UNM BOOK DRAFT

September 4, 2003 5:21 pm

Functional and Structured Tensor Analysis for Engineers

A casual (intuition-based) introduction to vector and tensor analysis with reviews of popular notations used in contemporary materials modeling

R. M. Brannon

~~University of New Mexico, Albuquerque~~

Contact author at ~~rmbrann@sandia.gov~~

NOTE: When using Adobe's "acrobat reader" to view this document, the page numbers in acrobat will not coincide with the page numbers shown at the bottom of each page of this document.

Note to draft readers: The most useful textbooks are the ones with fantastic indexes. The book's index is rather new and still under construction.

It would really help if you all could send me a note whenever you discover that an important entry is missing from this index. I'll be sure to add it.

This work is a community effort. Let's try to make this document helpful to others.

FUNCTIONAL AND STRUCTURED TENSOR ANALYSIS FOR ENGINEERS

A casual (intuition-based) introduction to vector and tensor analysis with reviews of popular notations used in contemporary materials modeling

Rebecca M. Brannon[†]
[†]~~University of New Mexico Adjunct professor~~
~~rmbrann@sandia.gov~~

Abstract

Elementary vector and tensor analysis concepts are reviewed in a manner that proves useful for higher-order tensor analysis of anisotropic media. In addition to reviewing basic matrix and vector analysis, the concept of a tensor is covered by reviewing and contrasting numerous *different* definition one might see in the literature for the term “tensor.” Basic vector and tensor operations are provided, as well as some lesser-known operations that are useful in materials modeling. Considerable space is devoted to “philosophical” discussions about relative merits of the many (often conflicting) tensor notation systems in popular use.

Acknowledgments

An indeterminately large (but, of course, countable) set of people who have offered advice, encouragement, and fantastic suggestions throughout the years that I've spent writing this document. I say *years* because the seeds for this document were sown back in 1986, when I was a co-op student at Los Alamos National Laboratories, and I made the mistake of asking my supervisor, Norm Johnson, "what's a tensor?" His reply? "read the appendix of R.B. "Bob" Bird's book, *Dynamics of Polymeric Liquids*. I did — and got hooked. Bird's appendix (which has nothing to do with polymers) is an outstanding and succinct summary of vector and tensor analysis. Reading it motivated me, as an undergraduate, to take my first graduate level continuum mechanics class from Dr. H.L. "Buck" Schreyer at the University of New Mexico. Buck Schreyer used multiple underlines beneath symbols as a teaching aid to help his students keep track of the different kinds of strange new objects (tensors) appearing in his lectures, and I have adopted his notation in this document. Later taking Buck's beginning and advanced finite element classes further improved my command of matrix analysis and partial differential equations. Buck's teaching pace was fast, so we all struggled to keep up. Buck was careful to explain that he would often cover esoteric subjects principally to enable us to effectively read the literature, though sometimes merely to give us a different perspective on what we had already learned. Buck armed us with a slew of neat tricks or fascinating insights that were rarely seen in any publications. I often found myself "secretly" using Buck's tips in my own work, and then struggling to figure out how to explain how I was able to come up with these "miracle instant answers" — the effort to reproduce my results using conventional (better known) techniques helped me learn better how to communicate difficult concepts to a broader audience. While taking Buck's continuum mechanics course, I simultaneously learned variational mechanics from Fred Ju (also at UNM), which was fortunate timing because Dr. Ju's refreshing and careful teaching style forced me to make enlightening connections between his class and Schreyer's class. Taking thermodynamics from A. Razanni (UNM) helped me improve my understanding of partial derivatives and their applications (furthermore, my interactions with Buck Schreyer helped me figure out how gas thermodynamics equations generalized to the solid mechanics arena). Following my undergraduate experiences at UNM, I was fortunate to learn advanced applications of continuum mechanics from my Ph.D advisor, Prof. Walt Drugan (U. Wisconsin), who introduced me to even more (often completely new) viewpoints to add to my tensor analysis toolbelt. While at Wisconsin, I took an elasticity course from Prof. Chen, who was enamoured of doing all proofs entirely in curvilinear notation, so I was forced to improve my abilities in this area (curvilinear analysis is not covered in this book, but it may be found in a separate publication, Ref. [6]. A slightly different spin on curvilinear analysis came when I took Arthur Lodge's "Elastic Liquids" class. My *third* continuum mechanics course, this time taught by Millard Johnson (U. Wisc), introduced me to the usefulness of "Rossetta stone" type derivations of classic theorems, done using multiple notations to make them clear to every reader. It was here where I conceded that no single notation is superior, and I had better become darn good at them *all*. At Wisconsin, I took a class on Greens functions and boundary value problems from the noted mathematician R. Dickey, who really drove home the importance of projection operations in physical applications, and instilled in me the irresistible habit of examining operators for their properties and classifying them as outlined in our class textbook [12]; it was Dickey who finally got me into the habit of looking for analogies between seemingly unrelated operators and sets so that my strong knowledge. Dickey himself got sideswiped by this habit when I solved one of his exam questions by doing it using a technique that I had learned in Buck Schreyer's continuum mechanics class and which I realized would also work on the exam question by merely re-interpreting the vector dot product as the inner product that applies for continuous functions. As I walked into my Ph.D. defense, I warned Dickey (who was on my committee) that my thesis was really just a giant application of the projection theorem, and he replied "most are, but you are distinguished by *recognizing* the fact!" Even though neither this book nor very many of my other publications (aside from Ref. [6], of course) employ curvilinear notation, my exposure to it has been invaluable to lend insight to the relationship between so-called "convected coordinates" and "unconvected reference spaces" often used in materials modeling. Having gotten my first exposure to tensor analysis from reading Bird's polymer book, I naturally felt compelled to take his macromolecular fluid dynamics course at U. Wisc, which solidified several concepts further. Bird's course was immediately followed by an applied analysis course, taught by ____, where more correct "mathematician's" viewpoints on tensor analysis were drilled into me (the textbook for this course [17] is outstanding, and don't be swayed by the fact that "chemical engineering" is part of its title — the book applies to *any* field of physics). These and numerous other academic mentors I've had throughout my career have given me a wonderfully balanced set of analysis tools, and I wish I could thank them enough.

For the longest time, this "Acknowledgement" section said only "*Acknowledgements to be added. Stay tuned...*" Assigning such low priority to the acknowledgements section was a gross tactical error on my part. When my colleagues offered assistance and suggestions in the earliest days of error-ridden *rough* drafts of this book, I thought to myself "I should thank them in my acknowledgements section." A few years later, I sit here trying to recall the droves of early reviewers. I remember contributions from Glenn Randers-Pherson because his advice for one of my other publications proved to be incredibly helpful, and he did the same for this more elementary document as well. A few folks (Mark Christen, Allen Robinson, Stewart Silling, Paul Taylor, Tim Trucano) in my former department at Sandia National Labs also came forward with suggestions or helpful discussions that were incorporated into this book. While in my new department at Sandia National Laboratories, I continued to gain new insight, especially from Dan Segalman and Bill Scherzinger.

Part of what has driven me to continue to improve this document has been the numerous encouraging remarks (approximately one per week) that I have received from researchers and students all over the world who have stumbled upon the pdf draft version of this document that I originally wrote as a student's guide when I taught Continuum Mechanics at UNM. I don't recall the names of people who sent me encouraging words in the early days, but some recent folks are Ricardo Colorado, Vince Owens, Dave Doolinand Mr. Jan Cox. Jan was especially inspiring because he was so enthusiastic about this work that he spent an entire afternoon disscussing it with me after a business trip I made to his home city, Oakland CA. Even some professors [such as Lynn Bennethum (U. Colorado), Ron Smelser (U. Idaho), Tom Scarpas (TU Delft), Sanjay Arwad (JHU), Kaspar William (U. Colorado), Walt Gerstle (U. New Mexico)] have told me that they have directed their own students to the web version of this document as supplemental reading.

In Sept. 2002, Bob Cain sent me an email asking about printing issues of the web draft; his email signature had the Einstein quote that you now see heading Chapter 1 of this document. After getting his permission to also use that quote in my own document, I was inspired to begin every chapter with an ice-breaker quote from my personal collection.

I still need to recognize the many folks who have sent helpful emails over the last year. Stay tuned.

Contents

Figures

Preface

Math and science journals often have extremely restrictive page limits, making it virtually impossible to present a coherent development of complicated concepts by working upward from basic concepts. Furthermore, scholarly journals are intended for the presentation of *new* results, so detailed explanations of known results are generally frowned upon (even if those results are not well-known or well-understood). Consequently, only those readers who are *already* well-versed in a subject have any hope of effectively reading the literature to further expand their knowledge. While this situation is good for experienced researchers and specialists in a particular field of study, it can be a frustrating handicap for less experienced people or people whose expertise lies elsewhere. This book serves these individuals by presenting several known theorems or mathematical techniques that are useful for the analysis material behavior. Most of these theorems are scattered willy-nilly throughout the literature. Several rarely appear in elementary textbooks. Most of the results in this book can be found in advanced textbooks on functional analysis, but these books tend to be overly generalized, so the application to specific problems is unclear. Advanced mathematics books also tend to use notation that might be unfamiliar to the typical research engineer. This book presents derivations of theorems only where they help clarify concepts. The range of applicability of theorems is also omitted in certain situations. For example, describing the applicability range of a Taylor series expansion requires the use of complex variables, which is beyond the scope of this document. Likewise, unless otherwise stated, I will always implicitly presume that functions are "well-behaved" enough to permit whatever operations I perform. For example, the act of writing df/dx will implicitly tell you that I am assuming that f can be written as a function of x and (furthermore) this function is differentiable. In the sense that much of the usual (but distracting) mathematical provisos are missing, I consider this document to be a work of engineering despite the fact that it is concerned principally with mathematics. While I hope this book will be useful to a broader audience of readers, my personal motivation is to establish a single bibliographic reference to which I can point from my more stilted and terse journal publications.

~~Rebecca Brannon,~~ ~~rmbrann@sandia.gov~~
~~Sandia National Laboratories~~
~~July 11, 2003 1:03 pm.~~

"It is important that students bring a certain ragamuffin, barefoot, irreverence to their studies; they are not here to worship what is known, but to question it"

— J. Bronowski [The Ascent of Man]

FUNCTIONAL AND STRUCTURED TENSOR ANALYSIS FOR ENGINEERS:

a casual (intuition-based) introduction to vector and tensor analysis with reviews of popular notations used in contemporary materials modeling

> *"Things should be described as simply as possible, but no simpler."* — *A. Einstein*

1. Introduction

RECOMMENDATION: To get immediately into tensor analysis "meat and potatoes" go now to page 21. If, at any time, you become curious about what has motivated our style of presentation, then consider coming back to this introduction, which just outlines scope and philosophy.

There's no need to read this book in step-by-step progression. Each section is nearly self-contained. If needed, you can backtrack to prerequisite material (e.g., unfamiliar terms) by using the index.

This book reviews tensor algebra and tensor calculus using a notation that proves useful when extending these basic ideas to higher dimensions. Our intended audience comprises students and professionals (especially those in the material modeling community) who have previously learned vector/tensor analysis only at the rudimentary level covered in freshman calculus and physics courses. Here in this book, you will find a presentation of vector and tensor analysis aimed only at "preparing" you to read properly rigorous textbooks. You are *expected* to refer to more classical (rigorous) textbooks to more deeply understand each theorem that we present casually in this book. Some people can readily master the stilted mathematical language of generalized math theory without ever caring about what the equations *mean* in a physical sense — what a shame. Engineers and other "applications-oriented" people often have trouble getting past the supreme generality in classical textbooks (where, for example, numbers are complex and sets have arbitrary or infinite dimensions). To service these people, we will limit attention to ordinary engineering contexts where numbers are *real* and the world is three-dimensional. Newcomers to engineering tensor analysis will also eventually become exasperated by the apparent disconnects between jargon and definitions among practitioners in the field — some professors define the word "tensor" one way while others will define it so dramatically differently that the two definitions don't appear to have anything to do with one another. In this book we will alert you about these terminology conflicts, and provide you with means of converting between notational systems (structures), which are essential skills if you wish to effectively read the literature or to communicate with colleagues.

After presenting basic vector and tensor analysis in the form most useful for ordinary three-dimensional real-valued engineering problems, we will add some layers of complexity that begin to show the path to unified theories without walking too far down it. The idea will be to explain that many theorems in higher-dimensional realms have perfect analogs with the ordinary concepts from 3D. For example, you will learn in this book how to obliquely project a vector onto a plane (i.e, find the "shadow" cast by an arrow when you hold it up in the late afternoon sun), and we demonstrate in other (separate) work that the act of solving viscoplasticity models by a return mapping algorithm is perfectly analogous to vector projection.

Throughout this book, we use the term "ordinary" to refer to the three dimensional physical space in which everyday engineering problems occur. The term "abstract" will be used later when extending *ordinary* concepts to higher dimensional spaces, which is the principal goal of generalized tensor analysis. Except where otherwise stated, the basis $\{\underset{\sim}{e}_1, \underset{\sim}{e}_2, \underset{\sim}{e}_3\}$ used for vectors and tensors in this book will be assumed *regular* (i.e., orthonormal and right-handed). Thus, all indicial formulas in this book use what most people call rectangular Cartesian components. The abbreviation "RCS" is also frequently used to denote "Rectangular Cartesian System." Readers interested in irregular bases can find a discussion of curvilinear coordinates at ~~**http://www.me.unm.edu/~rmbrann/gobag.html**~~ (however, that document presumes that the reader is already familiar with the notation and basic identities that are covered in this book).

STRUCTURES and SUPERSTRUCTURES

If you dislike philosophical discussions, then please skip this section. You may go directly to page 21 without loss.

Tensor analysis arises naturally from the study of linear operators. Though tensor analysis is interesting in its own right, engineers learn it because the operators have some physical significance. Junior high school children learn about zeroth order tensors when they are taught the mathematics of straight lines, and the most important new concept at that time is the *slope* of a line. In freshman calculus, students learn to find *local* slopes (i.e., tangents to curves obtained through differentiation). Freshman students are also given a discomforting introduction to first-order tensors when they are told that a vector is "something with magnitude and direction". For scientists, these concepts begin to "gel" in physics classes (where "useful" vectors such as velocity or electric field are introduced, and vector operations such as the cross-product begin to take on useful meanings). As students progress, eventually their attention focuses on the vector operations themselves. Some vector operations (such as the dot product) start with two vectors to produce a scalar. Other operations (such as the cross product) produce another vector as output. Many fundamental vector operations are linear, and the concept of a tensor emerges as naturally as the concept of *slope* emerged when you took junior high algebra. Other vector operations are nonlinear, but a "tangent *tensor*" can be constructed in the same sense that a tangent to a nonlinear curve can be found by freshman calculus students.

The *functional* or *operational* concept of a tensor deals directly with the physical meaning of the tensor as an operation or a transformation. The "book-keeping" for characterizing the transformation is accomplished through the use of *structures*. A structure is simply a notation or syntax — it is an arrangement of individual constituent "parts" written down on the page following strict "blueprints." For example, a *matrix* is a structure constructed by writing down a collection of numbers in tabular form (usually 3×3, 3×1, or 1×3 arrays for engineering applications). The arrangement of two letters in the form y^x is a structure that *represents* raising y to the power x. In computer programing, the structure "**y^x**" is often used to represent *the same operation*. The notation $\frac{dy}{dx}$ is a *structure* that symbolically represents the *operation* of differentiating y with respect to x, and this operation is sometimes represented using the alternative structure "$y_{,x}$". All of these examples of structures should be familiar to you. Though you probably don't remember it, they were undoubtedly quite strange and foreign when you first saw them. Tensor notation (tensor structures) will probably affect you the same way. To make matters worse, unlike the examples we cited here, tensor notation varies widely among different researchers. One person's tensor notation often dramatically conflicts with notation adopted by another researcher (their notations can't coexist peacefully like y^x and "**y^x**"). Neither researcher has committed an atrocity — they are both within rights to use whatever notation they desire. Don't get into cat fights with others about their notation preferences. People select notation in a way that works best for their application *or for the audience they are trying to reach*. Tensor analysis is such a rich field of study that variants in tensor notation are a fact of life, and attempts to impose uniformity is short-sighted folly. However, you are justified in criticizing another person's notation if they are not self-consistent within a single publication.

The assembly of symbols, $\frac{a}{b}$, is a standard structure for division and rs is a standard structure for multiplication. Being essentially the study of structures, mathematics permits us to construct unambiguous meanings of "superstructures" such as $\frac{ab}{rs}$ and consistency rules (i.e., theorems) such as

$$\frac{ab}{rs} = \frac{b}{s} \text{ if } a = r \tag{1.1}$$

We've already mentioned that the same operation might be denoted by *different* structures (e.g., "**y^x**" means the same thing as y^x). Conversely, it's not unusual for structures to be *overloaded*, which means that an identical arrangement of symbols on the page can have different *meaning* depending on the meanings of the constituent "parts" or depending on context. For example, we mentioned that "$\frac{ab}{rs} = \frac{b}{s}$ if $a = r$", but everyone knows that you shouldn't use the same rule to cancel the "d"s in a derivative $\frac{dy}{dx}$ to claim it equals $\frac{y}{x}$. The derivative is a different structure. It shares *some* manipulation rules with fractions, but not all. Handled carefully, structure overloading can be a powerful tool. If, for example, α and β are numbers and $\boldsymbol{v}$ is a vector, then structure overloading permits us to write $(\alpha+\beta)\boldsymbol{v} = \alpha\boldsymbol{v}+\beta\boldsymbol{v}$. Here, we overloaded the addition symbol "+"; it represents addition of numbers on the left side but addition of vectors on the right. Structure overloading also permits us to assert the heuristically appealing theorem $\frac{dy}{dx}\frac{dx}{dz} = \frac{dy}{dz}$; in this context, the horizontal bar does *not* denote division, so you have to prove this theorem — you can't just "cancel" the "dx"s as if these really were fractions. The power of overloading (making derivatives *look* like fractions) is evident here because of the heuristic *appearance* that they cancel just like regular fractions.

In this book, we use the phrase "tensor structure" for any tensor notation system that is internally(self)-consistent, and which *everywhere* obeys its own rules. Just about any person will claim that his or her tensor notation is a structure, but careful inspection often reveals structure violations. In this book, we will describe one particular tensor notation system that is, we believe, a reliable structure.* Just as other researchers adopt a notation system to best suit their applications, we have adopted our structure because it appears to be ideally suited to generalization to higher-order applications in materials constitutive modeling. Even though we will carefully outline our tensor structure rules, we will also call attention to alternative notations used by other people. Having command of multiple notation systems will position you to most effectively communicate with others. Never (unless you are a professor) force someone else to learn *your* tensor notation preferences — you should speak to others in *their* language if you wish to gain their favor.

We've already seen that different structures are routinely used to represent the same function or operation (e.g. y^x means the same thing as "**y^x**"). Ideally, a structure should be selected to best match the application at hand. If no conventional structure seems to do a good job, then you should feel free to invent your own structures or superstructures. However, structures must always come equipped with unambiguous rules for definition, assembly, manipulation, and interpretation. Furthermore, structures should obey certain "good citizenship" provisos.

