$$

where

$$\nu_{ij} = \begin{cases} 1, & i = j = 0 \\ -1, & i = j \neq 0 \\ 0, & otherwise \end{cases}.$$

Note, using the full range for i and j, that

$$\boxed{\nu_{ij} = 2\delta_{i0}\,\delta_{j0} - \delta_{ij}\,.}$$

The **quaternion product** structure constant κ_{Kij} allows us to multiply two quaternions $\underline{p}$ and $\underline{q}$ via

$$(\underline{pq})_K = \kappa_{Kij}\, p_i\, q_j$$

where the nonzero κ components are given by: (i, j, k = 1 . . . 3)

$$\begin{aligned}\kappa_{000} &= 1\\ \kappa_{ij0} &= \delta_{ij}\\ \kappa_{i0k} &= \delta_{ik}\\ \kappa_{0jk} &= -\delta_{jk}\\ \kappa_{ijk} &= \epsilon_{ijk}\end{aligned}$$

Note, using the full range, for i, j, and k that

$$\boxed{\kappa_{ijk} = \delta_{ij}\,\delta_{k0} + \delta_{ik}\,\delta_{j0} - \delta_{i0}\,\delta_{jk} + \epsilon^0_{0ijk}}$$

The **quaternion-vector product** structure constant $\kappa^v{}_{ki\ell}$ allows us to multiply a quaternion $\underline{q}$ and a vector $\underline{v}$ via

$$\boxed{(\underline{qv})_k = \kappa^v{}_{ki\ell}\, q_i\, v_\ell}$$

where the nonzero κ^v components are given by: (i, j, k = 1 . . . 3)

$$\begin{aligned}\kappa^v{}_{i0k} &= \delta_{ik}\\ \kappa^v{}_{0jk} &= -\delta_{jk}\\ \kappa^v{}_{ijk} &= \epsilon_{ijk}\end{aligned}$$

Note, using the full range for i, j, and k, that

$$\boxed{\kappa^v{}_{ijk} = \delta_{ik}\,\delta_{j0} - \delta_{i0}\,\delta_{jk} + \epsilon^0_{0ijk}}$$

The **vector-quaternion product** structure constant ${}^v\kappa_{k\ell i}$ allows us to multiply a vector $\underline{v}$ and a quaternion $\underline{q}$ via

$$\boxed{(\underline{vq})_k = {}^v\kappa_{k\ell i}\, v_\ell\, q_i}$$

where the nonzero ${}^v\kappa$ components are given by: (i, j, k = 1 . . . 3)

$$\begin{aligned}{}^v\kappa_{ij0} &= \delta_{ij}\\ {}^v\kappa_{0jk} &= -\delta_{jk}\\ {}^v\kappa_{ijk} &= \epsilon_{ijk}\end{aligned}$$

Note, using the full range for i, j, and k that

$$\boxed{{}^v\kappa_{ijk} = \delta_{ij}\,\delta_{k0} - \delta_{i0}\,\delta_{jk} + \epsilon^0_{0ijk}}$$

The **vector-vector product** is the conventional cross product.

The **quaternion to rotation matrix** structure constant $\mu_{ijk\ell}$ allows us to create a rotation[7] matrix R

$$\boxed{R_{ij} = \mu_{ijk\ell}\, q_k\, q_\ell\, / (q_N\, q_N\,).}$$

It is straightforward to express μ in terms of the κ's:

$$\boxed{\mu_{ijk\ell} = \kappa_{ikp}\, {}^v\kappa_{pjN}\, \nu_{N\ell}}$$

To derive this relation, we note that the i-th component of the rotation of vector $\underline{a}$ is given by: (i, j, k = 1 . . . 3)

$$\begin{aligned}(Rot(\underline{a}))_i &= (qaq^{-1})_i\,, i = 1,2,3.\\ &= \kappa_{ikp}\ \ q_k\,(aq^{-1})_p\\ &= \kappa_{ikp}\ \ q_k\, {}^v\kappa_{pjN}\, a_j\ \ (q^{-1})_N\\ &= \kappa_{ikp}\ \ q_k\, {}^v\kappa_{pjN}\, a_j\ \ \nu_{N\ell}\, q_\ell\, /(q \cdot q)\\ &= (\kappa_{ikp}\, {}^v\kappa_{pjN}\ \ \nu_{N\ell}\,)\ \ q_k\, q_\ell\, /(q\cdot q) a_j\end{aligned}$$