> *(i)* If other people use different notations from your own, then you should clearly provide an explanation of the meaning of *your* structures. For example, in tensor analysis, the structure

* Readers who find a breakdown in our structure are encouraged to notify us.

$\boldsymbol{A}:\boldsymbol{B}$ often has different meanings, depending on who writes it down; hence, if you use this structure, then you should always define what you mean by it.

(ii) Notation should not grossly violate commonly adopted "standards." By "standards," we are referring to those everyday bread-and-butter structures that come *implicitly* endowed with certain definitions and manipulation rules. For example, "$x + y$" had darned will better stand for addition — only a deranged person would declare that the structure "$x + y$" means division of x by y (something that the rest of us would denote by $\frac{x}{y}$, x / y, $x \div y$ or even $y \overline{) x}$). Similarly, the words you use to describe your structures should not conflict with universally recognized lexicon of mathematics. (see, for example, our discussion of the phrase "inner product.")

(iii) Within a single publication, notation should be applied consistently. In the continuum mechanics literature, it is not uncommon for the structure $\nabla \boldsymbol{v}$ (called the gradient of a vector) to be defined in the nomenclature section in terms of a matrix whose ij components are $\partial v_j / \partial x_i$. Unfortunately, however, within the same publication, some inattentive authors later denote the "velocity gradient" by $\nabla \boldsymbol{v}$ but with components $\partial v_i / \partial x_j$ — that's a structure self-consistency violation!

(iv) Exceptions to structure definitions are sometimes unavoidable, but the exception should always be made clear to the reader. For example, in this book, we will define some implicit summation rules that permit the reader to know that certain things are being summed without a summation sign present. There are times, however, that the summation rules must be suspended and structure consistency demands that these instances must be carefully called out.

What is a scalar? What is a vector?

This physical introduction may be skipped. You may go directly to page 21 without loss.

We will frequently exploit our assumption that you have some familiarity with vector analysis. You are expected to have a vague notion that a "scalar" is something that has magnitude, but no direction; examples include temperature, density, time, etc. At the very least, you presumably know the sloppy definition that a vector is "something with length and direction." Examples include velocity, force, and electric field. You are further presumed to know that an ordinary engineering vector can be described in terms of three components referenced to three unit base vectors. A prime goal of this book is to improve this baseline "undergraduate's" understanding of scalars and vectors.

In this book, scalars are typeset in plain italic ($a, b, c, \ldots$). Vectors are typeset in bold with a single under-tilde (for example, $\underset{\sim}{\boldsymbol{v}}$), and their components are referred to by numbered subscripts (v_1, v_2, v_3). Introductory calculus courses usually denote the orthonormal Cartesian base vectors by $\{\vec{\boldsymbol{i}}, \vec{\boldsymbol{j}}, \vec{\boldsymbol{k}}\}$, but why give up so many letters of the alphabet? We will use numerically subscripted symbols such as $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ or $\{\underset{\sim}{\boldsymbol{E}}_1, \underset{\sim}{\boldsymbol{E}}_2, \underset{\sim}{\boldsymbol{E}}_3\}$ to denote the orthonormal base vectors.

As this book progresses, we will improve and refine our terminology to ultimately provide the mathematician's definition of the word "vector." This rigorous (and therefore abstract) definition is based on testing the properties of a candidate set of objects for certain behaviors under proposed definitions for addition and scalar multiplication. Many engineering textbooks define a vector according to how the components change upon a change of basis. This component transformation viewpoint is related to the more general mathematician's definition of "vector" because it is a specific instance of a *discerning definition* of membership in what the mathematician would see as a candidate set of "objects." For many people, the mathematician's definition of the word "vector" sparks an epiphany where it is seen that a lot of things in math and in nature function just like ordinary (engineering) vectors. Learning about one set of objects can provide valuable insight into a new and unrelated set of objects if it can be shown that both sets are vector spaces in the abstract mathematician's sense.

What is a tensor?

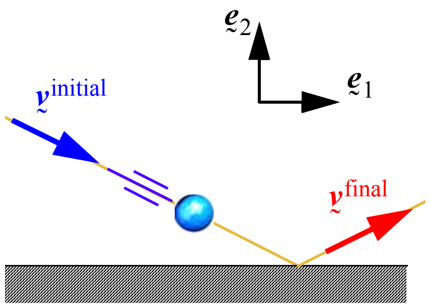
This section may be skipped. You may go directly to page 21 without loss.

In this book we will assume you have virtually *zero* pre-existing knowledge of tensors. Nonetheless, it will be occasionally convenient to talk about tensor concepts prior to carefully defining the word "tensor," so we need to give you a vague notion about what they are. Tensors arise when dealing with functions that take a vector as input and produce a vector as output. For example, if a ball is thrown at the ground with a certain velocity (which is a vector), then classical physics principals can be use to come up with a formula for the velocity vector *after* hitting the ground. In other words, there is presumably a function that takes the initial velocity vector as input and produces the final velocity vector as output: $\underset{\sim}{\boldsymbol{v}}^{\text{final}} = f(\underset{\sim}{\boldsymbol{v}}^{\text{initial}})$. When grade school kids learn about scalar functions ($y = f(x)$), they first learn about straight lines. Later on, as college freshman, they learn the brilliant principle upon which calculus is based: namely, nonlinear functions can be regarded as a collection of *infinitesimal* straight line segments. Consequently, the study of straight lines forms an essential foundation upon which to study the nonlinear functions that appear in nature. Like scalar functions, *vector-to-vector* functions might be linear or non-linear. Very loosely speaking, a vector-to-vector transformation $\underset{\sim}{\boldsymbol{y}} = f(\underset{\sim}{\boldsymbol{x}})$ is linear if the components of the output vector $\underset{\sim}{\boldsymbol{y}}$ can be computed by a square 3×3 matrix $[m]$ acting on the input vector $\underset{\sim}{\boldsymbol{x}}$:*

[*] If you are not familiar with how to multiply a 3×3 matrix times a 3×1 array, see page 22.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{1.2}$$

Consider, for example, our function that relates the pre-impact velocity to the post-impact velocity for a ball bouncing off a surface. Suppose the surface is frictionless and the ball is perfectly elastic. If the normal to the surface points in the 2-direction, then the second component of velocity will change sign while the other components will remain unchanged. This relationship can be written in the form of Eq. (1.2) as


$\underset{\sim}{e}_2$
$\underset{\sim}{e}_1$
$\underset{\sim}{v}^{\text{initial}}$
$\underset{\sim}{v}^{\text{final}}$

$$\begin{bmatrix} v_1^{\text{final}} \\ v_2^{\text{final}} \\ v_3^{\text{final}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1^{\text{initial}} \\ v_2^{\text{initial}} \\ v_3^{\text{initial}} \end{bmatrix} \tag{1.3}$$

The matrix $[M]$ in Eq. (1.2) plays a role similar to the role played by the slope m in the most rudimentary equation for a scalar straight line, $y = mx$.* For any linear vector-to-vector transformation, $\underset{\sim}{y} = f(\underset{\sim}{x})$, there always exists a second-order tensor [which we will typeset in bold with two under-tildes, $\underset{\approx}{M}$] that completely characterizes the transformation.† We will later explain that a tensor $\underset{\approx}{M}$ always has an associated 3×3 matrix of components. Whenever we write an equation of the form

$$\underset{\sim}{y} = \underset{\approx}{M} \bullet \underset{\sim}{x}, \tag{1.4}$$

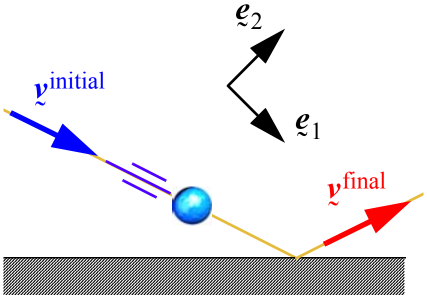
it should be regarded as a symbolic (more compact) expression equivalent to Eq. (1.2). As will be discussed in great detail later, a tensor is more than just a matrix. Just as the components of a vector change when a different basis is used, the components of the 3×3 matrix that characterizes a tensor will also change when the underlying basis changes. Conversely, if a given 3×3 matrix fails to transform in the necessary way upon a change of basis, then that matrix must *not* correspond to a tensor. For example, let's consider again the bouncing ball model, but this time, we will set up the basis differently. If we had declared that the normal to the surface pointed in the 3-direction instead of the 2-direction, then Eq. (1.3) would have ended up being

* Incidentally, the operation $y = mx + b$ is *not* linear. The proper term is "affine." Note that $y - b = mx$. Thus, by studying linear functions, you are only a step away from affine functions (just add the constant term after doing the linear part of the analysis).

† Existence of the tensor is ensured by the Representation Theorem, covered later in Eq. 9.7.

$$\begin{bmatrix} v_1^{\text{final}} \\ v_2^{\text{final}} \\ v_3^{\text{final}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v_1^{\text{initial}} \\ v_2^{\text{initial}} \\ v_3^{\text{initial}} \end{bmatrix} \tag{1.5}$$

Note that changing the basis forced a change in the $[M]$ matrix. Less trivially, if we had set up the basis by rotating it $45°$ clockwise, then the formula would have been given by the far less intuitive or obvious relationship


$\underset{\sim}{e}_2$
$\underset{\sim}{v}^{\text{initial}}$
$\underset{\sim}{e}_1$
$\underset{\sim}{v}^{\text{final}}$

$$\begin{bmatrix} v_1^{\text{final}} \\ v_2^{\text{final}} \\ v_3^{\text{final}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1^{\text{initial}} \\ v_2^{\text{initial}} \\ v_3^{\text{initial}} \end{bmatrix} \tag{1.6}$$

We have not yet covered the formal process for determining how the components of the tensor $\underset{\approx}{M}$ must vary with a change in basis, so don't be dissuaded if you don't know how we came up with the components shown in Eq. (1.6). One thing you can do at this stage is double-check the equation for some special cases where you know what the answer should be. For example, with this rotated basis, if the ball has an incoming trajectory that happens to be parallel to $\underset{\sim}{e}_1$, then examining the picture should tell you that the outgoing trajectory should be parallel to $\underset{\sim}{e}_2$, and the above matrix equation does indeed predict this result. Another special case you can consider is when the incoming trajectory is headed straight down toward the surface so that $\underset{\sim}{v}^{\text{initial}}$ is parallel to $\underset{\sim}{e}_1 - \underset{\sim}{e}_2$, which corresponds to a component array $\{1, -1, 0\}$. Then the matrix operation of Eq. (1.6) would give

$$\begin{bmatrix} v_1^{\text{final}} \\ v_2^{\text{final}} \\ v_3^{\text{final}} \end{bmatrix} \text{ is parallel to } \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \text{ or } \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \tag{1.7}$$

This means the outgoing final velocity is parallel to $\underset{\sim}{e}_2 - \underset{\sim}{e}_1$, which (referring to the sketch) is straight up away from the surface, as expected. The key point here is: if you know the component matrix for a tensor with respect to one basis, then there exists a formal procedure (discussed later in this book) that will tell you what the component matrix must look like with respect to a different basis.

At this point, we have provided only an extremely vague and undoubtedly disquieting notion of the meaning of the word "tensor." The sophistication and correctness of this preliminary definition is on a par with the definition of a vector as "something with length and direction." A tensor is the next step in complexity — it is a mathematical abstraction or book-keeping tool that characterizes how something with length and direction transforms into *something else* with length and direction. It plays a role in vector analysis similar to the concept of slope in algebra.

Examples of tensors in materials mechanics.

This section may be skipped. You may go directly to page 21 without loss.

The stress tensor. In materials modeling, the "stress tensor" plays a pivotal role. If a blob of material is subjected to loads (point forces, body forces, distributed pressures, etc.) then it generally reacts with some sort of internal resistance to these loads (viscous, inertial, elastic, etc.). As a "thought experiment", imagine that you could pass a plane through the blob (see Fig. 1.1). To keep the remaining half-blob in the same shape it was in before you sliced it, you would need to approximate the effect of the removed piece by imposing a **traction** (i.e., force per unit area) applied on the cutting plane.

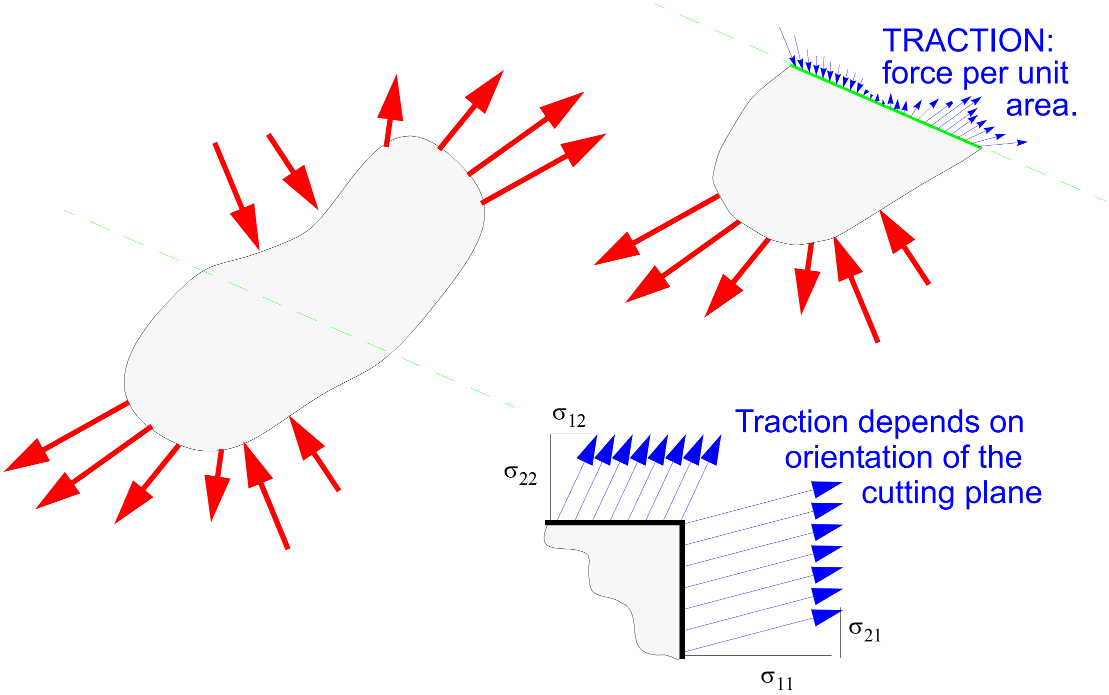


Figure 1.1. *The concept of traction.* When a body is conceptually split in half by a planar surface, the effect of one part of the body on the other is approximated by a "traction", or force per unit area, applied on the cutting plane. Traction is an excellent mathematical model for macroscale bodies (i.e., bodies containing so many atom or molecules that they may be treated as continuous). Different planes will generally have different traction vectors.

Force is a vector, so traction (which is just force per unit area) must be a vector too. Intuitively, you can probably guess that the traction vector needs to have different values at different locations on the cutting plane, so traction naturally is a function of the position vector $\underset{\sim}{x}$. The traction at a particular location $\underset{\sim}{x}$ also depends on the orientation of the cutting plane. If you pass a differently oriented plane through the *same point* $\underset{\sim}{x}$ in a body, then the traction vector at that point will be different. In other words, traction depends on both the location in the body and the orientation of the cutting plane. Stated mathematically, the traction vector $\underset{\sim}{t}$ at a particular position $\underset{\sim}{x}$ varies as a function of the plane's outward unit normal $\underset{\sim}{n}$. This is a vector-to-vector transformation! In this case, we have one vector (traction) that depends on two vectors, $\underset{\sim}{x}$ and $\underset{\sim}{n}$. Whenever attempting to understand a function of two variables, it is always a good idea to consider variation of each variable separately, observing how the function behaves when only one variable changes while the other is held constant. Presumably, *at a given location* $\underset{\sim}{\boldsymbol{x}}$, a functional relationship exists between the plane's orientation $\underset{\sim}{\boldsymbol{n}}$ and the traction vector $\underset{\sim}{\boldsymbol{t}}$. Using the continuum mechanics version of the famous $F = ma$ dynamics equation, Cauchy proved that this relationship between traction and the plane orientation must be *linear*. Whenever you discover that a relationship is linear, you can call upon a central concept of tensor analysis* to immediately state that it is expressible in the form of Eq. (1.2). In other words, there must exist a *tensor*, which we will denote $\underset{\approx}{\sigma}$ and refer to as "**stress**," such that

$$\underset{\sim}{\boldsymbol{t}} = \underset{\approx}{\sigma} \bullet \underset{\sim}{\boldsymbol{n}} \qquad (1.8)$$

Remember that this conclusion resulted from considering variation of $\underset{\sim}{\boldsymbol{n}}$ while holding $\underset{\sim}{\boldsymbol{x}}$ fixed. The dependence of traction on $\underset{\sim}{\boldsymbol{x}}$ might still be nonlinear, but it is a truly monumental discovery that the dependence on $\underset{\sim}{\boldsymbol{n}}$ is so beautifully simple. Written out, showing the independent variables explicitly,

$$\underset{\sim}{\boldsymbol{t}}(\underset{\sim}{\boldsymbol{x}}, \underset{\sim}{\boldsymbol{n}}) = \underset{\approx}{\sigma}(\underset{\sim}{\boldsymbol{x}}) \bullet \underset{\sim}{\boldsymbol{n}} \qquad (1.9)$$

This means the stress tensor itself varies through space (generally in a nonlinear manner), but the dependence on the cutting plane's normal $\underset{\sim}{\boldsymbol{n}}$ is *linear*. As suggested in Fig. 1.1, the components of the stress tensor can be found if the traction is known on the faces of the cube whose faces are aligned with the coordinate directions. Specifically, the j^{th} column of the component matrix $[\underset{\approx}{\sigma}]$ contains the traction vector acting on the j^{th} face of the cube. These "stress elements" don't really have finite spatial extent — they are *infinitesimal* cubes and the tractions acting on each face really represent the traction vectors acting on the three coordinate planes that pass through the *same point* in the body.

* Namely, the Representation Theorem covered later in Eq. 9.7.

The deformation gradient tensor. The stress tensor characterizes the local orientation-dependent loads (force per area) experienced by a body. A different tensor — the "deformation gradient" — characterizes the local volume changes, local orientation changes, and local shape changes associated with deformation. If you paint an infinitesimal square onto the surface of a blob of putty, then the square will deform into a parallelogram (Fig. 1.2).

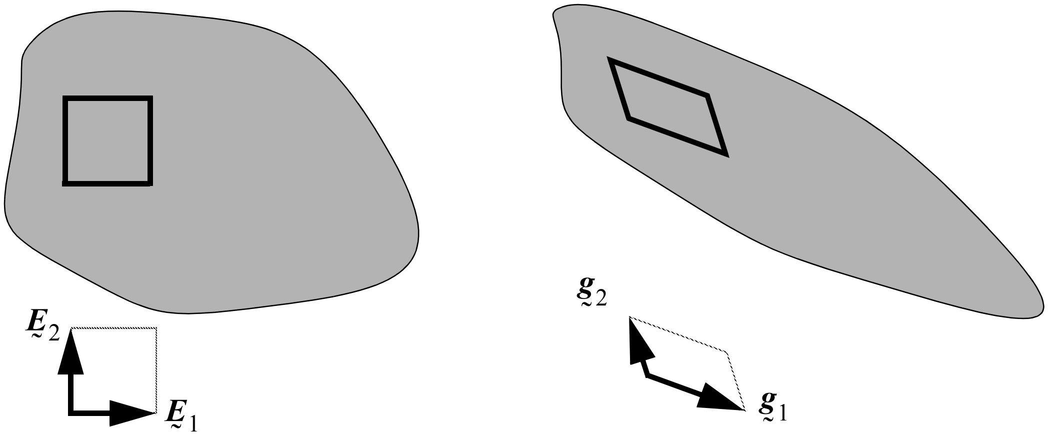

$\underset{\sim}{\mathbf{E}}_2$
$\underset{\sim}{\mathbf{E}}_1$
$\underset{\sim}{\mathbf{g}}_2$
$\underset{\sim}{\mathbf{g}}_1$

Figure 1.2. Stretching silly putty. The square flows with the material to become a parallelogram. Below each figure, is shown how the square and parallelogram can be described by two vectors.

The unit* base vectors $\{\underset{\sim}{\mathbf{E}}_1, \underset{\sim}{\mathbf{E}}_2\}$ forming the edges of the *initial* square, will stretch and rotate to become new vectors, $\{\underset{\sim}{\mathbf{g}}_1, \underset{\sim}{\mathbf{g}}_2\}$, forming the edges of the deformed parallelogram. These ideas can be extended into 3D if one pretends that a *cube* could be "painted" inside the putty. The three unit vectors forming the edges of the initial cube deform into three stretched and rotated vectors forming the edges of the deformed parallelepiped. Assembling the three $\underset{\sim}{\mathbf{g}}_i$ vectors into columns of a 3×3 matrix will give you the matrix of the deformation gradient tensor. Of course, this is only a qualitative description of the deformation gradient tensor. A more classical (and quantified) definition of the deformation gradient tensor starts with the assertion that each point $\underset{\sim}{\mathbf{x}}$ in the currently deformed body must have come from some unique initial location $\underset{\sim}{\mathbf{X}}$ in the initial undeformed reference configuration, you can therefore claim that a *mapping function* $\underset{\sim}{\mathbf{x}} = f(\underset{\sim}{\mathbf{X}})$ must exist. This is a vector-to-vector transformation, but it is generally *not* linear. Recall that tensors characterize *linear* functions that transform vectors to vectors. However, just as a nonlinear algebraic function (e.g., a parabola or a cosine curve or any other nonlinear function) can be viewed as approximately linear in the limit of infinitesimal portions (the local slope of the straight tangent line is determined by differentiating the function), the deformation mapping is linear when expressed in terms of infinitesimal material line segments $d\underset{\sim}{\mathbf{X}}$ and $d\underset{\sim}{\mathbf{x}}$. Specifically, if $\underset{\sim}{\mathbf{x}} = f(\underset{\sim}{\mathbf{X}})$, then the deformation gradient tensor $\underset{\approx}{\mathbf{F}}$ is defined so that $d\underset{\sim}{\mathbf{x}} = \underset{\approx}{\mathbf{F}} \bullet d\underset{\sim}{\mathbf{X}}$. Not surprisingly, the Cartesian component matrix for $\underset{\approx}{\mathbf{F}}$ is given by

[*] Making the infinitesimal square into a *unit* square is merely a matter of choosing a length unit appropriately. All that really matters here is the *ratio* of deformed lengths to initial lengths.

$F_{ij} = \partial x_i / \partial X_j$. While this might be the mathematical formula you will need to use to actually compute the deformation gradient, it is extremely useful to truly understand the basic physical meaning of the tensor too (i.e., how it shows how squares deform to parallelepipeds). All that is needed to determine the components of this (or any) tensor is knowledge of how that transformation changes any three linearly independent vectors.

Vector and Tensor notation — philosophy

This section may be skipped. You may go directly to page 21 without loss.

Tensor notation unfortunately remains non-standardized, so it's important to at least scan any author's tensor notation section to become familiar with his or her definitions and overall approach to the subject. Authors generally select a vector and tensor notation that is well suited for the physical problem of interest to them. In general, no single notation should be considered superior to another.

Our tensor analysis notational preferences are motivated to simplify our other (more complicated and contemporary) applications in materials modeling. Different technical applications frequently call for different notational conventions. The unfortunate consequence is that it often takes many years to master tensor analysis simply because of the numerous (often conflicting) notations currently used in the literature. Table 1.1, for example, shows a sampling of how our notation might differ from other books you might read about tensor analysis. This table employs some conventions (such as implicit indicial notation) that we have not yet defined, so don't worry that some entries are unclear. The only point of this table is to emphasize that you must not presume that the notation you learn in this book will necessarily jibe with the notation you encounter elsewhere. Note, for example, that our notation $\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}$ is completely different from what other people might intend when they write $\boldsymbol{A} \bullet \boldsymbol{B}$. As a teaching tool, we indicate tensor order (also called rank, to be defined soon) by the number of "under-tildes" placed under a symbol. You won't see this done in most books, where tensors and vectors are typically typeset in bold and it is up to you to keep track of their tensor order.

Table 1.1: Some conflicting notations

Operation	Cartesian Indicial Notation	Our Notation	Other Notations
Linear transformation of a vector $\underset{\sim}{\boldsymbol{x}}$ into a new vector $\underset{\sim}{\boldsymbol{v}}$	$v_i = F_{ij}x_j$	$\underset{\sim}{\boldsymbol{v}} = \underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}}$	$\boldsymbol{v} = \boldsymbol{Fx}$
Composition of two tensors $\underset{\approx}{\boldsymbol{A}}$ and $\underset{\approx}{\boldsymbol{B}}$	$C_{ij} = A_{ik}B_{kj}$	$\underset{\approx}{\boldsymbol{C}} = \underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}$	$\boldsymbol{C} = \boldsymbol{AB}$
Inner product of two tensors $\underset{\approx}{\boldsymbol{A}}$ and $\underset{\approx}{\boldsymbol{B}}$	$s = A_{ij}B_{ij}$	$s = \underset{\approx}{\boldsymbol{A}} : \underset{\approx}{\boldsymbol{B}}$	$s = \boldsymbol{A} \bullet \boldsymbol{B}$
Dot product of a vector $\underset{\sim}{\boldsymbol{w}}$ into a linear transformation	$s = w_i F_{ij} x_j$	$\underset{\sim}{\boldsymbol{w}} \bullet \underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}}$	$s = \boldsymbol{w} \bullet \boldsymbol{Fx}$

In this book, we will attempt to cover the most popular tensor analysis notations. One important notation system *not* covered in this book is the one used with general curvilinear coordinates. You can recognize (or suspect) that a person is using general curvilinear notation if they write formulas with indices positioned as both subscripts and superscripts (for example, where we would write $v_i = F_{ij} x_j$ in Cartesian notation, a person using curvilinear notation might instead write something like $v^i = F^i{}_j x^j$). When an author is using general curvilinear notation, their calculus formulas will look somewhat similar to the Cartesian calculus formulas we present in this book, but their curvilinear formulas will usually have additional terms involving strange symbols like $\{{k \atop ij}\}$ or Γ^k_{ij} called "Christoffel" symbols. Whenever you run across indicial formulas that involve these symbols or when the author uses a combination of subscripts and superscripts, then you are probably reading an analysis written in general curvilinear notation, which is *not* covered in this book. In this case, you should use this book as a starting point for first learning tensors in Cartesian systems, and then move on to our separate book [6] for generalizations to curvilinear notation. An alternative approach is to "translate" an author's curvilinear equations into equivalent Cartesian equations by changing all superscripts into ordinary subscripts and by setting every Christoffel symbol equal to zero. This translation is permissible only if you are certain that the original analysis applies to a Euclidean space (i.e., to a space where it is *possible* to define a Cartesian coordinate system). If, for example, the author's analysis was presented for the 2D curvilinear surface of a sphere, then it cannot be translated into Cartesian notation because the surface of a sphere is a non-Euclidean space (you can't draw a map of the world on a 2D piece of paper without distorting the countries). On the other hand, if the analysis was presented for ordinary 3D space, and the author merely chose to use a spherical coordinate system, then you *are* permitted to translate the results into Cartesian notation because ordinary 3D space admits the introduction of a Cartesian system.

Any statement we make here in this book that is cast in *direct structured notation* applies equally well to Cartesian and curvilinear systems. Direct structured equations never used components or base vectors. They represent physical operations with meanings quite independent of whatever coordinate or basis you happen to use. For example, when we say that $\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{w}}$ equals the magnitudes of $\underset{\sim}{\mathbf{v}}$ and $\underset{\sim}{\mathbf{w}}$ times the cosine of the angle between them, that interpretation is valid regardless of your coordinate system. However, when we say $\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{w}} = v_1 w_1 + v_2 w_2 + v_3 w_3$, then that statement (because it involves indexed components) holds only for Cartesian systems. The physical operation $\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{w}}$ is computed one way in Cartesian coordinates and another way in curvilinear — the value and *meaning* of the final result is the same for both systems.

> *"Change isn't painful, but resistance to change is."* — *unattributed*

2. Terminology from functional analysis

RECOMMENDATION: Do *not* read this section in extreme detail. Just scan it to get a basic idea of what terms and notation are defined here. Then go into more practical stuff starting on page 21. Everything discussed in this section is listed in the index, so you can come back here to get definitions of unfamiliar jargon as the need arises.

Vector, tensor, and matrix analysis are subsets of a more general area of study called functional analysis. One purpose of this book is to specialize several overly-general results from functional analysis into forms that are the more convenient for "real world" engineering applications where generalized abstract formulas or notations are not only not necessary, but also damned distracting. Functional analysis deals with operators and their properties. For our purposes, an operator may be regarded as a function $f(x)$. If the argument of the function is a vector and if the result of the function is also vector, then the function is usually called a **transformation** because it transforms one vector to become a new vector.

In this book, any *non-underlined* quantity is just an ordinary number (or, using more fancy jargon, scalar[*] or field member). Quantities such as $\underset{\sim}{\mathbf{v}}$ or $\underset{\sim}{\mathbf{a}}$ with a single squiggly underline (tilde) are vectors. Quantities such as $\underset{\approx}{\mathbf{A}}$ or $\underset{\approx}{\mathbf{T}}$ with *two* under-tildes are second-order tensors. In general, the number of under-tildes beneath a symbol indicates to you the order of that tensor (for this reason, scalars are sometimes called zeroth-order tensors and vectors are called first-order tensors). Occasionally, we will want to make statements that apply equally well to tensors of *any* order. In that case, we might use single *straight* underlines. Quantities with single *straight* underlines (e.g., $\underline{x}$ or $\underline{y}$) might represent scalars, vectors, tensors, or other abstract objects. We follow this convention throughout the text; namely, when discussing a concept that applies equally well to a tensor of *any* order (scalar, vector, second-order tensor), then we will use *straight* underlines or, possibly only bold typesetting with no underlines at all.[†] When discussing "objects" of a *particular*

[*] Strictly speaking, the term "scalar" does *not* apply to any old number. A scalar must be a number (such as temperature or density) whose value does not change when you reorient the basis. For example, the magnitude of a vector is a scalar, but any individual component of a vector (whose value *does* depend on the basis) is not a scalar — it is just a number.

order, then we will use “under-tildes”, and the total number of under-tildes will equal the order of the object. The use of under-tildes and underlines is a teaching tool. In journal publications, you will usually see vectors and tensors typeset in bold with no underlines, in which case it will be up to you to keep track of the tensor order of the quantities.

Some basic terminology from functional analysis is defined very loosely below. More mathematically correct definitions will be given later, or can be readily found in the literature [e.g., Refs 33, 28, 29, 30, 31, 12]. Throughout the following list, you are presumed to be dealing with a set of “objects” (scalars, vectors, or perhaps something more exotic) for which scalar multiplication and “object” addition have well-understood meanings that you (or one of your more creative colleagues) have dreamed up. The diminutive single dot “ $\cdot$ ” multiplication symbol represents ordinary multiplication when the arguments are just scalars. Otherwise, it represents the appropriate inner product depending on the arguments (e.g., it’s the vector dot “ $\bullet$ ” product if the arguments are vectors; it’s the tensor double dot “**:**” product — defined later — when the arguments are tensors); a mathematician’s definition of the “inner product” may be found on page 233.

- A “**linear combination**” of two *objects* $\underset{\sim}{x}$ and $\underset{\sim}{y}$ is any *object* $\underset{\sim}{r}$ that can be expressed in the form $\underset{\sim}{r} = \alpha\underset{\sim}{x} + \beta\underset{\sim}{y}$ for some choice of scalars α and β. A “**linear combination**” of three *objects* ($\underset{\sim}{x}$, $\underset{\sim}{y}$, and $\underset{\sim}{z}$) is any *object* $\underset{\sim}{r}$ that can be expressed in the form $\underset{\sim}{r} = \alpha\underset{\sim}{x} + \beta\underset{\sim}{y} + \gamma\underset{\sim}{z}$. Of course, this definition makes sense only if you have an unambiguous understanding of what the *objects* represent. Moreover, you must have a definition for scalar multiplication and addition of the objects. If, for example, the *“objects”* are 1×2 matrices, then scalar multiplication $\alpha\underset{\sim}{x}$ of some matrix $\underset{\sim}{x} = [x_1, x_2]$ would be defined $[\alpha x_1, \alpha x_2]$ and the linear combination $\alpha\underset{\sim}{x} + \beta\underset{\sim}{y}$ would be a 1×2 matrix given by $[\alpha x_1 + \beta y_1, \alpha x_2 + \beta y_2]$.

- A function f is “**linear**” if $f(\alpha\underset{\sim}{x} + \beta\underset{\sim}{y}) = \alpha f(\underset{\sim}{x}) + \beta f(\underset{\sim}{y})$ for all α, β, $\underset{\sim}{x}$, and $\underset{\sim}{y}$. This means that applying the function to a linear combination of objects will give the same result as instead first applying the function to the objects, and *then* computing the linear combination afterward. Linearity is a profoundly useful property. Incidentally, the definition of linearity demands that a linear function must give zero when applied to zero: $f(\underset{\sim}{0}) = \underset{\sim}{0}$. Therefore, the classic formula for a straight line, $y = f(x) = mx + b$, is *not* a linear function unless the line passes through the origin (i.e., unless $b = 0$). Most people (including us) will sloppily use the term “linear” anyway, but the correct term for the straight line function is “affine.”

- A transformation g is “**affine**” if it can be expressed in the form $g(\underset{\sim}{x}) = f(\underset{\sim}{x}) + \underset{\sim}{b}$, where $\underset{\sim}{b}$ is constant and f is a linear function.

- A transformation f is “**self-adjoint**” if $\underset{\sim}{y} \cdot f(\underset{\sim}{x}) = \underset{\sim}{x} \cdot f(\underset{\sim}{y})$. When applied to a linear

† At this point, you are not expected to already know what is meant by the term “tensor,” much less the “order” of a tensor or the meaning of the phrase “inner product.” For now, consider this section to apply to scalars and vectors. Just understand that the concepts reviewed in this section will also apply in more general tensor settings, once learned.

vector-to-vector transformation, the property of self-adjointness will imply that the associated tensor must be symmetric (or "hermitian" if complex vectors are permitted. *This document limits its scope to real vectors except where explicitly noted otherwise, so don't expect comments like this to continue to litter the text. It's your job to remember that many formulas and theorems in this book might or might not generalize to complex vectors.*

- A transformation f is a **projector** if $f(f(\underset{\sim}{x})) = f(\underset{\sim}{x})$. The term "idempotent" is also frequently used. A projector is a function that will keep on returning the same result if it is applied more than once. Projectors that appear in classical Newtonian physics are usually linear, although there are many problems of engineering interest that involve nonlinear projectors -- if one is attuned enough to look for them.

- Any operator f must have a **domain** of admissible values of $\underset{\sim}{x}$ for which $f(\underset{\sim}{x})$ is well-defined. Throughout this book, the domain of a function must be inferred by you so that the function "makes sense." For example, if $f(x) = 1/x$, then you are expected to infer that the domain is the set of *nonzero* x. We aren't going to waste your time by saying it. Furthermore, throughout this book, all scalars, vectors and tensors are assumed to be real unless otherwise stated. Consequently, whenever you see x^2, you may assume the result is non-negative unless you are explicitly told that x might be complex.

- The "**codomain**" of an operator is the set of all y values such that $y = f(x)$. For example, if $f(x) = x^2$, then the codomain is the set of nonnegative numbers,[*] whereas the *range* is the set of reals. The term **range space** will often be used to refer to the range of a *linear* operator.

- A set S is said to be "**closed**" under a some particular operation if application of that operation to a member of S always gives a result that is *itself* a member of S. For example, the set of all symmetric matrices[†] is closed under matrix addition because the sum of two symmetric matrices is itself a symmetric matrix. By contrast, set of all orthogonal matrices is *not* closed under matrix addition because the sum of two orthogonal matrices is not generally itself an orthogonal matrix. Similarly, the set of all unit vectors is not closed under vector addition because the sum of two unit vectors does not result in a unit vector.

- The **null space** of an operator is the set of all $\underset{\sim}{x}$ for which $f(\underset{\sim}{x}) = \underset{\sim}{0}$.

- For each input $\underset{\sim}{x}$, a well-defined **proper** operator f must give a unique output $\underset{\sim}{y} = f(\underset{\sim}{x})$. In other words, a single $\underset{\sim}{x}$ must never correspond to two or more possible values of y. The operator is called **one-to-one** if the reverse situation also holds.

[*] This follows because we have already stated that x is to be presumed real.

[†] Matrices are defined in the next section.

Namely, f is one-to-one if each y in the codomain of f is obtained by a *unique* x such that $\underset{\sim}{y}=f(\underset{\sim}{x})$. For example, the function $f(x) = x^2$ is *not* one-to-one because a single value of y can be obtained by two values of x (e.g., $y=4$ can be obtained by $x=2$ or $x=-2$).