Since we can represent these rotations with a three dimensional rotation matrix $\underline{\underline{R}}$, and

$$(Rot(\underline{a}))_i = R_{ij}\, a_j$$

for all a_j, we can eliminate a_j from both sides of the equation, yielding

$$R_{ij} = \kappa_{ikp}\, {}^v\kappa_{pjN}\, \nu_{N\ell}\, q_k\, q_\ell\, /(q\cdot q).$$

Thus,

$$\boxed{\mu_{ijk\ell} = \kappa_{ikp}\, {}^v\kappa_{pjN}\, \nu_{N\ell}\,.}$$

7 What is Angular Velocity in 3 and greater dimensions?

Using the axis/angle representation of rotation, matrices, and quaternions, we define, derive, interpret and demonstrate the compatibility between the two main classic equations relating the angular velocity vector $\underline{\omega}$, the rotation itself, and the derivative of the rotation.

Matrix Eqn:

$$\underline{\underline{m}}' = \underline{\omega} \times \underline{\underline{m}}$$

[7] Note that $q_N\, q_N = q \cdot q$ in the following equation.

or

$$\underline{\underline{m}}' = \underline{\underline{\omega}}^* \; \underline{\underline{m}}$$

where

$$(\underline{\underline{\omega}}^*)_{ik} = \epsilon_{ijk}\,\omega_j$$

Quaternion Eqn:

$$\underline{q}' = \frac{1}{2} \begin{pmatrix} 0 \\ \underline{\omega} \end{pmatrix} \underline{q}$$

Definition of angular velocity in three dimensions

Consider a time varying rotation Rot(t), (for instance, represented with a matrix function or quaternion function), which brings an object from body coordinates to world coordinates.

In three dimensions, we define angular velocity $\underline{\omega}$ as the vector quantity

1. whose direction is the instantaneous unit vector axis of rotation of the time varying rotation and
2. whose magnitude is the angular rate of rotation around the instantaneous axis.

We derive angular velocity both for matrix representations and for quaternion representations of rotation.

The direction of the instantaneous axis of rotation can be obtained by using a matrix-to-axis operator on the the relative rotation from t to $t+h$.

In symbolic form, angular velocity is given by

$$\underline{\omega} = \lim_{h\to 0} \frac{\text{Axis}(\text{RelRot}(t,h))\ \text{Angle}(\text{RelRot}(t,h))}{h}$$

for either representation method.

Matrix representation

Let $\underline{\underline{M}}(t)$ be a time varying rotation matrix which takes us from body coordinates to world coordinates, and let $\underline{\underline{N}}(t,h)$ be the relative rotation from M(t) to M(t+h).

In other words, since N takes us from M(t) to M(t+h),

$$\underline{\underline{M}}(t+h) = \underline{\underline{N}}(t,h)\ \underline{\underline{M}}(t)$$

so

$$\begin{aligned}
\text{RelRot}(t,h) &= \underline{\underline{N}}(t,h) \\
\underline{\underline{N}}(t,h) &= \underline{\underline{M}}(t+h)\ \underline{\underline{M}}^T(t) \\
N_{ij}(t,h) &= M_{ip}(t+h)\ (M^T)_{pj}(t) \\
N_{ij}(t,h) &= M_{ip}(t+h)\ M_{jp}(t) \\
&= \left(M_{ip}(t) + hM'_{ip} + \mathcal{O}(h^2)\right)\ M_{jp}(t) \\
\text{so} & \\
N_{ij}(t,h) &= \delta_{ij} + hM'_{ip}(t)\ M_{jp}(t) + \mathcal{O}(h^2)
\end{aligned}$$