- Given two proper functions $\underset{\sim}{y} = g(t)$ and $\underset{\sim}{x} = h(t)$, you may presume that a **parametric** relationship exists between $\underset{\sim}{y}$ and $\underset{\sim}{x}$, but this relationship (sometimes called an **implicit function**) might not be a proper function at all. Because g and h are proper functions, it is true that each value of the parameter t will correspond to unique values of $\underset{\sim}{y}$ and $\underset{\sim}{x}$. When these values are assembled together into a graph or table over the range of every possible value of t, then the result is called a **phase** diagram or **phase** space. For example, if $y = \sin t$ and $x = \cos t$, then the phase diagram would be a circle in y versus x phase space.
- If a function is one-to-one, then it is invertible. The **inverse** f^{-1} is defined such that $\underset{\sim}{x}=f^{-1}(\underset{\sim}{y})$.
- A set of "*objects*" is **linearly independent** if no member of the set can be written as a linear combination of the other members of the set. If, for example, the *"objects*" are 1×2 matrices, then the three-member set $\{[1, 2], [3, 4], [5, 6]\}$ is *not* linearly independent because the third matrix can be expressed as a linear combination of the first two matrices; namely, $[5, 6] = (-1)[1, 2] + (2)[3, 4]$.
- The **span** of a collection of vectors is the set of all vectors that can be written as a linear combination of the vectors in the collection. For example, the span of the two vectors $\{1, 1, 0\}$ and $\{1, -1, 0\}$ is the set of all vectors expressible in the form $\alpha_1\{1, 1, 0\} + \alpha_2\{1, -1, 0\}$. This set of vectors represents any vector $\{x_1, x_2, x_3\}$ for which $x_3=0$. The starting collection of vectors does not have to be linearly independent in order for the span to be well-defined. Linear spaces are often described by using spans. For example, you might hear someone refer to "the plane spanned by vectors $\underset{\sim}{a}$ and $\underset{\sim}{b}$," which simply means the plane containing $\underset{\sim}{a}$ and $\underset{\sim}{b}$.
- The **dimension** of a set or a space equals the minimum quantity of "numbers" that you would have to specify in order to uniquely identify a member of that set. In practice, the dimension is often determined by counting some nominally sufficient quantity of numbers *and then subtracting the number of independent constraints that those numbers must satisfy*. For example, ordinary engineering vectors are specified by giving three numbers, so they are nominally three dimensional. However, the set of all *unit* vectors is *two-dimensional* because the three components of a unit vector $\underset{\sim}{n}$ must satisfy the one constraint, $n_1^2 + n_2^2 + n_3^2 = 1$. We later find that an engineering "tensor" can be specified in terms of a 3×3 matrix, which has nine components. Therefore engineering "tensor space" is nine-dimensional. On the other hand, the set of all *symmetric* tensors is *six*-dimensional because the nine nominal

components must obey *three* constraints ($T_{12} = T_{21}$, $T_{23} = T_{32}$, and $T_{31} = T_{13}$).

- Note that the set of all unit vectors forms a two-dimensional subset of the 3D space of ordinary engineering vectors. This 2D subset is curvilinear — each unit vector can be regarded as a point on the surface of the unit sphere. Sometimes a subset will be flat. For example, the set of all vectors whose first component is zero (with respect to some *fixed* basis) represents a "flat" space (it is the plane formed by the second and third coordinate axes). The set of all vectors with all three components being equal is geometrically a straight line (pointing in the 111 direction). It is always worthwhile spending a bit of time getting a feel for the geometric shape of subsets. If the shape is "flat" (e.g. a plane or a straight line), then it is called a **linear manifold** (defined better below). Otherwise it is called **curvilinear**. If a surface is curved but could be "unrolled" into a flat surface or into a line, then the surface is called **Euclidean**; qualitatively, a space is Euclidean if it is always possible to set up a coordinate grid covering the space in such a manner that the coordinate grid cells are all equal sized squares or cubes. The surface of a cylinder is both curvilinear and Euclidean. By contrast, the surface of a sphere is curvilinear and non-Euclidean. Mapping a non-Euclidean space to Euclidean space will always involve distortions in shape and/or size. That's why maps of the world are always distorted when printed on two-dimensional sheets of paper.

- If a set is closed under *vector* addition and scalar multiplication (i.e., if every linear combination of set members gives a result that is also in the set), then the set is called a **linear manifold**, or a **linear space**. Otherwise, the set is **curvilinear**. The set of all unit vectors is a curvilinear space because a linear combination of unit vectors does *not* result in a unit vector. Linear manifolds are like planes that pass through the origin, though they might be "**hyperplanes**," which is just a fancy word for a plane of more than just two dimensions. Linear spaces can also be one-dimensional. Any *straight* line that passes through the origin is a linear manifold.

- *Zero* must always be a member of a linear manifold, and this fact is often a great place to start when considering whether or not a set is a linear space. For example, you can assert that the set of unit vectors is *not* a linear space by simply noting that the zero vector is *not* a unit vector.

- A plane that does *not* pass through the origin must *not* be a linear space. We know this simply because such a plane does not contain the zero vector. This kind of plane is called an "affine" space. An "**affine**" space is a set that *would* become a linear space if the origin were to be moved to any single point in the set. For example, the point $(0, b)$ lies on the straight line defined by the equation, $y = mx + b$. If you move the origin from $O = (0, 0)$ to a new location $O^* = (0, b)$, and introduce a change of variables $x^* = x - 0$ and $y^* = y - b$, then the equation for this *same* line described with respect to this *new* origin would become $y^* = mx^*$, which *does* describe a linear space. Stated differently, a set S is **affine** if every member $\underline{\boldsymbol{x}}$ in that set is expressible in the form of a constant vector $\underline{\boldsymbol{d}}$ plus a vector $\underline{\boldsymbol{x}}^*$ that *does* belong to a linear space. Thus, learning about the properties of linear spaces is sufficient to learn most of what you need to know about affine spaces.

- Given an n-dimensional linear space, a subset of members of that space is **basis** if every member of the space can be expressed as a linear combination of members of the subset. A basis always contains exactly as many members as the dimension of the space.

- A "**binary**" operation is simply a function or transformation that has two arguments. For example, $f(x, y) = x^2 \cos y$ is a binary operation.

- A binary operation $f(\underset{\sim}{x}, \underset{\sim}{y})$ is called "**bilinear**" if it is linear with respect to each of its arguments individually; i.e., $f(\alpha_1 \underset{\sim}{x}_1 + \alpha_2 \underset{\sim}{x}_2, \underset{\sim}{y}) = \alpha_1 f(\underset{\sim}{x}_1, \underset{\sim}{y}) + \alpha_2 f(\underset{\sim}{x}_2, \underset{\sim}{y})$ and $f(\underset{\sim}{x}, \beta_1 \underset{\sim}{y}_1 + \beta_2 \underset{\sim}{y}_2) = \beta_1 f(\underset{\sim}{x}, \underset{\sim}{y}_1) + \beta_2 f(\underset{\sim}{x}, \underset{\sim}{y}_2)$. Later on, after we introduce the notion of tensors, we will find that scalar-valued bilinear functions are always expressible in the form $f(\underset{\sim}{\boldsymbol{x}}, \underset{\sim}{\boldsymbol{y}}) = \underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{y}}$, where $\underset{\approx}{\boldsymbol{A}}$ is a constant second-order tensor.

- The notation for an ordinary derivative dy/dx will, in this book, carry with it several implied assumptions. The very act of writing dy/dx tells you that y is expressible *solely* as a function of x and that function is differentiable.

- An "equation" of the form $y = y(x)$ is not an equation at all. This will be our shorthand notation indicating that y is expressible as a function of x.

- The notation for a *partial* derivative $\partial y/\partial x$ tells you that y is expressible as a function of x *and something else*. A partial derivative is *meaningless* unless you know what the "something else" is. Consider, for example, polar coordinates r and θ related to Cartesian coordinates x and y by $x = r\cos\theta$ and $y = r\sin\theta$. Writing $\partial y/\partial r$ is sloppy. You might suspect that this derivative is holding θ constant, but it might be that it was really intended to hold x constant. All partial derivatives in this book will indicate what variable or variables are being held constant by showing them as subscripts. Thus, for example, $(\partial y/\partial r)_\theta$ is completely different from $(\partial y/\partial r)_x$. An exception to this convention exists for derivatives with respect to subscripted quantities. If for example, it is known that z is a function of three variables s_1, s_2, s_3, then $\partial z/\partial s_2$ should be interpreted to mean $(\partial z/\partial s_2)_{s_1, s_3}$.

- An expression $f(x, y)dx + g(x, y)dy$ is called an **exact differential** if there exists a function $u(x, y)$ such that $du = fdx + gdy$. A necessary and sufficient condition for the potential function u to exist is $(\partial f/\partial y)_x = (\partial g/\partial x)_y$. If so, then it must be true that $f(x, y) = (\partial u/\partial x)_y$ and $g(x, y) = (\partial u/\partial y)_x$. You would integrate these equations to determine $u(x, y)$. Keep in mind that the "constant" of integration with respect to x must be a *function* $h(y)$.

- *IMPORTANT (notation discussion)*. An identical restatement of the above discussion of **exact differentials** can be given by using different notation where the symbols x_1 and x_2 are used instead of x and y. Similarly, the symbols f_1 and f_2 can be used to denote the functions instead of f and g. In ensemble, the collection $\{x_1, x_2\}$ can be denoted symbolically by $\underset{\sim}{x}$. With this change, the previous definition reads as follows: An expression $f_1 dx_1 + f_2 dx_2$ is called an **exact differential** if and only if the following two conditions are met: (1) $f_k = f_k(\underset{\sim}{x})$ * and (2) there exists a function $u(\underset{\sim}{x})$ such that $du = f_1 dx_1 + f_2 dx_2$. If so, then it must be true that $f_k = \partial u / \partial x_k$, which (because k takes values from 1 to 2) represents a set of two equations that may be integrated to solve for u. A necessary and sufficient condition for the potential function u to exist (i.e., for the equations to be integrable) is $\partial f_1 / \partial x_2 = \partial f_2 / \partial x_1$. When using variable symbols that are subscripted as we have done here it is understood that partial differentiation with respect to one subscripted quantity holds the other subscripted quantity constant. For example, the act of writing $\partial f_1 / \partial x_2$ tells the reader that f_1 can be written as a function of x_1 and x_2 and it is understood that x_1 is being held constant in this partial derivative. Recall that, if the equations are integrable, then it will be true that $f_k = \partial u / \partial x_k$. Consequently, the integrability condition, $\partial f_1 / \partial x_2 = \partial f_2 / \partial x_1$ is asserting that $\partial^2 u / \partial x_1 \partial x_2 = \partial^2 u / \partial x_2 \partial x_1$ — in other words, the mixed partial derivatives must give the same result regardless of the order of differentiation. Note that the expression $du = f_1 dx_1 + f_2 dx_2$ can be written in symbolic (structured) notation as $du = \underline{f} \cdot d\underset{\sim}{x}$ and the expression $f_k = \partial u / \partial x_k$ can be written $\underline{f} = \nabla u$, where the gradient is taken with respect to $\underset{\sim}{x}$. The increment in work associated with a force $\underline{f}$ pushing a block a distance $d\underset{\sim}{x}$ along a *frictional* surface is an example of a differential form $\underline{f} \cdot d\underset{\sim}{x}$ that is *not* an exact differential. In this case where no potential function exists, but the expression is still *like an increment*, it is good practice to indicate that the expression is *not* an exact differential by writing a “slash” through the “d”, as in $\text{đ}u = \underline{f} \cdot d\underset{\sim}{x}$; for easier typesetting, some people write $\delta u = \underline{f} \cdot d\underset{\sim}{x}$. By contrast, the increment in work associated with a force force $\underline{f}$ pushing a block a distance $d\underset{\sim}{x}$ against a linear spring is an example of a differential form $\underline{f} \cdot d\underset{\sim}{x}$ that *is* an exact differential (the potential function is $u = \frac{1}{2} k (\underset{\sim}{x} \cdot \underset{\sim}{x})$, where k is the spring constant. For the frictional block, the work accumulates in a *path-dependent* manner. For the spring, the work is path-independent (it only depends on the current value of $\underset{\sim}{x}$, not on all the values it might have had in the past). By the way, a spring does not have to be linear in order for a potential function to exist. The most fundamental requirement is that the force must be expressible as a proper function of position — always check this first.

* This expression is not really an equation. It is just a standard way of indicating that each f_k function depends on $\underset{\sim}{x}$, which means they each can be expressed as functions of x_1 and x_2.

> *"There are a thousand hacking at the branches of evil to one who is striking at the root."*
> *— Henry Thoreau*

3. Matrix Analysis (and some matrix calculus)

Tensor analysis is neither a subset nor a superset of matrix analysis — tensor analysis *complements* matrix analysis. For the purpose of this book, only the following concepts are required from matrix analysis:*

Definition of a matrix

A **matrix** is an ordered array of numbers that are arranged in the form of a "table" having N rows and M columns. If one of the dimensions (N or M) happens to equal 1, then the term "**vector**" is often used, although we prefer the term "**array**" in order to avoid confusion with vectors in the physical sense. A matrix is called "**square**" if $M=N$. We will usually typeset matrices in plain text with brackets such as $[A]$. Much later in this document, we will define the term "tensor" and we will denote tensors by a bold symbol with two under-tildes, such as $\underset{\approx}{\mathbf{A}}$. We will further find that each tensor can be described through the use of an associated 3×3 matrix of components, and we will denote the matrix associated with a tensor by simply surrounding the tensor in square brackets, such as $[\underset{\approx}{\mathbf{A}}]$ or sometimes just $[A]$ if the context is clear.

For matrices of dimension $N \times 1$, we also use braces, as in $\{v\}$; namely, if $N=3$, then

$$\{v\} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{3.1}$$

For matrices of dimension $1 \times M$, we use angled brackets $<v>$; Thus, if $M=3$, then

$$<v> = [v_1, v_2, v_3] \tag{3.2}$$

If attention must be called to the dimensions of a matrix, then they will be shown as subscripts, for example, $[A]_{M \times N}$. The number residing in the i^{th} row and j^{th} column of $[A]$ will be denoted A_{ij}.

[*] Among the references listed in our bibliography, we recommend the following for additional reading: Refs. 26, 23, 1, 36. For quick reference, just about any Schaum's outline or CRC handbook will be helpful too.

Component matrices associated with vectors and tensors (notation explanation)

In this book, vectors will be typeset in bold with one single “under-tilde” (for example, $\underset{\sim}{\mathbf{v}}$) and the associated three components of the vector with respect to some implicitly understood basis will be denoted $\{\underset{\sim}{\mathbf{v}}\}$ or $<\underset{\sim}{\mathbf{v}}>$, depending on whether those components are collected into a column or row matrix, respectively. Similarly, second-order tensors (to be defined later) will be denoted in bold with *two* under-tildes (for example $\underset{\approx}{\mathbf{T}}$). Tensors are often described in terms of an associated 3×3 matrix, which we will denote by placing square brackets around the tensor symbol (for example, $[\underset{\approx}{\mathbf{T}}]$ would denote the *matrix* associated with the *tensor* $\underset{\approx}{\mathbf{T}}$). As was the case with vectors, the matrix of components is presumed referenced to some mutually understood underlying basis — changing the basis will not change the tensor $\underset{\approx}{\mathbf{T}}$, but it *will* change its associated matrix $[\underset{\approx}{\mathbf{T}}]$. These comments will make more sense later.

The matrix product

The matrix product of $[A]_{M \times R}$ times $[B]_{R \times N}$ is a new matrix $[C]_{M \times N}$ written

$$[C] = [A][B] \tag{3.3}$$

Explicitly showing the dimensions,

$$[C]_{M \times N} = [A]_{M \times R}[B]_{R \times N} \tag{3.4}$$

Note that the dimension R must be common to both matrices on the right-hand side of this equation, and this common dimension must reside at the “abutting” position (the trailing dimension of $[A]$ must equal the leading dimension of $[B]$)

The matrix product operation is defined

$$C_{ij} = \sum_{k=1}^{R} A_{ik}B_{kj},$$

where i takes values from 1 to M,
and j takes values from 1 to N. (3.5)

The summation over k ranges from 1 to the common dimension, R. Each individual component C_{ij} is simply the product of the i^{th} row of $[A]$ with the j^{th} column of $[B]$, which is the mindset most people use when actually computing matrix products.

SPECIAL CASE: a matrix times an array. As a special case, suppose that $[F]$ is a square matrix of dimension $N \times N$. Suppose that $\{v\}$ is an array (i.e., column matrix) of dimension $N \times 1$. Then

$$\{u\} = [F]\{v\} \tag{3.6}$$

must be an array of dimension $N \times 1$ with components given by

$$u_i = \sum_{k=1}^{N} F_{ik} v_k, \quad \text{where } i \text{ takes values from 1 to } N \tag{3.7}$$

SPECIAL CASE: ***inner*** *product of two arrays.*

As another special case, suppose the dimensions M and N in Eq. (3.5) both equal 1. Now we are talking about the matrix product of two *arrays*. Then the free indices i and j in Eq. (3.5) simply range from 1 to 1 giving the result

$$\text{If } M = 1 \text{ and } N = 1, \; C_{11} = \sum_{k=1}^{R} A_{1k} B_{k1} \tag{3.8}$$

When working with matrices with only one row or only one column, recall that explicit mention of the "1" in the index formulas is usually omitted. Also, 1×1 matrices (like the matrix $[C]$ in this case) are typeset without showing any subscripts at all. Consequently this result would be written

$$C = \sum_{k=1}^{R} A_k B_k \tag{3.9}$$

In other words, this array "inner product" simply sums over every product of corresponding components from each array. This array inner product is called the "dot" product in 3D engineering vector analysis. When $\{A\}$ and $\{B\}$ are arrays, this inner product will often be seen written using array notation as

$$\{A\}^T\{B\} \qquad \text{or} \qquad <A>\{B\} \tag{3.10}$$

SPECIAL CASE: ***outer*** *product of two arrays.*

As a very different special case, suppose that the *common* dimension R equals 1. Then the matrices in Eq. (3.5) again get treated and typeset as arrays instead of matrices but this time, the *summation* over R becomes trivial (there is only one term in the sum, so there isn't really a sum at all). Specifically

If $R = 1$, then Eq. (3.5) becomes simply

$$C_{ij} = A_i B_j$$
$$\text{where } i \text{ takes values from 1 to } M, \quad \text{and } j \text{ takes values from 1 to } N. \tag{3.11}$$

Unlike the inner product which produces a single number, this "outer product" multiplies to arrays together to obtain a matrix. When $\{A\}$ and $\{B\}$ are arrays, this outer product will often be seen written using array notation as

$$\{A\}\{B\}^T \qquad \text{or} \qquad \{A\}<B> \tag{3.12}$$

EXAMPLE: Consider two arrays:

$$<u> = \begin{bmatrix} 1 & -3 & 4 \end{bmatrix} \qquad \text{and} \qquad <v> = \begin{bmatrix} 2 & 5 & -7 \end{bmatrix} \tag{3.13}$$

The inner product between $\{u\}$ and $\{v\}$ is

$$<u>\{v\} = \begin{bmatrix} 1 & -3 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ -7 \end{bmatrix} = (1)(2) + (-3)(5) + (4)(-7) = -41 \tag{3.14}$$

This also equals the inner product between $\{v\}$ and $\{u\}$ because the inner product is commutative. The inner product is just a single number.