Expressing $\underline{\omega}$ in terms of M and M'

To find ω, we can convert $N(t,h)$ to axis/angle form, to find the direction of the axis and the angular rate of rotation, and take the limit as h goes to zero.

$$\underline{\omega} = \lim_{h\to 0} \frac{\text{Axis}(N(t,h))\ \text{Angle}(N(t,h))}{h}$$

However, a simpler method is available. We note that as $h \to 0$, Angle(N) $\to$ sin(Angle(N)). Thus, the product of the matrix Axis operator and the Angle operator in the limit will equal the matrix SinAxis operator (which is easier to compute).

$$\begin{aligned}
\underline{\omega} &= \lim_{h\to 0} \frac{\text{SinAxis}(\underline{\underline{N}}(t,h))}{h} \\
\omega_i &= \tfrac{1}{2} \lim_{h\to 0} \frac{\epsilon_{pqi}\ N_{pq}(t,h)}{h} \\
&= \tfrac{1}{2} \lim_{h\to 0} \epsilon_{pqi}\ \left(\delta_{pq} + hM'_{ps}(t)\ M_{qs}(t)\right) \\
\textit{Thus}, & \\
\omega_i &= \tfrac{1}{2}\ \epsilon_{ipq}\ M'_{ps}(t)\ M_{qs}(t)
\end{aligned}$$

Note that the SinAxis operator is described in identity 21; also note that

$$\omega_{[i]} = M'_{[i+1]s}\, M_{[i+2]s})$$

or

$$\begin{aligned}
\omega_1 &= M'_{2s}\, M_{3s} \\
\omega_2 &= M'_{3s}\, M_{1s} \\
\omega_3 &= M'_{1s}\, M_{2s}
\end{aligned}$$

Expressing M and M' in terms of $\underline{\omega}$

We can express $M'_{ps}(t)\ M_{qs}(t)$ in terms of $\underline{\omega}$ by multiplying both sides by ϵ_{ijk}.

$$\begin{aligned}
\omega_j &= \tfrac{1}{2}\,\epsilon_{pqj}\ M'_{ps}(t)\ M_{qs}(t) \\
\epsilon_{ijk}\,\omega_j &= \tfrac{1}{2}\,\epsilon_{ijk}\,\epsilon_{pqj}\ M'_{ps}(t)\ M_{qs}(t) \\
&= \tfrac{1}{2}\,(\delta_{jp}\,\delta_{kq} - \delta_{jq}\,\delta_{kp})\ M'_{ps}(t)\ M_{qs}(t) \\
&= \tfrac{1}{2}\,(M'_{js}(t)\ M_{ks}(t) - M'_{ks}(t)\ M_{js}(t)) \\
\epsilon_{ijk}\,\omega_j &= M'_{js}(t)\ M_{ks}(t)
\end{aligned}$$

Expressing M' in terms of $\underline{\omega}$ and M

We take the equation in the previous section, and multiply by M_{kp}:

$$\begin{aligned}
M_{kp}\,\epsilon_{ijk}\,\omega_j &= M'_{js}(t)\ M_{ks}(t) M_{kp} \\
M_{kp}\,\epsilon_{ijk}\,\omega_j &= M'_{jp}(t) \\
&\text{so} \\
M'_{jp}(t) &= \epsilon_{ijk}\,\omega_j\,M_{kp}
\end{aligned}$$

Thus, we have derived the matrix equation presented in the introduction.

Quaternion representation

Let $\underline{q}(t)$ be a time varying rotation quaternion which takes us from body coordinates to world coordinates, and let $\underline{p}(t,h)$ be the relative rotation from q(t) to q(t+h).

In other words, since p takes us from q(t) to q(t+h),

$$\underline{q}(t+h) = \underline{p}(t,h)\ \underline{q}(t)$$

so

$$\begin{aligned}
\text{RelRot}(t,h) &= \underline{p}(t,h) \\
\underline{p}(t,h) &= \underline{p}(t+h)\ \underline{p}^{-1}(t)
\end{aligned}$$

Expressing $\underline{\omega}$ in terms of q and q'

To find ω, we can convert $p(t,h)$ to axis/angle form, to find the direction of the axis and the angular rate of rotation, and take the limit as h goes to zero.

$$\underline{\omega} = \lim_{h\to 0} \frac{\text{Axis}(p(t,h))\ \text{Angle}(p(t,h))}{h}$$

However, as in the case with the matrices, a simpler method is available. We note that as $h \to 0$, Angle(N) $\to$ sin(Angle(N)). Thus, as before, the product of the matrix Axis operator and the Angle operator in the limit will equal twice the quaternion VectorPart operator which returns the vector portion of the quaternion.[8]

$$\begin{aligned}
\underline{\omega} &= 2\lim_{h\to 0} \frac{\text{VectorPart}(\underline{p}(t,h))}{h} \\
&= 2\lim_{h\to 0} \frac{\text{VectorPart}(\underline{q}(t+h)q^{-1})}{h} \\
&= 2\text{VectorPart}(1 + q'(t)q^{-1})
\end{aligned}$$

so

$$\begin{pmatrix} 0 \\ \underline{\omega} \end{pmatrix} = 2q'(t)q^{-1}$$

Expressing q' in terms of $\underline{\omega}$ and q

We take the equation in the previous section, and multiply by $\frac{1}{2}q$ on the right:

$$q'(t) = \tfrac{1}{2}\begin{pmatrix} 0 \\ \underline{\omega} \end{pmatrix} q$$

Thus, we have derived the quaternion equation presented in the introduction.

Relating angular velocity to rotational basis vectors

Let basis vector e_p be the p-th column of M, so the i-th element of the p-th basis vector is given by

$$(\underline{e}_p)_i = M_{ip}$$

Thus,

$$(\underline{e}'_p)_i = \epsilon_{ijk}\,\omega_j\,(\underline{e}_p)_k$$

or

$$\underline{e}'_p = \omega_j \times \underline{e}_p$$

[8] The VectorPart operator can be thought of as a HalfSinAxis operator on the unit quaternion.

What about non-unit quaternions?

Let $\underline{Q} = m\underline{q}$ be a non-unit quaternion (with magnitude m, and q is a unit quaternion).

$$\begin{aligned} Q' &= (m\underline{q})' \\ &= m\underline{q}' + m'\underline{q} \\ &= m\tfrac{1}{2}\omega q + \underline{q} d/dt(Q \cdot Q)^{1/2} \\ &= \tfrac{1}{2}\omega Q + \underline{Q}/(Q \cdot Q)^{1/2} d/dt(Q \cdot Q)^{1/2} \\ &= \tfrac{1}{2}\omega Q + Q \cdot Q'\underline{Q}/Q \cdot Q \end{aligned}$$

So

$$(I - QQ)Q' = \tfrac{1}{2}\omega Q$$

or

$$Q' = \tfrac{1}{2}(I + \tfrac{QQ}{(1 - Q \cdot Q)})\omega Q$$

Alternate derivation of $\underline{\omega}$ (works in N dimensions)

Let M(t) be an N dimensional rotation matrix which brings an object from body coordinates to world coordinates.

Note that

$$M_{in} M_{kn} = \delta_{ik}$$

Taking the derivative of both sides,

$$M'_{in} M_{kn} + M_{in} M'_{kn} = 0$$

which means

$$\begin{aligned} M'_{in} M_{kn} &= -M_{in} M'_{kn} = A_{ik} \\ &= -A_{ki} \end{aligned}$$

Thus,

$$M'_{in} M_{kn} = A_{ik}$$

To solve for M', multiply both sides by M_{kj}. So

$$M'_{ij} = A_{ik} M_{kj}$$

In N dimensions, the antisymmetric matrix A takes the place of the angular velocity vector ω.