Now consider the outer product, which results in a full matrix and is not commutative. The outer product between $\{u\}$ and $\{v\}$ is

$$\{u\}<v> = \begin{bmatrix} 1 \\ -3 \\ 4 \end{bmatrix} \begin{bmatrix} 2 & 5 & -7 \end{bmatrix} = \begin{bmatrix} (1)(2) & (1)(5) & (1)(-7) \\ (-3)(2) & (-3)(5) & (-3)(-7) \\ (4)(2) & (4)(5) & (4)(-7) \end{bmatrix} = \begin{bmatrix} 2 & 5 & -7 \\ -6 & -15 & 21 \\ 8 & 20 & -28 \end{bmatrix} \tag{3.15}$$

The outer product between $\{v\}$ and $\{u\}$ is

$$\{v\}<u> = \begin{bmatrix} 2 \\ 5 \\ -7 \end{bmatrix} \begin{bmatrix} 1 & -3 & 4 \end{bmatrix} = \begin{bmatrix} (2)(1) & (2)(-3) & (2)(4) \\ (5)(1) & (5)(-3) & (5)(4) \\ (-7)(1) & (-7)(-3) & (-7)(4) \end{bmatrix} = \begin{bmatrix} 2 & -6 & 8 \\ 5 & -15 & 20 \\ -7 & 21 & -28 \end{bmatrix} \tag{3.16}$$

Note that the resulting outer product matrices are not equal (they differ by a transpose), so the outer product is not commutative. Outer products play a pivotal role in tensor analysis because they are related to dyads, which are primitive kinds of tensors. The component arrays for the base vectors in vector analysis are

$$<\underset{\sim}{e}_1> = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$<\underset{\sim}{e}_2> = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$<\underset{\sim}{e}_3> = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

Note that the inner product between a base vector and itself equals 1, and the inner product between a base vector and any of the other two different base vectors equals zero. Later on, we will use the notation $[\underset{\sim}{e}_i\underset{\sim}{e}_j]$ to denote the *outer* product, $\{\underset{\sim}{e}_i\}<\underset{\sim}{e}_j>$, of the "$i^{\text{th}}$" base vector with the "$j^{\text{th}}$" base vector. The result will be that $[\underset{\sim}{e}_i\underset{\sim}{e}_j]$ is a 3×3 matrix that has zeros everywhere except for a 1 in the ij location. For example,

$$[\underset{\sim}{e}_1\underset{\sim}{e}_1] = \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix} \qquad [\underset{\sim}{e}_1\underset{\sim}{e}_2] = \begin{bmatrix}0&1&0\\0&0&0\\0&0&0\end{bmatrix} \qquad [\underset{\sim}{e}_1\underset{\sim}{e}_3] = \begin{bmatrix}0&0&1\\0&0&0\\0&0&0\end{bmatrix}$$

$$[\underset{\sim}{e}_2\underset{\sim}{e}_1] = \begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix} \qquad [\underset{\sim}{e}_2\underset{\sim}{e}_2] = \begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix} \qquad [\underset{\sim}{e}_2\underset{\sim}{e}_3] = \begin{bmatrix}0&0&0\\0&0&1\\0&0&0\end{bmatrix}$$

$$[\underset{\sim}{e}_3\underset{\sim}{e}_1] = \begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix} \qquad [\underset{\sim}{e}_3\underset{\sim}{e}_2] = \begin{bmatrix}0&0&0\\0&0&0\\0&1&0\end{bmatrix} \qquad [\underset{\sim}{e}_3\underset{\sim}{e}_3] = \begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix} \tag{3.18}$$

The Kronecker delta

The **Kronecker delta** is a symbol δ_{ij} whose value depends on the subscripts i and j. Specifically,

$$\delta_{ij} = \begin{cases} 1 & \text{if } i{=}j \\ 0 & \text{if } i \neq j \end{cases} \tag{3.19}$$

The identity matrix

The identity matrix, denoted $[I]$, has all zero components except 1 on the diagonal. For example, the 3×3 identity is

$$[I] = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \tag{3.20}$$

The ij component of the identity is given by Kronecker delta, δ_{ij}. That is, recalling Eq. (3.19),

$$\begin{bmatrix}\delta_{11}&\delta_{12}&\delta_{13}\\\delta_{21}&\delta_{22}&\delta_{23}\\\delta_{31}&\delta_{32}&\delta_{33}\end{bmatrix} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \tag{3.21}$$

Note that, for any array $\{v\}$

$$[I]\{v\} = \{v\} \tag{3.22}$$

In component form, this equation is written

$$\sum_{k=1}^{3} \delta_{ik} v_k = v_i \tag{3.23}$$

This represents a set of three equations. The "free index" i takes the values 1, 2, and 3.

By the way, referring to Eq. (3.18), note that

$$[I] = [\underset{\sim}{e}_1\underset{\sim}{e}_1] + [\underset{\sim}{e}_2\underset{\sim}{e}_2] + [\underset{\sim}{e}_3\underset{\sim}{e}_3] \tag{3.24}$$

Equivalently, note that

$$[I] = \sum_{i=1}^{3}\sum_{j=1}^{3} \delta_{ij}[\underset{\sim}{e}_i\underset{\sim}{e}_j] \tag{3.25}$$

Derivatives of vector and matrix expressions

Junior-high-school students learn about simple functions of a single variable such as $y = x^2$ or $y = \ln x$ or $y = \sin x$. In calculus, they learn how to differentiate these functions. Eventually students begin working with functions (and even families of functions) of more than one variable such as $y = ue^w$, $s = u + w$, and $r = u/w$. Differentiating functions of more than one variable requires the chain rule for partial differentiation. In physical applications, two or more of the independent variables might be so intimately related to each other (e.g., perhaps they are the three components of a velocity vector) that it works better to denote them by the *same* symbol distinguishing between them only by numbered subscripts, such as v_1, v_2, v_3.

SIDEBAR (functional dependence structure rule). When working with a small number of variables, the structure $y = f(x)$ is a conventional way to communicate to readers that y is a function of x. However, serious scientific research often requires simultaneous processing of so many different variables that giving up a new letter of the alphabet (f) merely to indicate that y depends on x is not feasible. Consequently scientists frequently use the structure $y = y(x)$ to mean "*y* is a variable that is expressible as a function of *x*." Of course, if the focus of a discussion is aimed on the *function* itself (rather than on simply indicating dependencies), the function should be denoted by a symbol that differs from its output. Many times, you might see something like $y = \hat{y}(x)$, but using the hat would preclude your using it to denote something else later on. Those of us who are running extremely short on available letters of the alphabet don't even want to add "hat" decorations or other diacritical marks to denote functions — we need the hats and other marks to denote additional distinct variables.

In calculus classes, students learn that the chain rule may be used to find, say, the time derivative of a function $y = y(x)$ to give $\dot{y} = \frac{dy}{dx}\dot{x}$, where the superimposed dot denotes the time rate. In more complicated applications, one must often deal simultaneously with *families* of variables that might themselves depend on some other *family* of independent variables. You might, for example, be working with a set of equations of the form

$$y_1 = y_1(x_1, x_2, x_3) \tag{3.26}$$

$$y_2 = y_2(x_1, x_2, x_3) \tag{3.27}$$

In this case, the time derivative is obtained via the chain rule to give

$$\dot{y}_1 = \left(\frac{\partial y_1}{\partial x_1}\right)\dot{x}_1 + \left(\frac{\partial y_1}{\partial x_2}\right)\dot{x}_2 + \left(\frac{\partial y_1}{\partial x_3}\right)\dot{x}_3 \tag{3.28}$$

$$\dot{y}_2 = \left(\frac{\partial y_2}{\partial x_1}\right)\dot{x}_1 + \left(\frac{\partial y_2}{\partial x_2}\right)\dot{x}_2 + \left(\frac{\partial y_2}{\partial x_3}\right)\dot{x}_3 \tag{3.29}$$

SIDEBAR (structure rule for partial differentiation): Any derivative with respect to a member of a subscripted family of variables (in this case, x_1, x_2, and x_3), is *understood* to be taken holding the remaining members of that family constant. If a function depends additionally on other quantities, then the fact that they are being held constant must be noted explicitly by showing them as a subscript. If, for example, $z = z(x_1, x_2, x_3, s, r, u_1, u_2, u_3)$, the derivative of z with respect to x_2 would be written $\left(\frac{\partial z}{\partial x_2}\right)_{s,r,\underset{\sim}{u}}$, where $\underset{\sim}{u}$ is a shorthand "ensemble" representation for the three u_k variables.

The use of subscripts to demark different variables is especially convenient because the above two equations can be written compactly as

$$\dot{y}_i = \sum_{j=1}^{3} \left(\frac{\partial y_i}{\partial x_j}\right)\dot{x}_j \text{, where the free index } i \text{ takes values 1 and 2.} \tag{3.30}$$

Whenever you encounter a new function or transformation, it is a good idea to immediately note the derivative of that function because you will undoubtedly later need it for rate equations. Consequently, throughout this book, we will often cite derivatives for any new function that we introduce. To illustrate, we will now discuss derivatives of subscripted quantities *with respect to themselves.*

Derivative of an array with respect to itself

"Everyone" knows that $dx/dx = 1$. In other words, the derivative of a number with respect to itself equals unity.* If $y = x$ then $dy/dx = 1$. What is the generalization of this statement for an *array* of numbers? Specifically, we have $\underline{y} = \underset{\sim}{x}$, or written out in less cryptic (non-ensemble) form:

$$y_1 = x_1 \tag{3.31a}$$

$$y_2 = x_2 \tag{3.31b}$$

$$y_3 = x_3 \tag{3.31b}$$

This relationship can be regarded as the *identity* transformation in which a three component array $<y>$ identically equals another array $<x>$.

Then, trivially,

$$\frac{\partial y_1}{\partial x_1} = 1 \qquad \frac{\partial y_1}{\partial x_2} = 0 \qquad \frac{\partial y_1}{\partial x_3} = 0 \tag{3.32a}$$

$$\frac{\partial y_2}{\partial x_1} = 0 \qquad \frac{\partial y_2}{\partial x_2} = 1 \qquad \frac{\partial y_2}{\partial x_3} = 0 \tag{3.32b}$$

$$\frac{\partial y_3}{\partial x_1} = 0 \qquad \frac{\partial y_3}{\partial x_2} = 0 \qquad \frac{\partial y_3}{\partial x_3} = 1 \tag{3.32c}$$

This result can be written compactly as $\frac{\partial y_i}{\partial x_j} = \delta_{ij}$, or, since $y_i = x_i$,

* The word "unity" is less snobbishly known as the number "one" or more snobbishly as the multiplicative "identity". Actually "unity" is a better choice than "one" despite sounding pompous because phrases like "the result is one" leaves readers asking "one what?"

$$\boxed{\frac{\partial x_i}{\partial x_j} = \delta_{ij}} \qquad (3.33)$$

This is the array analog of the scalar identity, $dx/dx = 1$. Later on, when we introduce direct structural notation, the above result will be written*

$$\boxed{\frac{d\underset{\sim}{\mathbf{x}}}{d\underset{\sim}{\mathbf{x}}} = \underset{\approx}{\mathbf{I}}} \qquad (3.34)$$

Throughout this book, we will be defining various operations that take scalars, arrays, or matrices as input to construct some new scalar, array, or matrix as output. (The above trivial example was the identity operation. It took an array $<x>$ and spit $<x>$ right back out as output.) Knowing that derivatives of operations will eventually be needed, we will attempt to wrap up every discussion of new operations by giving you the expression for the function's derivative in both index and structured notation. The calculus machinery needed to *prove* the derivatives will not be discussed until Chapter 21 on page 251, so you should just consider the derivatives to be provided for future reference without proof.

Derivative of a matrix with respect to itself

Suppose we have an $N \times M$ matrix $[A]$. Performing an analysis similar to what was done above, we can assert that the derivative of A_{ij} with respect to A_{mn} (holding the other components constant) will be zero unless the subscripts on A_{ij} are exactly the same as those on A_{mn}, in which case the result would equal 1. In order for ij to equal mn, you must have $i=m$ and $j=n$. Therefore, we can assert that

$$\boxed{\frac{\partial A_{ij}}{\partial A_{mn}} = \delta_{im}\delta_{jn}},$$

where i and m range from 1 to N
and j and n range from 1 to M. (3.35)

Later on, when subscripts always range from 1 to 3 for vectors and tensors, this result will be cast into a new (structured) notation as

$$\boxed{\frac{\partial \underset{\approx}{\mathbf{A}}}{\partial \underset{\approx}{\mathbf{A}}} = \underset{\approx\approx}{\delta}}, \qquad (3.36)$$

where $\underset{\approx\approx}{\delta}$ will be shown to equal a fourth-order identity tensor.

* To understand why we used "∂" in Eq. (3.33) but "d" in (3.34), see the discussion on page 266.

The transpose of a matrix

The transpose of a matrix $[A]_{M \times N}$ is a new matrix $[B]_{N \times M}$ (note the reversed dimensions). The components of the transpose are

$$B_{ij} = A_{ji}$$
where i takes values from 1 to N,
and j takes values from 1 to M. (3.37)

The transpose of $[A]$ is written as $[A]^T$, and the notation A^T_{ij} means the ij component of $[A]^T$. Thus, the above equation may be written

$$A^T_{ij} = A_{ji}$$
where i takes values from 1 to N,
and j takes values from 1 to M. (3.38)

The dimensions of $[A]$ and $[A]^T$ are reverses of each other. Thus, for example, if $\{v\}$ is an $N \times 1$ matrix, then $\{v\}^T$ is a $1 \times N$ matrix. In other words,

$$\{v\}^T = \langle v \rangle \qquad \text{and} \qquad \langle v \rangle^T = \{v\} \tag{3.39}$$

The transpose of a product is the *reverse* product of the transposes. For example,

$$\begin{aligned}&([A][B])^T = [B]^T[A]^T\text{, and}\\&(\langle v \rangle[A])^T = [A]^T\langle v \rangle^T = [A]^T\{v\}\end{aligned} \tag{3.40}$$

Derivative of the transpose:

$$\frac{\partial A^T_{ij}}{\partial A_{rs}} = \delta_{is}\delta_{jr} \tag{3.41}$$

The inner product of two column matrices

The inner product of two column matrices, $\{v\}_{N \times 1}$ and $\{w\}_{N \times 1}$, each having the same dimension is defined

$$\{v\}^T\{w\}, \quad \text{or, using the angled-bracket notation, } \langle v \rangle\{w\} \tag{3.42}$$

Applying the definition of matrix multiplication, the result is a 1×1 matrix (which is just a single number) given by

$$\sum_{k=1}^{N} v_k w_k \tag{3.43}$$

If $\{v\}$ and $\{w\}$ contain components of two vectors $\underset{\sim}{v}$ and $\underset{\sim}{w}$ then the inner product gives the same result as the vector "dot" product $\underset{\sim}{v} \bullet \underset{\sim}{w}$, defined later. Noting that the above relationship treats the components of $\{v\}$ and $\{w\}$ in precisely the same way — the result is unchanged if these two arrays are swapped. However, the notation of Eq. (3.42) makes it *look* as though $\{v\}$ and $\{w\}$ play different roles because one of them has a transpose and the other does not. Thus, you might see use of alternative notations for the inner product that are more "equitable" looking such as

$$\{v\}*\{w\} = \{w\}*\{v\} = \sum_{k=1}^{N} v_k w_k \tag{3.44}$$

Here, the raised star ($*$) is being used as a new operator symbol — a structure. Later on, when dealing with vectors instead of arrays, we use a raised dot for the inner product (as in $\underset{\sim}{v} \bullet \underset{\sim}{w}$). Many people like to use a raised dot to denote any inner product regardless of the nature of the operand, but we will use the raised dot only for inner products between vectors. Different symbols will be later defined for different kinds of operands.

Derivatives of the inner product: The inner product is a *binary* operation (i.e., it has two arguments, $\{v\}$ and $\{w\}$. Thus, we can discuss formulas for partial derivatives with respect to one argument, holding the other constant:

$$\frac{\partial(\{v\}*\{w\})}{\partial v_i} = \frac{\partial}{\partial v_i}\left(\sum_{k=1}^{N} v_k w_k\right) = \sum_{k=1}^{N} \delta_{ki} w_k = w_i \tag{3.45}$$

similarly,

$$\frac{\partial(\{v\}*\{w\})}{\partial w_i} = \frac{\partial}{\partial w_i}\left(\sum_{k=1}^{N} v_k w_k\right) = \sum_{k=1}^{N} v_k \delta_{ki} = v_i \tag{3.46}$$

In direct (structured) notation, this result will later be written as

$$\frac{\partial(\underset{\sim}{v} \bullet \underset{\sim}{w})}{\partial \underset{\sim}{v}} = \underset{\sim}{w} \qquad \text{and} \qquad \frac{\partial(\underset{\sim}{v} \bullet \underset{\sim}{w})}{\partial \underset{\sim}{w}} = \underset{\sim}{v} \tag{3.47}$$

As a special case, consider the case that both arrays are the same. In this case where $\{v\} = \{w\}$, it isn't possible to take the derivative with respect to $\{v\}$ while holding $\{w\}$ constant. Since $\{w\}$ equals $\{v\}\}$, it must vary whenever $\{v\}$ varies, so let's work this one out from scratch:

$$\frac{\partial(\{v\}*\{v\})}{\partial v_i} = \frac{\partial}{\partial v_i}\left(\sum_{k=1}^{N} v_k v_k\right) = \sum_{k=1}^{N} (\delta_{ki} v_k + v_k \delta_{ki}) = 2v_i \tag{3.48}$$

This result is analogous to the simple scalar equation $dx^2/dx = 2x$. In structured vector notation,

$$\frac{d(\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{v}})}{d\underset{\sim}{\mathbf{v}}} = 2\underset{\sim}{\mathbf{v}} \qquad (3.49)$$

To understand why we used “d” instead of “∂” in this equation, refer to the “sidebar” on page 266.

The outer product of two column matrices.

The outer product of two column matrices, $\{a\}_{M \times 1}$ and $\{b\}_{N \times 1}$, not necessarily of the same dimension is defined

$$\{a\}\{b\}^T, \quad \text{or, using the angled-bracket notation, } \{a\}<b> \qquad (3.50)$$

For this case, the value of the “adjacent” dimension R in Eq. (3.5) is just 1, so the summation ranges from 1 to 1 (which means that it is just a solitary term).

The result of the outer product is an $M \times N$ matrix, whose ij component is given by $a_i b_j$. If $\{a\}$ and $\{b\}$ contain components of two vectors $\underset{\sim}{\mathbf{a}}$ and $\underset{\sim}{\mathbf{b}}$ then the outer product gives the matrix corresponding to the “dyadic” product, $\underset{\sim}{\mathbf{a}}\underset{\sim}{\mathbf{b}}$ (also often denoted $\underset{\sim}{\mathbf{a}} \otimes \underset{\sim}{\mathbf{b}}$), to be discussed in gory detail later.

The trace of a square matrix

A matrix $[A]_{N \times N}$ is called “square” because it has as many rows as it has columns. The trace of a square matrix is simply the sum of the diagonal components:

$$\boxed{\text{tr}[A] = p\sum_{k=1}^{N} A_{kk}} \qquad (3.51)$$

The trace operation satisfies the following properties:

$$\text{tr}([A]^T) = \text{tr}[A] \qquad (3.52)$$

$$\text{tr}([A][B]) = \text{tr}([B][A]) \qquad \text{(cyclic property)} \qquad (3.53)$$

Derivative of the trace. The trace of a matrix is a function of the matrix’s components. If, for example, $[A]$ is a 3×3 matrix, then

$$\text{tr}[A] = A_{11} + A_{22} + A_{33} \qquad (3.54)$$

Therefore

$$\frac{\partial \text{tr}[A]}{\partial A_{11}} = 1, \qquad \frac{\partial \text{tr}[A]}{\partial A_{12}} = 0, \qquad \text{etc.} \qquad (3.55)$$

Note that the derivative of $[A]$ with respect to A_{ij} equals 1 if i=j and 0 if $i \neq j$. This is precisely the definition of the Kronecker delta in Eq. (A.3.19). Thus,

$$\boxed{\frac{\partial \text{tr}[A]}{\partial A_{ij}} = \delta_{ij}} \qquad (3.56)$$

This result holds regardless of the size of the matrix. In structured tensor notation, this result is written

$$\frac{d\mathrm{tr}[\underset{\approx}{\boldsymbol{A}}]}{d\underset{\approx}{\boldsymbol{A}}} = \underset{\approx}{\boldsymbol{I}} \tag{3.57}$$

Note the distinction between Eq. (3.56) and (3.35); in one case we are differentiating the trace of a matrix with respect to the matrix, while, in the other case, we are differentiating the matrix itself with respect to itself. In Eq. (3.56) the result is the second-order identity (two free indices), whereas, in Eq. (3.35), the result is the fourth-order identity (four free indices).