8 Examples.

Example 1. To simplify the vector $\underline{a} \times (\underline{b} \times \underline{c})$ we use the re-association rules, the rearrangement rules, the permutation symbol subscript interchange rules, the ϵ-δ rule, the δ simplification rules, and the re-association and rearrangement rules:

$$\begin{aligned} (\underline{a} \times (\underline{b} \times \underline{c}))_i &= \epsilon_{i\boxed{j}\boxed{k}}\, a_{\boxed{j}}\, (\underline{b} \times \underline{c})_{\boxed{k}} \\ &= \epsilon_{ijk}\, a_j\, \epsilon_{knp}\, b_n\, c_p \\ &= (\epsilon_{ijk}\, \epsilon_{knp})a_j\, b_n\, c_p \\ &= (\epsilon_{kij}\, \epsilon_{knp})a_j\, b_n\, c_p \\ &= (\delta_{in}\, \delta_{jp} - \delta_{ip}\, \delta_{jn})a_j\, b_n\, c_p \\ &= a_j\, b_i\, c_j - a_j\, b_j\, c_i \\ &= (a_j\, c_j)b_i - (a_j\, b_j)c_i \end{aligned}$$

Therefore $\underline{a} \times (\underline{b} \times \underline{c}) = (\underline{a} \cdot \underline{c})\underline{b} - (\underline{a} \cdot \underline{b})\underline{c}$. Note that $\underline{a} \times (\underline{b} \times \underline{c}) \neq (\underline{a} \times \underline{b}) \times \underline{c}$.

Example 2. The equation $c = a_i\, b_i$ has two repeated subscripts in the term on the right, so the index "i" is bound to that term with an implicit summation. This is a scalar equation because there are no free indices on either side of the equation. In other words, $c = a_1\, b_1 + a_2\, b_2 + a_3\, b_3 = \underline{a} \cdot \underline{b}$

Example 3. To show that $\underline{a} \cdot (\underline{b} \times \underline{c}) = (\underline{a} \times \underline{b}) \cdot \underline{c} = \det(a, b, c)$

$$\begin{aligned} \underline{a} \cdot (\underline{b} \times \underline{c}) &= a_i\, (\underline{b} \times \underline{c})_i \\ &= a_i\, \epsilon_{ijk}\, b_j\, c_k \\ &= \epsilon_{ijk}\, a_i\, b_j\, c_k \\ &= \det(a, b, c) \\ \epsilon_{ijk}\, a_i\, b_j\, c_k &= (\epsilon_{ijk}\, a_i\, b_j)c_k \\ &= (\epsilon_{kij}\, a_i\, b_j)c_k \\ &= (\underline{a} \times \underline{b})_k\, c_k \\ &= (\underline{a} \times \underline{b}) \cdot \underline{c} \end{aligned}$$

Example 4. To derive identity (21) from identity (22), we multiply (22) by ϵ_{qji}:

$$\epsilon_{qji}\, R_{ij} = \epsilon_{qji}\, (r_i\, r_j + c_\theta\, (\delta_{ij} - r_i\, r_j)) + \epsilon_{qji}\, s_\theta\, \epsilon_{ikj}\, r_k\, .$$

The second factor of the first term on the right side of the above equation is symmetric in subscripts "i" and "j," so by the symmetric identity (10), the term is zero. So,

$$\epsilon_{qji}\, R_{ij} = \epsilon_{qji}\, \epsilon_{ikj}\, \sin\theta\, r_k\, .$$

Since

$$(\epsilon_{qji}\epsilon_{ikj}) = \epsilon_{iqj}\epsilon_{ikj}$$

$$= \delta_{qk}\delta_{jj} - \delta_{qj}\delta_{jk} = 3\delta_{qk} - \delta_{qk} = 2\delta_{qk}$$

$$\epsilon_{qji} R_{ij} = 2\delta_{qk} \sin\theta\ r_k$$

and

$$\epsilon_{qji} R_{ij} = 2\sin\theta\ r_q$$

which completes the derivation.

Example 5. To verify the matrix inverse identity (25), we multiply by the original matrix M_{im} on both sides, to see if we really have an identity.

$$(\underline{\underline{M}}^{-1})_{qi} M_{im} =^? \frac{1}{2}\frac{M_{im}\epsilon_{ijk}\epsilon_{qnp} M_{jn} M_{kp}}{\det M}.$$

$$\delta_{qm} =^? \frac{1}{2}\frac{\epsilon_{qnp}(\epsilon_{ijk} M_{im} M_{jn} M_{kp})}{\det M}$$

Using identity (17), this simplifies to

$$\delta_{qm} =^? \frac{1}{2}\frac{(\epsilon_{qnp}\epsilon_{mnp})\det M}{\det M}$$

so

$$\delta_{qm} = \frac{1}{2}(2\delta_{qm})$$

Thus, the identity is verified.

Example 6. To verify identity (26), we multiply by the determinant, and by ϵ_{qrs}. Identity (11) is used to eliminate the factor of 2. The other details are left to the reader.

Example 7 To discover the inverse of a matrix $\underline{\underline{A}}$ of the form

$$\underline{\underline{A}} = a\underline{\underline{I}} - b\underline{x}\underline{x}:$$

i.e.,

$$A_{ij} = (a\delta_{ij} - bx_i x_j).$$

We are looking for B_{jk} such that

$$A_{ij} B_{jk} = \delta_{ij}.$$

We assume the form of the inverse, within an undetermined constant σ:

$$B_{jk} = (\frac{\delta_{jk}}{a} + \sigma x_j x_k).$$

Since

$$A_{ij} B_{jk} = (a\delta_{ij} - bx_i x_j)(\frac{\delta_{jk}}{a} + \sigma x_j x_k)$$

$$= \delta_{ik} - \frac{b}{a}x_i\ x_k + a\sigma x_i x_k - b\sigma x_i x_k (x_j x_j)$$

$$= \delta_{ik} + (a\sigma - b/a - b\sigma(x_j x_j))x_i x_k,$$

we conclude that,

$$a\sigma - b/a - b\sigma(x_j x_j) = 0,$$

so

$$\sigma = \frac{b}{a^2 - ab(\underline{x}\cdot\underline{x})}.$$

Thus,

$$\underline{\underline{A}}^{-1} = \frac{\underline{\underline{I}}}{a} + \frac{b\underline{x}\ \underline{x}}{a^2 - ab(\underline{x}\cdot\underline{x})}$$

Example 8. Verifying the relationship between the axis-angle formulation and quaternions:

Consider a unit quaternion

$$\underline{q} = \begin{pmatrix} s \\ v \end{pmatrix} = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2)\underline{r} \end{pmatrix}$$

Thus,

$$s = c_{\theta/2}$$

and

$$\underline{v} = s_{\theta/2}\,\underline{r}.$$

The rotation

$$\begin{aligned} \begin{pmatrix} 0 \\ Rot(a) \end{pmatrix} &= qaq^{-1} \\ &= \begin{pmatrix} s \\ \underline{v} \end{pmatrix}\begin{pmatrix} 0 \\ \underline{a} \end{pmatrix}\begin{pmatrix} s \\ -\underline{v} \end{pmatrix} \\ &= \begin{pmatrix} s \\ \underline{v} \end{pmatrix}\begin{pmatrix} \underline{a}\cdot\underline{v} \\ s\underline{a} - \underline{a}\times\underline{v} \end{pmatrix} \end{aligned}$$