The matrix *inner* product

Given two matrices $[A]_{M\times N}$ and $[B]_{M\times N}$ *of the same class* (i.e., having the same dimensions), the matrix inner product is a scalar obtained by summing terms in which each component of $[A]$ multiplied by the corresponding component of $[B]$. Stated more explicitly, the matrix inner product (which we will denote using "$*$" as the operator symbol) is given by

$$[A]*[B] = \sum_{n=1}^{N}\sum_{m=1}^{M} A_{mn}B_{mn} \tag{3.58}$$

Note that this is identical to

$$[A]*[B] = \mathrm{tr}([A][B]^T) = \mathrm{tr}([A]^T[B]) \tag{3.59}$$

Of course, if the second dimension N happens to equal 1, then this matrix inner product becomes identical to the inner product defined in Eq. (3.44) for arrays.

Later on, after we define what is meant by the term "tensor", we will find that ordinary engineering tensors can be described via 3×3 component matrices, in much the same way that a vector $\underset{\sim}{\boldsymbol{u}}$ can be described by a 3×1 component array. If $[A]$ and $[B]$ contain components of two second-order tensors $\underset{\approx}{\boldsymbol{A}}$ and $\underset{\approx}{\boldsymbol{B}}$ then the matrix inner product gives the same result as what we will later refer to as the tensor "double dot" product $\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}}$, and the specialization of Eq. (3.58) using tensor notation is

$$\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}} = \sum_{n=1}^{3}\sum_{m=1}^{3} A_{mn}B_{mn} \tag{3.60}$$

Just as the dot product $\underset{\sim}{\boldsymbol{u}} \bullet \underset{\sim}{\boldsymbol{v}}$ between two vectors tells you something about the angle between those vectors, the double-dot product between two tensors also tells about the degree of alignment between them.

Derivative of the matrix inner product. In analog to Eqs. (3.45) through (3.47),

$$\frac{\partial([A]*[B])}{\partial A_{ij}} = B_{ij} \qquad \text{and} \qquad \frac{\partial([A]*[B])}{\partial B_{ij}} = A_{ij} \tag{3.61}$$

$$\frac{\partial(\underset{\approx}{\mathbf{A}}\mathbf{:}\underset{\approx}{\mathbf{B}})}{\partial\underset{\approx}{\mathbf{A}}} = \underset{\approx}{\mathbf{B}} \qquad \text{and} \qquad \frac{\partial(\underset{\approx}{\mathbf{A}}\bullet\underset{\approx}{\mathbf{B}})}{\partial\underset{\approx}{\mathbf{B}}} = \underset{\approx}{\mathbf{A}} \tag{3.62}$$

In analog to Eq. (3.49),

$$\frac{d(\underset{\approx}{\mathbf{A}}\mathbf{:}\underset{\approx}{\mathbf{A}})}{d\underset{\approx}{\mathbf{A}}} = 2\underset{\approx}{\mathbf{A}} \tag{3.63}$$

Magnitudes and positivity property of the inner product

The “magnitude” of a matrix* is given by

$$||[A]|| = +\sqrt{[A]*[A]} \tag{3.64}$$

or

$$||[A]|| = \sqrt{\sum_{n=1}^{3}\sum_{m=1}^{3} A_{mn}A_{mn}} = \sqrt{\sum_{n=1}^{3}\sum_{m=1}^{3} (A_{mn})^2} \tag{3.65}$$

Note that the positive square root must be taken. This equation is analogous to the ordinary scalar equation, $|x| = +\sqrt{x^2}$. Just as the operation $\sqrt{\underset{\sim}{\mathbf{u}}\bullet\underset{\sim}{\mathbf{u}}}$ gives you the magnitude of a vector, the operation $\sqrt{\underset{\approx}{\mathbf{A}}\mathbf{:}\underset{\approx}{\mathbf{A}}}$ gives the magnitude of a tensor:

$$||\underset{\approx}{\mathbf{A}}|| = \sqrt{\underset{\approx}{\mathbf{A}}\mathbf{:}\underset{\approx}{\mathbf{A}}} = \sqrt{\sum_{n=1}^{3}\sum_{m=1}^{3} A_{mn}A_{mn}} = \sqrt{\sum_{n=1}^{3}\sum_{m=1}^{3} (A_{mn})^2} \tag{3.66}$$

Careful readers should be asking themselves: How can we be sure that we won’t be trying to take the square root of a negative number? The answer is simple. Recalling our previous statement that all vectors, matrices, and tensors in this book are presumed to have *real* components unless otherwise indicated, you know that every term in the summation in Eq. (3.66) is the square of a real number†. Hence, the final result for the summation in Eq. (3.66) will not be negative. Taking the positive square root gives a positive real value for the tensor’s magnitude.

* Also called the L_2 norm.

† When dealing with matrices whose components might be complex, the definition of the inner product needs to be modified to be $[A]*[B] = \text{tr}([\bar{A}]^T[B])$, where the overbar denotes the complex conjugate. When this modification is made, each term in the operation $[A]*[A]$ becomes of the form $\bar{A}_{mn}A_{mn}$ which will be a nonnegative real number. The notation, $[\bar{A}]^T$, which represents the transpose of the conjugate (or, equivalently, the conjugate of the transpose) is commonly referred to as the “Hermitian” and is denoted $[A]^H$. Throughout this book, whenever you see a transpose operation, it is likely that the equation will generalize to complex components by replacing the transpose with the Hermitian — this rule-of-thumb is not absolute, so you should always double check against fully developed complex linear algebra textbooks.

The ability to use the inner product to compute a magnitude of a tensor (or matrix) is pivotal in the very special and carefully crafted mathematician's definition of "inner product." Among the required properties of the inner product, the most important is **positivity**, which says that the inner product of a matrix with itself must result in a positive number (or zero if and only if the matrix itself is zero).

Derivative of the magnitude. By the chain rule, using Eq. (3.63),

$$\frac{d\sqrt{\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{A}}}}{d\underset{\approx}{\boldsymbol{A}}} = \frac{d(\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{A}})^{1/2}}{d\underset{\approx}{\boldsymbol{A}}} = \frac{1}{2}(\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{A}})^{-1/2}\frac{d(\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{A}})}{d\underset{\approx}{\boldsymbol{A}}} = (\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{A}})^{-1/2}\underset{\approx}{\boldsymbol{A}} \tag{3.67}$$

or

$$\frac{d\|\underset{\approx}{\boldsymbol{A}}\|}{d\underset{\approx}{\boldsymbol{A}}} = \frac{\underset{\approx}{\boldsymbol{A}}}{\|\underset{\approx}{\boldsymbol{A}}\|} \tag{3.68}$$

The version of this equation that applies to vectors is

$$\frac{d\|\underset{\sim}{\boldsymbol{v}}\|}{d\underset{\sim}{\boldsymbol{v}}} = \frac{\underset{\sim}{\boldsymbol{v}}}{\|\underset{\sim}{\boldsymbol{v}}\|} \tag{3.69}$$

Physically, this shows that the derivative of the magnitude of a vector $\underset{\sim}{\boldsymbol{v}}$ with respect to the vector itself simply equals a unit vector in the direction of $\underset{\sim}{\boldsymbol{v}}$.

Norms. Let k be a real number. Consider the following summation in which each component of a matrix is raised to the power k, with the final result raised to the power $1/k$.

$$\|[A]\|_k = \left(\sum_{n=1}^{3}\sum_{m=1}^{3}(A_{mn})^k\right)^{1/k} \tag{3.70}$$

This expression is called the "L_k-norm" of the matrix, and it applies to *arrays* (i.e., $M \times 1$ matrices) as a special case. Note that Eq. (3.65) is the L_2 norm.

As k becomes larger in Eq. (3.70), the A_{mn} components that are largest in absolute value compared to the other components become even larger relative to the other components when raised to large values of the k exponent. In the limit as $k \to \infty$, the largest component of $[A]$ "swamps out" all of the other components in the summation, so that after raising the sum to the power $1/k$, it can be shown that the "L_∞-norm" produces is given by the absolute value of this dominant component:

$$\|[A]\|_\infty = \max_{m,\,n}|A_{mn}| \tag{3.71}$$

Although this and the other L_k norms are interesting and useful for *matrix* analysis, it turns out that they are not of much use in *vector and tensor* analysis. The reason revolves around a concept called "basis invariance." In vector analysis, the components of a vector $\underset{\sim}{\boldsymbol{v}}$ are frequently assembled into a 3×1 array $<v_1, v_2, v_3>$. The L_∞ norm of this array would be simply $\max(|v_1|, |v_2|, |v_3|)$. However, the components of a vector depend on your choice of basis. The components of the *same* vector $\underset{\sim}{\boldsymbol{v}}$ will have *different values* with respect some other choice of basis. The max component in one basis will not generally equal the max component in a different basis, so we say that the L_∞ norm is not invariant under basis transformations for vectors. In general the only L_k norm that *does* give the same result regardless of the basis used for vectors (and tensors) is the L_2-norm. Hence for vector and tensor analysis, only the L_2 norm (or its *weighted* generalization described below) is of interest.

Weighted or "energy" norms. Let $[M]$ denote a square positive-definite $N \times N$ matrix.* Then the *weighted* or *energy* norm of any vector $\{v\}$ is defined

$$||v||^2_{\text{energy}} = \frac{1}{2}\{v\}^T[M]\{v\} \tag{3.72}$$

The reason for the moniker "energy" should be clear because of the similarity that this equation has with the equation for kinetic energy $\frac{1}{2}mv^2$ for a single particle of mass m moving with speed v. As a matter of fact, the expression $\frac{1}{2}mv^2$ can be written in exactly the form of Eq. (3.72) by taking $[M] = m[I]$. In mechanics, the total kinetic energy of a rigid body is $\frac{1}{2}\{\omega\}^T[\Phi]\{\omega\}$, where $\{\omega\}$ contains the components of the angular velocity vector and $[\Phi]$ is the rotational moment of inertia of the body about the axis of rotation, which can be proved to be positive definite.

Note that ordinary L_2 norm of a vector is merely a special case of the more general weighted or energy norm of Eq. (3.72) obtained by setting $[M] = 2[I]$. Weighted norms can also be generalized to apply to matrices, as we will see in later discussions of material constitutive modeling.

Derivative of the energy norm. In component form, Eq. (3.72) may be written

$$||v||^2_{\text{energy}} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} v_i M_{ij} v_k \tag{3.73}$$

from which it follows that

$$\frac{\partial(||v||^2_{\text{energy}})}{\partial v_k} = \sum_{s=1}^{N} M_{ks} v_s \tag{3.74}$$

The above equations apply to the *square* of the energy norm. The derivative of the energy norm itself is

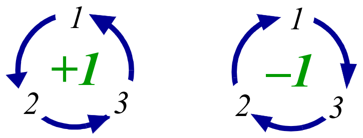
$$\frac{\partial(||v||_{\text{energy}})}{\partial v_p} = \frac{\partial\left(\sqrt{||v||^2_{\text{energy}}}\right)}{\partial v_k} = \frac{1}{||v||_{\text{energy}}}\sum_{s=1}^{N} M_{ks} v_s \tag{3.75}$$

* By "positive definite" we mean it satisfies the condition that $\{a\}^T[W]\{b\} > 0$ for all $N \times 1$ arrays $\{a\}$ and $\{b\}$ except, of course, arrays containing all zeros. See page 47 for details.

The 3D permutation symbol

The 3D permutation symbol (also known as the **alternating symbol** or the **Levi-Civita density**) is defined

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{if } ijk = 123,\ 231, \text{ or } 312 \\ -1 & \text{if } ijk = 321,\ 132, \text{ or } 213 \\ 0 & \text{otherwise} \end{cases} \qquad (3.76)$$



For example, $\varepsilon_{231}=1$, $\varepsilon_{213}=-1$, and $\varepsilon_{212}=0$. Note that the indices on ε_{ijk} may be permuted cyclically without changing the value of the result. Furthermore, inverting any two indices will change the sign of the value. Thus, the permutation symbol has the following properties:

$$\varepsilon_{ijk} = \varepsilon_{jki} = \varepsilon_{kij} = -\varepsilon_{jik} = -\varepsilon_{ikj} = -\varepsilon_{kji} \qquad (3.77)$$

The term "3D" is used to indicate that there are three subscripts on ε_{ijk} each of which take on values from 1 to 3.*

The ε-δ (E-delta) identity

If the alternating symbol is multiplied by another alternating symbol with exactly one index being summed, a very famous and extraordinarily useful result, called the ε-δ identity, applies. Namely,

$$\sum_{n=1}^{3} \varepsilon_{ijn}\varepsilon_{kln} = \delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}. \qquad (3.78)$$

Here, we have highlighted the index "*n*" in red to emphasize that it is summed, while the other indices (*i*, *j*, *k*, and *l*) are "free" indices taking on values from 1 to 3. Later on, we are going to introduce the "summation convention" which states that expressions having one index appearing exactly twice in a term should be understood summed over from 1 to 3 over that index. Index symbols that appear exactly once in one term are called "free indices," taking values from 1 to 3, and they must appear exactly once in all of the other terms. Using this convention, the above equation can be written as

$$\varepsilon_{ijn}\varepsilon_{kln} = \delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}. \qquad (3.79)$$

[*] Though not needed for our purposes, the 2D permutation symbol ε_{ij} is defined to equal zero if $i=j$, +1 if $ij=12$, and −1 if $ij = 21$. The 4D permutation symbol ε_{ijkl} is defined to equal zero if any of the four indices are equal; it is +1 if $ijkl$ is an *even* permutation of 1234 and −1 if $ijkl$ is an *odd* permutation. A permutation is simply a rearrangement. The permutation $ijkl$ is even if rearranging it back to 1234 can be accomplished by an even number of moves that exchange two elements at a time. A *cyclic* permutation of an *n*-D permutation symbol will change sign if n is even, but remain unchanged if n is odd. Thus, for our 3D permutation symbol, cyclic permutations don't change sign, whereas cyclic permutations of the 4D permutation symbol *will* change sign.

Because of the cyclic properties of the permutation symbol, the ε-δ identity applies whenever *any* index on the first ε matches *any* index on the second ε. For example, the above equation would apply to the expression $\varepsilon_{nij}\varepsilon_{kln}$ because $\varepsilon_{nij} = \varepsilon_{ijn}$. The *negative* of the ε-δ identity would also apply to the expression $\varepsilon_{inj}\varepsilon_{kln}$ because $\varepsilon_{inj} = -\varepsilon_{ijn}$. Of course, if a negative permutation is also required to place the summation index at the end of the second ε, then the *positive* of the ε-δ identity would again apply.

To make an expression fit the index structure of Eq. (3.78), most people laboriously apply the cyclic property to each alternating symbol until the summed index is located at the trailing side on both of them. Keeping track of whether or not these manipulations will require changing the final sign of the right hand side of the ε-δ identity is one of the most common and avoidable careless mistakes made when people use this identity. Even once the summation index has been properly positioned at the trailing end of each alternating symbol, most people then apply a slow (and again error-prone) process of figuring out where the free indices go. Typically people apply a "left-right/outside-inside" rule. By this, we mean that the free indices on the *left* sides of ε_{ijn} and ε_{kln} are the indices that go on the first δ, then the *right* free indices go on the second δ, then the *outer free* indices go on the third δ, and (finally) the *inner* free indices go on the last δ. The good news is... **you don't have to do it this way!** By thinking about the ε-δ identity in a completely different way, you can avoid *both* the initial rearrangement of the indices on the alternating symbols and the slow left-right-out-in placement of the indices. Let's suppose you want to apply the ε-δ identity to the expression $\varepsilon_{imk}\varepsilon_{pin}$. First write a "skeleton" of the identity as follows

$$\varepsilon_{imk}\varepsilon_{pin} = \delta_{??}\delta_{??} - \delta_{??}\delta_{??} \tag{3.80}$$

Our goal is to find a *rapid* and error-minimizing way to fill in the question marks with the correct index symbols. Once you have written the skeleton, look at the left-hand side to identify which index is summed. In this case, it is the index i. Next say out loud the four free indices in an order defined by "*cyclically moving forward from the summed index*" on each alternating symbol. Each alternating symbol has two free indices. To call out their names by moving cyclically forward, you simply say the name of the two indices to the *right* of the summed index, wrapping back around to the beginning if necessary. For example, the two indices cyclically forward from "*p*" in the sequence "*pqr*" are "*qr*"; the two indices cyclically forward from "*q*" are "*rp*"; the two indices forward from "*r*" are "*pq*". For the first alternating symbol in the skeleton of Eq. (3.80), the two indices cyclically forward from the summed index *i* are "*mk*" whereas the two indices cyclically forward from *i* in the second alternating symbol are "*np*". You can identify these pairs quickly without ever having to rearrange anything, and you can (in your head) group the pairs together to obtain a sequence of four free indices "*mknp*". The final step is to write these four indices onto the skeleton. If the indices are ordered 1234, then you should write the first two indices (first and second) on the skeleton like this

$$\delta_{1?}\delta_{2?} - \delta_{1?}\delta_{2?} \tag{3.81}$$

You write the last pair (third and fourth) in order (34) on the first term and in *reverse* order (43) on the last term:

$$\delta_{13}\delta_{24} - \delta_{14}\delta_{23} \tag{3.82}$$

Thus, for example, to place the free indices "*mknp*" onto the Kronecker deltas in Eq. (3.80), you would first take care of the "*mk*" by writing

$$\delta_{m?}\delta_{k?} - \delta_{m?}\delta_{k?} \tag{3.83}$$

Then you just finish off with the last two "*np*" free indices by writing them first in that order on the first term and in reverse order on the second term to obtain the final result:

$$\varepsilon_{imk}\varepsilon_{pin} = \delta_{mn}\delta_{kp} - \delta_{mp}\delta_{kn}\,. \tag{3.84}$$

This may seem a bit strange at first (especially if you are already stuck in the left-right-outer-inner mind set), but this method is *far quicker* and less error-prone. Give it a try until you become comfortable with it, and you probably won't dream of going back to your old way.