The scalar part becomes

$$sa\cdot v - sa\cdot v = 0$$

The vector part becomes

$$s(sa - a\times v) + (a\cdot v)v + v\times(sa - a\times v)$$

The i-th component of the vector part of the rotation becomes

$$\begin{aligned} &= s^2 a_i - 2s(a\times v)_i + (a\cdot v)v_i - \epsilon_{ijk}v_j\epsilon_{kpq}a_p v_q \\ &= s^2 a_i - 2s(a\times v)_i + (a\cdot v)v_i - (\epsilon_{ijk}\epsilon_{kpq})v_j a_p v_q \\ &= s^2 a_i - 2s(a\times v_i + (a\cdot v)v_i - (v_j a_i v_j - v_j a_j v_i) \\ &= s^2 a_i - v_j v_j a_i + 2(v_j a_j v_i - 2sa\times v \\ &= c^2_{\theta/2}a_i - s^2_{\theta/2}a_i + 2s^2_{\theta/2}r_i r_j a_j - s_\theta\epsilon_{ijk}a_j r_k \\ &= c_\theta a_i - s_\theta\epsilon_{ijk}a_j r_k + (1-c_\theta)r_i r_j a_j \\ &= (c_\theta\delta_{ij} - s_\theta\epsilon_{ijk}r_k + (1-c_\theta)r_i r_j)a_j \\ &= (r_i r_j + c_\theta(\delta_{ij} - r_i r_j) - s_\theta\epsilon_{ijk}r_k)a_j \end{aligned}$$

which verifies the relationship (see identity 22).

8.1 Sample identities using the no-sum operator.

A few identities using the no-sum operator are listed. This is not an exhaustive exploration; the purpose is to give an intuitive feeling for the terminology. It is hoped that this new terminology may be helpful in the development of multicomponent symbolic manipulative skill.

$$\sum_i (a_i b_i) = (\begin{pmatrix} a_1 b_1 \\ a_2 b_2 \\ a_3 b_3 \end{pmatrix})_i$$

$$\sum_i (\delta_{ij} a_i) = \sum_j (\delta_{ij} a_j) = (\begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{pmatrix})_{ij}$$

$$b_i \sum_i (\delta_{ij} a_i) = \sum_j (a_j b_j)$$

$$b_j \sum_i (\delta_{ij} a_i) = \sum_i (a_i b_i)$$

$$\epsilon_{plk} \sum_k (\delta_{kj} g_k) = \sum_j (\epsilon_{plj} g_j)$$

$$1_i \sum_i (a_i b_i) = a_i b_i$$

In the prefix subscript form, the nosum symbol is not written into the expression — the prefix subscript is sufficient by itself. Thus, the above expressions may also be represented via:

$$({}_i \; a_i b_i) = (\begin{pmatrix} a_1 b_1 \\ a_2 b_2 \\ a_3 b_3 \end{pmatrix})_i$$

$$({}_i \; \delta_{ij} a_i) = ({}_j \; \delta_{ij} a_j) = (\begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{pmatrix})_{ij}$$

$$b_i ({}_i \; \delta_{ij} a_i) = ({}_j \; a_j b_j)$$

$$b_j ({}_i \; \delta_{ij} a_i) = ({}_i \; a_i b_i)$$

$$\epsilon_{plk} ({}_k \; \delta_{kj} g_k) = ({}_j \; \epsilon_{plj} g_j)$$

$$1_i ({}_i \; a_i b_i) = a_i b_i$$

Please note that many of the above algebraic sub-expressions are not tensors — they do not follow the tensor transformation rules from one coordinate system to another.

8.2 Equations of motion of rigid Bodies using Esn.

Given a rigid body with mass m, density $\rho(x, y, z)$, position of center of mass in the world $\underline{x}$, position of center of mass in the body at its origin, a net force $\underline{F}$, net torque $\underline{T}$, momentum $\underline{\mathrm{p}}$, angular momentum $\underline{\mathrm{L}}$, angular velocity $\underline{\omega}$, rotation quaternion $\underline{q}$, rotational inertia tensor in body coordinates $\underline{\underline{I}}^{body}$, the equations of motion for the rigid body in the lab frame becomes:

$$d/dt \; x_i = p_i / m$$

$$d/dt \; q_i = 1/2 \kappa^v{}_{ijk} \, \omega_j \, q_k$$

$$d/dt \; p_i = F_i$$

$$d/dt \; L_i = T_i$$

where

$$\omega_i = (I^{(-1)})_{ij} L_j ,$$

$$(I^{(-1)})_{ij} = m_{ip} m_{js} ((I^{body})^{(-1)})_{ps}$$

$$m_{ij} = \mu_{ijkl} \, q_k \, q_l / (q \cdot q),$$

and

$$(I^{body})_{ij} = \int_{body} \rho(x, y, z)(\delta_{ij} x_k x_k - x_i x_j) \, dx \, dy \, dz.$$

A point in the body $\underline{b}^{body}$ transforms to the point in space $\underline{b}$ through the following relation:

$$b_i = M_{ij} (b^{body})_j + x_i$$

9 References

Segel L., (1977) **Mathematics Applied to Continuum Mechanics**, Macmillan, New York

Misner, Charles W., Kip S. Thorne, and John Archibald Wheeler, **Gravitation,** W.H. Freeman and Co., San Francisco, 1973.

Appendix A - Derivation of 3D epsilon-delta rule.

To derive the identity,

$$\epsilon_{\boxed{i}jk} \, \epsilon_{\boxed{i}pq} = \delta_{jp} \, \delta_{kq} - \delta_{jq} \, \delta_{kp}$$

first evaluate the expression with independent indices for j, k, p, and q, and then expand the sum over the bound index i, letting i equal $\{1, 2, 3\}$, but in the order starting with j, then $j+1$, and $j+2$ (modulo 3, plus 1).

$$
\begin{aligned}
\epsilon_{i[jk]}\ \epsilon_{i[pq]} = & \quad \epsilon_{[jjk]}\ \epsilon_{[jpq]} \\
& + \epsilon_{[j+1][jk]}\ \epsilon_{[j+1][pq]} \\
& + \epsilon_{[j+2][jk]}\ \epsilon_{[j+2][pq]}
\end{aligned}
$$

The first term is zero, due to the repeated index j. There is a nonzero contribution only where $k = j+2$ in the second term, and $k = j+1$ in the third.

$$
\begin{aligned}
= & -1_{[k=j+2]}\ \epsilon_{[j+1]pq} \\
& +1_{[k=j+1]}\ \epsilon_{[j+2]pq}
\end{aligned}
$$

Expanding the two nonzero terms of the permutation symbols:

$$
\begin{aligned}
= & -1_{[k=j+2]}\ (1_{[p=j+2][q=j]} - 1_{[p=j][q=j+2]}) \\
& +1_{[k=j+1]}\ (-1_{[p=j+1][q=j]} + 1_{[p=j][q=j+1]})
\end{aligned}
$$

Expand, add and subtract the same term, then collect into positive and negative terms:

$$
\begin{aligned}
= & -1_{[k=j+2]}\ 1_{[p=j+2][q=j]} \\
& +1_{[k=j+2]}\ 1_{[p=j][q=j+2]} \\
& -1_{[k=j+1]}\ 1_{[p=j+1][q=j]} \\
& +1_{[k=j+1]}\ 1_{[p=j][q=j+1]}) \\
& +1_{[k=j]}\ 1_{[p=j][q=j]}) \\
& -1_{[k=j]}\ 1_{[p=j][q=j]}) \\
= & \Big(-1_{[k=j]}\ 1_{[p=j]} - 1_{[k=j+1]}\ 1_{[p=j+1]} \\
& \quad - 1_{[k=j+2]}\ 1_{[p=j+2]}\Big)\ 1_{[q=j]} \\
& + \Big(1_{[k=j]}\ 1_{[q=j]} + 1_{[k=j+1]}\ 1_{[q=j+1]} \\
& + \quad 1_{[k=j+2]}\ 1_{[q=j+2]}\Big)\ 1_{[p=j]}
\end{aligned}
$$

Since j takes on only three values; the first parenthetic expressions is 1 if k equals p, and the other is 1 if k equals q.

$$
\begin{aligned}
= & \ 1_{[p=j]}\, 1_{[q=k]} - 1_{[q=j]}\, 1_{[p=k]} \\
= & \ \delta_{pj}\, \delta_{qk} - \delta_{qj}\, \delta_{pk}
\end{aligned}
$$