The ε-δ (E-delta) identity with multiple summed indices

Recall that the ε-δ identity is given by

$$\sum_{n=1}^{3} \varepsilon_{ijn}\varepsilon_{kln} = \delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}\,. \tag{3.85}$$

What happens if you now consider the case of two alternating symbols multiplied side-by-side with *two* indices being summed? This question is equivalent to throwing a summation around the above equation in such a manner that you add up only those terms for which $j=l$. Then

$$\begin{aligned}\sum_{j=1}^{3}\sum_{n=1}^{3} \varepsilon_{ijn}\varepsilon_{kjn} &= \sum_{j=1}^{3}(\delta_{ik}\delta_{jj} - \delta_{ij}\delta_{jk}) \\ &= \delta_{ik}(\delta_{11}+\delta_{22}+\delta_{33}) \quad - \quad (\delta_{i1}\delta_{1k}+\delta_{i2}\delta_{2k}+\delta_{i3}\delta_{3k}) \\ &= 3\delta_{ik} - \delta_{ik} \\ &= 2\delta_{ik}\end{aligned} \tag{3.86}$$

Note that we simplified the first term by noting that $\delta_{11}+\delta_{22}+\delta_{33} = 1+1+1 = 3$. The second term was simplified by noting that $\delta_{i1}\delta_{1k}+\delta_{i2}\delta_{2k}+\delta_{i3}\delta_{3k}$ will be zero if $i \neq k$ or it will equal 1 if $i=k$. Thus, it must be simply δ_{ik}.

Using similar logic, the ε-δ identity with *all indices summed* is equivalent to setting $i=k$ in the above equation, summing over each instance so that the result is *six*. To summarize using the summation conventions,

$$\boxed{\varepsilon_{ijn}\varepsilon_{kjn} = 2\delta_{ik}} \tag{3.87}$$

$$\boxed{\varepsilon_{ijk}\varepsilon_{ijk} = 6} \tag{3.88}$$

Determinant of a square matrix

The simplest way to explain what is meant by a determinant is to define it recursively. In this section, we show how the determinant of a 3×3 matrix can be alternatively defined by using the three-dimensional permutation symbol of Eq. (3.76).

A 1×1 matrix is just a single number. The determinant of a 1×1 matrix is defined to equal its solitary component. Thus,

$$\det[A_{11}] \equiv A_{11} \tag{3.89}$$

The determinant of a 2×2 matrix is defined by

$$\det\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \equiv A_{11}A_{22} - A_{12}A_{21} \tag{3.90}$$

The determinant of a 3×3 matrix is defined by

$$\det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

$$\begin{aligned} \equiv \quad & (A_{11}A_{22}A_{33} + A_{12}A_{23}A_{31} + A_{13}A_{21}A_{32}) \\ & - (A_{13}A_{22}A_{31} + A_{11}A_{23}A_{32} + A_{12}A_{21}A_{33}) \end{aligned} \tag{3.91}$$

Note that we have arranged this formula such that the first indices in each factor are 123. For the positive terms, the second indices are all the positive permutations of 123. Namely: 123, 231, and 312. For the negative terms, the second indices are all the *negative* permutations of 123. Namely: 321, 132, and 213. This relationship may be written compactly by using the permutation symbol ε_{ijk} from Eq. (3.76). Namely, if $[A]$ is a 3×3 matrix, then

$$\boxed{\det[A] = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} \varepsilon_{ijk}A_{1i}A_{2j}A_{3k}} \tag{3.92}$$

This definition can be extended to square matrices of arbitrarily large dimension by using the n-dimensional permutation symbol (see footnote on page 36). Alternatively, for square matrices of arbitrarily large dimension, the determinant can be defined recursively as

$$\boxed{\det[A]_{N\times N} = \sum_{j=1}^{N} A_{ij}A_{ij}^C} \qquad \text{(no implied summation on index } i\text{)} \tag{3.93}$$

where i is a free index taking any convenient value from 1 to N (any choice for i will give the same result). The quantity A_{ij}^C is called the "**cofactor**" of A_{ij}, and it is defined by

$$A_{ij}^C = (-1)^{i+j}\det[M_{ij}]_{(N-1)\times(N-1)} \tag{3.94}$$

Here $[M_{ij}]$ is the submatrix obtained by striking out the i^{th} row and j^{th} column of $[A]$. The determinant of $[M_{ij}]$ is called the "**minor**" associated with A_{ij}. By virtue of the $(-1)^{i+j}$, the cofactor component A_{ij}^C is often called the "**signed minor**." Further details about cofactors, including examples of how to compute them for 2×2 and 3×3 matrices are given on page 42. The formula in Eq. (3.93) is almost never used in numerical calculations because it requires too many multiplications,* but it frequently shows up in theoretical analyses.

The index i in Eq. (3.94) may be chosen for convenience (usually a row with several zeros is chosen to minimize the number of sub-determinants that must be computed). The above definition is recursive because $\det[A]_{N\times N}$ is defined in terms of smaller $(N-1)\times(N-1)$ determinants, which may in turn be expressed in terms of $(N-2)\times(N-2)$ determinants, and so on until the determinant is expressed in terms of only 1×1 determinants, for which the determinant is defined in Eq. (3.89). As an example, consider using Eq. (3.93) to compute the determinant of a 3×3 matrix. Choosing $i=1$, Eq. (3.93) gives

$$\det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = A_{11}\det\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} - A_{12}\det\begin{bmatrix} A_{21} & A_{23} \\ A_{31} & A_{33} \end{bmatrix} + A_{13}\det\begin{bmatrix} A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix}, \tag{3.95}$$

Alternatively choosing $i=2$, Eq. (3.93) gives

$$\det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = -A_{21}\det\begin{bmatrix} A_{11} & A_{13} \\ A_{32} & A_{33} \end{bmatrix} + A_{22}\det\begin{bmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{bmatrix} - A_{23}\det\begin{bmatrix} A_{11} & A_{12} \\ A_{31} & A_{32} \end{bmatrix}, \tag{3.96}$$

After using Eq. (3.90) to compute the 2×2 submatrices, both of the above expressions give the same final result as Eq. (3.91).

* Specifically, for large values of the dimension N, the number of multiplications required to evaluate the determinant using Crammer's rule (as Eq. 3.93 is sometimes called) approaches $(e-1)N!$, where e is the base of the natural logarithm. An ordinary personal computer would require a few million years to compute a 20×20 determinant using Cramer's rule! Far more efficient decomposition methods [__] can be used to compute determinants of large matrices.

Some key properties of the determinant are listed below:

$$\det([A]^T) = \det[A] \tag{3.97}$$

$$\det([A][B]) = (\det[A])(\det[B]) \tag{3.98}$$

$$\det(\alpha[A]_{N \times N}) = \alpha^N \det[A] \tag{3.99}$$

$$\det([A]^{-1}) = \frac{1}{\det[A]} \tag{3.100}$$

If $[B]$ is obtained by swapping two rows (or two columns) of $[A]$, then $\det[B] = -\det[A]$. (3.101)

If any row of $[A]$ can be written as a linear combination of the other rows, then $\det[A]=0$. A special case is that $\det[A]=0$ if any two rows of $[A]$ are equal. (3.102)

For 3×3 determinants, the last two properties allow us to generalize Eq. (3.92) to read

$$\varepsilon_{pqr}\det[A] = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} \varepsilon_{ijk}A_{pi}A_{qj}A_{rk} \tag{3.103}$$

or, using the summation convention in which repeated indices are *understood* to be summed (and, for clarity, now shown in red),

$$\varepsilon_{pqr}\det[A] = \varepsilon_{ijk}A_{pi}A_{qj}A_{rk} \tag{3.104}$$

This expression is frequently cited in continuum mechanics textbooks as the indicial definition of the determinant of a 3×3 matrix. Multiplying the above formula by ε_{pqr} and summing over p q and r (and using Eq. 3.88) reveals that

$$\det[A] = \frac{1}{6}\varepsilon_{pqr}A_{pi}A_{qj}A_{rk}\varepsilon_{ijk} \tag{3.105}$$

Here, there are implied summations over the indices *i,j,k,p,q,* and *r*. If it were expanded out, the above expression would contain 729 terms, so it is obviously not used to actually compute the determinant. However, it is not at all uncommon for expressions like this to show up in analytical analysis, and it is therefore essential for the analyst to recognize that the right-hand-side simplifies so compactly.

Incidentally, note that

$$\varepsilon_{ijk}\varepsilon_{pqr} = \det\begin{bmatrix} \delta_{ip} & \delta_{iq} & \delta_{ir} \\ \delta_{jp} & \delta_{jq} & \delta_{jr} \\ \delta_{kp} & \delta_{kq} & \delta_{kr} \end{bmatrix} \tag{3.106}$$

If any two columns or rows are interchanged, the sign of this determinant is reversed, consistent with the cyclic property of the permutation symbol. Furthermore, if any two indices of ijk are equal, then two rows will be identical, making the determinant zero, again consistent with the definition of the permutation symbol. If, on the other hand, one of the three ijk indices is set equal to one of the three pqr indices, and the common index is summed from 1 to 3, then the ε-δ identity is recovered.

More about cofactors

Given a matrix $[A]_{N \times N}$, the **cofactor** matrix $[A]^C$ is an $N \times N$ matrix whose components are obtained by application of the very strange formula that we first introduced in Eq. (3.94):

$$A_{ij}^C = (-1)^{i+j} \det[M_{ij}]_{(N-1) \times (N-1)} \tag{3.107}$$

where (recall) $[M_{ij}]$ is the submatrix obtained by striking out the i^{th} row and j^{th} column of $[A]$. The determinant of $[M_{ij}]$ is called the "**minor**" associated with A_{ij}. The **cofactor** A_{ij}^C simply equals the minor times $(-1)^{i+j}$.

It might seem that something with such an awkward definition would be of only limited usefulness, but it turns out that cofactors are fantastically convenient and they appear frequently in practical physics applications (unless trained to look for cofactors, however, many people don't recognize them when they appear). As a rule of thumb: whenever you are analyzing a problem in which things are fluxing across or acting upon area elements that are stretching and rotating in time, "think cofactors."

The remainder of this section illustrates how to compute cofactors in 2 and 3 dimensions, finishing with some derivative formulas involving cofactors.

SIDEBAR: cofactors of a 2×2 matrix

Consider a 2×2 matrix, $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$. To find the minor associated with A_{11}, you strike out row 1 and column 1, and take the determinant of what's left. For a 2×2 matrix, the only thing left is pretty trivial — it's just the A_{22} component. The determinant of a single number is just that number, so the minor associated with A_{11} is just A_{22}. The *cofactor*, which is the *signed* minor found by multiplying by $(-1)^{1+1} = 1$. Thus, $A_{11}^C = A_{22}$ for a 2×2 matrix. The minor associated with A_{12} is found by striking out row 1 and column 2 and taking the determinant of what's left which is just A_{21}. The *cofactor* is found by multiplying this minor by $(-1)^{1+2} = -1$ to obtain $A_{12}^C = -A_{21}$. Proceeding onward for each component eventually gives

$$\begin{aligned} A_{11}^C &= A_{22} \qquad & A_{12}^C &= -A_{12} \\ A_{21}^C &= -A_{12} \qquad & A_{22}^C &= A_{11} \end{aligned} \qquad \textit{These apply to a } \mathbf{2 \times 2} \textit{ matrix only!} \tag{3.108}$$

SIDEBAR: cofactors of a 3×3 matrix

Consider a 2×2 matrix, $\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$. To find the minor associated with A_{11}, you strike out row 1 and column 1, and take the determinant of what's left, namely the 2×2 matrix $\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix}$, which has determinant $A_{22}A_{33} - A_{32}A_{23}$. The *cofactor*, which is the *signed* minor, is found by multiplying this result by $(-1)^{1+1} = 1$. Thus, $A^C_{11} = A_{22}A_{33} - A_{32}A_{23}$. For index structure pattern recognition (discussed below), we will write this result in an equivalent form with the factors in the last term swapped so that $A^C_{11} = A_{22}A_{33} - A_{23}A_{32}$. The minor associated with A_{12} is found by striking out row 1 and column 2 and taking the determinant of the remaining 2×2 matrix $\begin{bmatrix} A_{21} & A_{23} \\ A_{31} & A_{33} \end{bmatrix}$ to obtain $A_{21}A_{33} - A_{31}A_{23}$. The *cofactor* is found by multiplying this minor by $(-1)^{1+2} = -1$ to obtain $A^C_{12} = A_{31}A_{23} - A_{21}A_{33}$, which we will write equivalently as $A^C_{12} = A_{23}A_{31} - A_{21}A_{33}$. Proceeding onward for each cofactor eventually gives

$$\begin{array}{lll} A^C_{11} = A_{22}A_{33} - A_{23}A_{32} & A^C_{12} = A_{23}A_{31} - A_{21}A_{33} & A^C_{13} = A_{21}A_{32} - A_{22}A_{31} \\ A^C_{21} = A_{32}A_{13} - A_{33}A_{12} & A^C_{22} = A_{33}A_{11} - A_{31}A_{13} & A^C_{23} = A_{31}A_{12} - A_{32}A_{11} \\ A^C_{31} = A_{12}A_{23} - A_{13}A_{22} & A^C_{32} = A_{13}A_{21} - A_{11}A_{23} & A^C_{33} = A_{11}A_{22} - A_{12}A_{21} \end{array} \tag{3.109}$$

These apply to a 3×3 matrix only!

The index structure of these cofactors can be expressed in the form

$$A^C_{mn} = A_{\overline{m}\,\overline{n}}A_{\underline{m}\,\underline{n}} - A_{\overline{m}\,\underline{n}}A_{\underline{m}\,\overline{n}} \qquad \textit{Applies to a } 3 \times 3 \textit{ matrix only!} \tag{3.110}$$

Here, an overbar on an index indicates a cyclic step forward and underbar indicates a cyclic step backward. Specifically,

$$\begin{array}{l} \text{if } m = 1\text{, then } \overline{m}=2 \text{ and } \underline{m}=3 \\ \text{if } m = 2\text{, then } \overline{m}=3 \text{ and } \underline{m}=1 \\ \text{if } m = 3\text{, then } \overline{m}=1 \text{ and } \underline{m}=2 \end{array} \tag{3.111}$$

Finally, for 3×3 matrices, cofactor components may be expressed in terms of the permutation symbol as follows

$$A^C_{ij} = \tfrac{1}{2}\varepsilon_{ipr}\varepsilon_{jqs}A_{pq}A_{rs}, \tag{3.112}$$

where, for compactness, we have used implicit summation conventions.

Cofactor-inverse relationship. The cofactor matrix $[A]^C$ is well-defined even $[A]$ is singular. If, however, $[A]$ happens to be invertible, then

$$[A]^C = (\det[A])[A]^{-T} \tag{3.113}$$

This identity will be later discussed in more detail in the tensor analysis chapters.

Derivative of the cofactor. For a 3×3 matrix [A], differentiating Eq. (3.112) gives

$$\frac{\partial A^C_{ij}}{\partial A_{rs}} = \sum_{m=1}^{3} \sum_{n=1}^{3} \varepsilon_{irm} A_{mn} \varepsilon_{njs} \tag{3.114}$$

Derivative of a determinant (IMPORTANT)

The determinant of a matrix $[A]$ is computed from the components of $[A]$. Hence, the determinant may be regarded as a *function* of these components. Here we want to compute the derivatives of $\det[A]$ with respect to any individual component of $[A]$, while holding the other components constant.

Consider, for example, a 2×2 determinant. Derivatives of Eq. (3.90) with respect to individual components gives.

$$\frac{\partial(\det[A]_{2\times 2})}{\partial A_{11}} = A_{22} \qquad \frac{\partial(\det[A]_{2\times 2})}{\partial A_{12}} = -A_{21}$$
$$\frac{\partial(\det[A]_{2\times 2})}{\partial A_{21}} = -A_{12} \qquad \frac{\partial(\det[A]_{2\times 2})}{\partial A_{22}} = A_{11} \tag{3.115}$$

The formulas are different for a 3×3 matrix. Specifically, differentiating Eq. (3.91) with respect to each individual component gives

$$\frac{\partial(\det[A]_{3\times 3})}{\partial A_{11}} = A_{22}A_{33} - A_{23}A_{32}\,, \qquad \frac{\partial(\det[A]_{3\times 3})}{\partial A_{12}} = A_{23}A_{31} - A_{21}A_{33}\,, \quad \text{etc.} \tag{3.116}$$

Comparing Eq. (3.115) with (3.108) and comparing (3.116) with (3.110) show that, *regardless of the size of the matrix*, the derivative of a determinant with respect to a matrix is given by the cofactor!

$$\frac{\partial(\det[A])}{\partial A_{ij}} = A^C_{ij} \tag{3.117}$$

Frequently, the structure $|A|$ is used as a streamlined alternative to the notation $\det[A]$, in which case this result would be written

$$\boxed{\frac{\partial |A|}{\partial A_{ij}} = A^C_{ij}}\text{, where } |A| \equiv \det[A] \tag{3.118}$$

Though we have only proved it for the special cases of 2×2 and 3×3 matrices, this result holds regardless of the dimension of [A]. This result is valid even if the matrix $[A]$ is singular. If, however, $[A]$ happens to be invertible, then the above result may be written

$$\boxed{\frac{\partial |A|}{\partial A_{ij}} = |A| A^{-T}_{ij}}\text{, when } [A] \text{ is invertible} \tag{3.119}$$

Here, A_{ij}^{-T} denotes the ij components of the inverse of the transpose (or, equivalently, the transpose of the inverse) of $[A]$. Inverses and transposes are discussed soon.

Rates of determinants. Whenever you need the derivative of a determinant with respect to something *other* than an individual component of the matrix, you should still nevertheless *start* with Eq. (3.118), and then apply the chain rule. In continuum mechanics, for example, material motion is characterized through the use of a tensor called the deformation gradient, which has a component matrix denoted $[F]$. The "Jacobian" is given by $J = \det[F]$ and, physically, it represents the ratio of deformed to undeformed volume of a material element. For dynamic problems, it is often desirable to know the time derivative of the Jacobian because it characterizes the volumetric expansion or contraction rate. If J is a function of time t, then it follows that the components of $[F]$ must be functions of time, so we can apply the chain rule of partial differentiation.

$$\frac{dJ}{dt} = \frac{\partial J}{\partial F_{11}}\frac{dF_{11}}{dt} + \frac{\partial J}{\partial F_{12}}\frac{dF_{12}}{dt} + \ldots + \frac{\partial J}{\partial F_{33}}\frac{dF_{33}}{dt} \tag{3.120}$$

We now know that the derivative of a determinant with respect to a component is given by the cofactor for that component. Hence, this may be written

$$\frac{dJ}{dt} = F_{11}^C\frac{dF_{11}}{dt} + F_{12}^C\frac{dF_{12}}{dt} + \ldots + F_{33}^C\frac{dF_{33}}{dt} \tag{3.121}$$

or, employing the popular alternative notational structure in which a superposed single dot denotes time differentiation,

$$\dot{J} = F_{11}^C\dot{F}_{11} + F_{12}^C\dot{F}_{12} + \ldots + F_{33}^C\dot{F}_{33} \tag{3.122}$$

or

$$\dot{J} = \sum_{i=1}^{3}\sum_{j=1}^{3} F_{ij}^C\dot{F}_{ij} \tag{3.123}$$

Recognizing that this expression is the matrix inner product, we may write

$$\dot{J} = [F]^C * [\dot{F}] \qquad \text{(or, in tensor notation, } \dot{J} = \underset{\approx}{F}^C : \dot{\underset{\approx}{F}}\text{)} \tag{3.124}$$

If $[F]$ is invertible (which it indeed is for deformation gradients), then this result may be written

$$\dot{J} = J\sum_{i=1}^{3}\sum_{j=1}^{3} F_{ij}^{-T}\dot{F}_{ij} = J\sum_{i=1}^{3}\sum_{j=1}^{3} F_{ji}^{-1}\dot{F}_{ij} = J\sum_{i=1}^{3}\sum_{j=1}^{3} \dot{F}_{ij}F_{ji}^{-1} \tag{3.125}$$

or

$$\dot{J} = J\mathrm{tr}([\dot{F}][F]^{-1}) \qquad \text{(or, in tensor notation, } \dot{J} = J\ \mathrm{tr}(\dot{\underset{\approx}{F}} : \underset{\approx}{F}^{-1})\text{)} \tag{3.126}$$

This result holds for any matrix $[F]$ even though we have here described it using the deformation gradient as an example.

Derivatives of determinants with respect to vectors. Suppose that $J = \det[F]$ is to be differentiated with respect to an *array* $\{v\}$. Again, Eq. (3.118) is the place to start in conjunction with the chain rule.

$$\frac{\partial J}{\partial v_k} = \frac{\partial J}{\partial F_{ij}}\frac{\partial F_{ij}}{\partial v_k} = F_{ij}^C\frac{\partial F_{ij}}{\partial v_k} \tag{3.127}$$

where, we are here using implicit summation over the indices i and j. If $[F]$ is invertible, then this result may be written

$$\frac{\partial J}{\partial v_k} = J\ F_{ij}^{-T}\frac{\partial F_{ij}}{\partial v_k} \tag{3.128}$$

Principal sub-matrices and principal minors

A so-called $n \times n$ principal submatrix of a square matrix $[A]_{N\times N}$ is any $n \times n$ submatrix (where $n \le N$) whose diagonal components are *also* diagonal components of the larger matrix. For example,

$$\begin{bmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{bmatrix} \tag{3.129}$$

is a principal submatrix, whereas $\begin{bmatrix} A_{12} & A_{13} \\ A_{22} & A_{23} \end{bmatrix}$ is *not* a principal submatrix. For a 3×3 matrix, there are *three* 1×1 principal submatrices (identically equal to the diagonal components), *three* 2×2 principal submatrices, and only *one* 3×3 principal submatrix (equal to the matrix $[A]$ itself).

A sequence of 1×1, 2×2, … $N \times N$ submatrices is **nested** if the 1×1 matrix is a submatrix of the 2×2 matrix, *and* the 2×2 matrix is a submatrix of the next larger submatrix, and so forth.

A **principal minor** is the determinant of any principal submatrix. The term "*nested* minors" means the determinants of a set of nested submatrices.

Matrix invariants

The k^{th} "characteristic" invariant, denoted I_k, of a matrix $[A]$ is the sum of all possible $k \times k$ principal minors. For a 3×3 matrix, these three invariants are

$$I_1 = A_{11} + A_{22} + A_{33} \tag{3.130a}$$

$$I_2 = \det\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} + \det\begin{bmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{bmatrix} + \det\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} \tag{3.130b}$$

$$I_3 = \det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{3.130c}$$

Warning: if the matrix is non-symmetric, the characteristic invariants are not a *complete* set of independent invariants. If all three characteristic invariants of a *symmetric* matrix are zero, then the matrix itself is zero. However, as discussed later, it is possible for all three characteristic invariants of a non-symmetric matrix to be zero *without* the matrix itself being zero.

Derivatives: (without proof)

$$\frac{\partial I_1}{\partial A_{mn}} = \delta_{mn} \tag{3.131}$$

$$\frac{\partial I_2}{\partial A_{mn}} = I_1\delta_{mn} - A_{nm} = I_1\delta_{mn} - A^T_{mn} \tag{3.132}$$

$$\frac{\partial I_3}{\partial A_{mn}} = A^C_{mn} \tag{3.133}$$

Alternative invariant sets. As will be discussed throughout this book, the above three invariants are not the *only* invariants. Check "invariant" in the index for details.

Positive definite

A square matrix $[B]_{N \times N}$ is positive definite if and only if

$$\{v\}^T[B]\{v\} > 0 \qquad \text{for all } \{v\} \tag{3.134}$$

In indicial notation, this requirement is

$$\sum_{i=1}^{N}\sum_{j=1}^{N} v_i B_{ij} v_j > 0 \tag{3.135}$$

Written out explicitly for the special case of a 2×2 matrix

$$B_{11}v_1v_1 + B_{12}v_1v_2 + B_{21}v_2v_1 + B_{22}v_2v_2 > 0 \tag{3.136}$$

Note that the middle two terms can be combined and written as $2\left(\frac{B_{12}+B_{21}}{2}\right)v_2v_1$. Similarly, you can write the first term as $2\left(\frac{B_{11}+B_{11}}{2}\right)$. The third term can also be so written. Thus, the requirement for positive definiteness depends only on the *symmetric part* of the matrix $[B]$. The non-symmetric part has no influence on whether or not a matrix is positive definite. Consequently, you may replace Eq. (3.134) by the equivalent, but more carefully crafted, statement:

$[B]$ is positive definite if and only if $\{v\}^T[A]\{v\} > 0$ for all $\{v\}$, where $[A]$ is the symmetric part of $[B]$.

It can be shown that a matrix is positive definite if and only if the characteristic invariants *of the symmetric part* of the matrix are all positive.*

Fortunately, there is an *even simpler* test for positive definiteness: you only have to verify that any *nested* set of principal minors are all positive! This calculation is easier than finding the invariants themselves because it requires evaluation of only one principal minor determinant of each size (you don't have to evaluate *all* of them). See page __ for further details.

The cofactor-determinant connection

Let $[A]^C$ denote the matrix of **cofactors** A_{ij}^C associated with a square matrix $[A]_{N\times N}$. The *transpose* of the cofactor matrix is also sometimes called the **adjugate matrix** (not to be confused with "adjoint"). Recall the definition of the cofactor given in Eq. (3.94):

$$A_{ij}^C = (-1)^{i+j}\det[M_{ij}]_{(N-1)\times(N-1)} \tag{3.137}$$

By virtue of Eq. (3.97), note that the transpose of the cofactor matrix is identically equal to the cofactor matrix associated with $[A]^T$. In other words, the cofactor and transpose operations commute:

$$([A]^C)^T = ([A]^T)^C \tag{3.138}$$

As a short hand, we generally eliminate the parentheses and simply write $[A]^{CT}$ to mean the transpose of the cofactor (or, equivalently, the cofactor of the transpose). The generalization of Eq. (3.93) is

* It is possible to construct a matrix that has all positive invariants, but whose symmetric part does *not* have all positive invariants.

$$\sum_{k=1}^{N} A_{ik}A_{jk}^{C} = \begin{cases} 0 & \text{if} \quad i \neq j \\ \det[A] & \text{if} \quad i=j \end{cases} \tag{3.139}$$

Written more compactly,

$$\sum_{k=1}^{N} A_{ik}A_{jk}^{C} = \det[A]\ \delta_{ij} \tag{3.140}$$

Written in matrix form,

$$[A][A]^{CT} = (\det[A])\ [I] \tag{3.141}$$

It turns out that the location of the transpose and cofactor operations is inconsequential — the result will be the same in all cases. Namely,

$$\boxed{[A][A]^{CT} = [A]^{C}[A]^{T} = [A]^{T}[A]^{C} = [A]^{CT}[A] = (\det[A])\ [I]} \tag{3.142}$$

Inverse

The inverse of a matrix $[A]$ is the matrix denoted $[A]^{-1}$ for which

$$[A][A]^{-1} = [A]^{-1}[A] = [I] \tag{3.143}$$

If the inverse exists, then it is unique. If the inverse does *not* exist, then the matrix $[A]$ is said to be “non-invertible” or “singular.” A necessary and sufficient condition for the inverse to exist is that the determinant must be nonzero:

$$\det[A] \neq 0 \tag{3.144}$$

Comparing Eqs. (3.142) and (3.143), note that the inverse may be readily computed from the cofactor by

$$[A]^{-1} = \frac{[A]^{CT}}{\det[A]} \tag{3.145}$$

While this definition does uniquely define the inverse, it must never be used as a definition of the cofactor matrix. The cofactor matrix is well-defined and generally nonzero even if the matrix $[A]$ is singular.

Eigenvalues and eigenvectors

As mentioned in Eq. (3.92), a nonzero vector (array) $\{p\}$ is called an eigenvector of a square matrix $[A]$ if there exists a scalar λ, called the eigenvalue, such that $[A]\{p\} = \lambda\{p\}$. In order for this equation to have a non-trivial (nonzero) solution, the determinant of the matrix $[A] - \lambda[I]$ must be zero. Setting this determinant to zero results in a *polynomial* equation, called the characteristic equation, for λ. If $[A]$ is a 2×2 matrix, the equation will be quadratic. If $[A]$ is a 3×3 matrix, the equation will be cubic, and so forth. We highly recommend that you do *not* construct the matrix $[A] - \lambda[I]$ and then set its determinant equal to zero. While that would certainly work, it allows for too many opportunities to make an arithmetic error. Instead, the fastest way to generate the characteristic equation is to *first* find all of the characteristic invariants of $[A]$. These invariants are the coefficients in the characteristic equation, alternating sign, as follows

For $[A]_{2\times2}$, the characteristic equation is

$$\lambda^2 - I_1\lambda + I_2 = 0,$$

$$\text{where } I_1 = A_{11} + A_{22}\text{, and } I_2 = \det\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{3.146}$$

For $[A]_{3\times3}$, the characteristic equation is

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0,$$

$$\text{where } I_1 = A_{11} + A_{22} + A_{33},$$

$$I_2 = \det\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} + \det\begin{bmatrix} A_{11} & A_{13} \\ A_{31} & A_{33} \end{bmatrix} + \det\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix}\text{, and}$$

$$I_3 = \det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{3.147}$$

For $[A]_{4\times4}$, the characteristic equation is $\lambda^4 - I_1\lambda^3 + I_2\lambda^2 - I_3\lambda + I_4 = 0$. Higher dimension matrices are similar.

Because the characteristic equation is a polynomial equation, an $N \times N$ matrix will have up to N possible eigenvalues. For each solution λ_i there exists at least one corresponding eigenvector $\{p\}_i$, which is determined by solving

$$[A]\{p\}_i = \lambda_i\{p\}_i \qquad \textit{(no sum on } i\textit{)}. \tag{3.148}$$

The solution for $\{p\}_i$ will have an undetermined magnitude and, for symmetric matrices, it is conventional to set the magnitude to one. For non-symmetric matrices, however, the normalization convention is different, as discussed in relation to Eq. (15.44).

If an eigenvalue λ_i has algebraic multiplicity m (i.e., if the characteristic equation gives a root λ_i repeated m times), then there can be no more than a total of m independent eigenvectors associated with that eigenvalue — there might be fewer (though there is always *at least* one). If the matrix $[A]$ is symmetric, then it is well known [1] that it is *always* possible to find m independent eigenvectors. The directions of the eigenvectors when the multiplicity m is greater than one are arbitrary. However, the one thing that *is* unique is the *span* of these vectors (see page 17), and it is conventional to set the eigenvectors to any orthonormal set of vectors lying in the span. For *non-symmetric* matrices, it might happen that an eigenvalue of multiplicity m corresponds to a total of $\mu < m$ linearly independent eigenvectors, where μ is called the *geometric multiplicity*. For example, the matrix

$$\begin{bmatrix} 5 & 3 \\ 0 & 5 \end{bmatrix} \tag{3.149}$$

Has an eigenvalue $\lambda = 5$ with algebraic multiplicity of two. To find the associate eigenvector(s), you must solve

$$\begin{bmatrix} 5 & 3 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = 5 \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{3.150}$$

Multiplying this out gives

$$5p_1 + 3p_2 = 5p_1 \tag{3.151}$$

$$5p_2 = 5p_2 \tag{3.152}$$

The second equation gives us no information, and the first equation gives the constraint that $p_2 = 0$. Therefore, even though the eigenvalue had algebraic multiplicity of *two*, you have only *one* eigenvector (geometric multiplicity equals one) which is given by $\{1, 0\}$. When the geometric multiplicity of an eigenvector is less than the algebraic multiplicity, then there does still exist a subspace that is uniquely associated with the multiple eigenvalue. However, characterizing this subspace requires solving a “generalized eigenproblem” to construct additional vectors that will combine with the one or more ordinary eigenvectors to form a set of vectors that span the space. The process for doing this is onerous, and we have not yet *personally* happened upon any engineering application for which finding these generalized eigenvectors provides any useful information, so we will not cover the details. Instructions for the process can be found in [23,22,26]. If the generalized eigenvectors are truly sought, then they can be found via the “`JordanDecomposition`” command in *Mathematica* [27] (see discussion below to interpret the result).

Similarity transformations. Suppose that you have a set of eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ for a matrix $[A]$, possibly with some of these eigenvalues having algebraic multiplicities greater than one. Let $[L]$ denote the matrix whose columns contain the corresponding eigenvectors (augmented, where necessary, to include *generalized* eigenvectors for the cases where the geometric multiplicity is less than the algebraic multiplicity; the ordinary eigenvectors corresponding to a given eigenvalue should always, by convention, be entered into columns of $[L]$ *before* the generalized eigenvectors). Then it can be shown that the original matrix $[A]$ satisfies the similarity transformation

$$[A] = [L][\Lambda][L]^{-1} \tag{3.153}$$

If there are no generalized eigenvectors contained in the matrix $[L]$, then the matrix $[\Lambda]$ is diagonal, with the diagonal components being equal to the eigenvalues. In this case, the original matrix $[A]$ is said to be “diagonalizable.” If, on the other hand, $[L]$ contains any generalized eigenvectors, then $[\Lambda]$ still contains the eigenvalues on the diagonal, but it additionally will contain a "1" in the $k-1, k$ position (i.e., a "1" just above the eigenvalue in the k^{th} column) corresponding to each k^{th} generalized eigenvector. In this form, the matrix $[\Lambda]$ is said to be in **Jordan canonical form**. For example, the similarity transformation corresponding to Eq. (3.149) is

$$\begin{bmatrix}5&3\\0&5\end{bmatrix} = \begin{bmatrix}1&0\\0&1/3\end{bmatrix}\begin{bmatrix}5&1\\0&5\end{bmatrix}\begin{bmatrix}1&0\\0&1/3\end{bmatrix}^{-1} \qquad (3.154)$$

This result can be obtained in *Mathematica* [27] via the command **`JordanDecomposition[{{5,3},{0,5}}]`**. The presence of the "1" in the 12 position of the $[\Lambda]$ matrix implies that the second column of $[L]$ must contain a generalized eigenvector.

A matrix $[A]$ is "**diagonalizable**" if the $[\Lambda]$ matrix has no "1s" on any off-diagonal. Consequently, there are no generalized eigenvectors. In general, even though $[\Lambda]$ is diagonal, that does *not* mean that $[A]$ is symmetric.

The matrix $[L]$ will be **orthogonal** (i.e., $[L]^{-1} = [L]^T$) if *and only if* the original matrix $[A]$ is symmetric. For symmetric matrices, there will *never* be any generalized eigenvectors (i.e., the algebraic and geometric eigenvalue multiplicities will always be equal), and the $[\Lambda]$ matrix will therefore *always* be fully diagonal (no "1" on any off-diagonal).

Finding eigenvectors by using the adjugate

Recall that $\{p\}$ is an eigenvector of $[A]$ if $[A]\{p\} = \lambda\{p\}$, where λ is the eigenvalue. This equation can be written

$$[B]\{p\} = 0\text{, where } [B] = [A]-\lambda[I] \qquad (3.155)$$

Recall that we determined the eigenvalue by setting $\det[B] = 0$. You can likewise use the $[B]$ matrix to determine the eigenvectors. Given that the determinant of $[B]$ is zero, Eq. (3.142) tells us that

$$[B][B]^{CT} = 0, \qquad (3.156)$$

from which it follows that *if there is any nonzero column of the adjugate matrix,* $[B]^{CT}$*, then that column must be an eigenvector of* $[B]$ *associated with the eigenvalue that was used to construct* $[B]$ *in Eq. (3.155).* It turns out that, for *distinct* eigenvalues (i.e., ones with multiplicity of 1), there will always be a non-zero column of $[B]^{CT}$, and therefore the adjugate matrix is a slick and easy way to find the eigenvector. Unfortunately, the situation is not so great when the eigenvalue is a double (or higher) root. For eigenvalues of multiplicity greater than one, it is still true that any nonzero column of $[B]^{CT}$ will be an eigenvector, but the adjugate might turn out to be zero or it might not capture *all* of the possible eigenvectors. Consider, for example,

$$[A] = \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix} \qquad (3.157)$$

For the eigenvalue $\lambda = 1$, the "B" matrix is

$[B] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$ for which the adjugate is $[B]^{CT} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, and therefore the

eigenvector is the only nonzero column, $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ (3.158)

However, for the double root eigenvalue, $\lambda = 0$, the "B" matrix is just $[A]$ itself, and the adjugate is the zero matrix. Lacking nonzero columns, this adjugate matrix gives us no guidance to determine the eigenvector, and traditional methods must instead be used. Our recommendation is: use the adjugate to compute eigenvectors associated with *distinct non-multiple* eigenvalues, but don't bother when the eigenvalue is a multiple root.

Eigenprojectors

Recall Eq. (3.153):

$$[A] = [L][\Lambda][L]^{-1} \tag{3.159}$$

Let's consider only the case that $[A]$ is diagonalizable so that $[\Lambda]$ is diagonal. Recall that each column of $[L]$ contains an eigenvector. Let $\{g\}_k$, called the left-eigenvector, denote the k^{th} column of $[L]$ (i.e., the k^{th} eigenvector). Let $<G>_k$, called the right-eigenvector, denote the k^{th} *row* of $[L]^{-1}$. Since $[L]^{-1}[L] = [I]$, it follows that

$$<G>_i\{g\}_j = \delta_{ij} \tag{3.160}$$

This equation states that the inner product of *different* left and right eigenvectors will be zero and the inner product of corresponding left and right eigenvectors will equal unity. This property is called **duality**.

For the moment, consider the case that all eigenvalues are distinct. The *outer* product of the left eigenvector $\{g\}_k$ with its own counterpart right eigenvector $<G>_k$ will produce a square matrix, which we will denote $[P]_k$ and refer to as the k^{th} eigenprojector:

$$[P]_k \equiv \{g\}_k<G>_k \qquad \text{(no sum on } k\text{)} \tag{3.161}$$

An easy way to compute the eigenprojector $[P]_k$ is

$$[P]_k = [L][\zeta]_k[L]^{-1} \tag{3.162}$$

where $[\zeta]_k$ is a matrix that has a 1 in the kk component and zeros everywhere else. Note that

$$[\zeta]_k[\zeta]_k = [\zeta]_k \qquad \text{(no sum on } k\text{)} \tag{3.163}$$

Consequently,

$$[P]_k[P]_k = [P]_k \qquad \text{(no sum on } k\text{)} \tag{3.164}$$

As will be discussed in great detail later, this property implies that the tensor $[P]_k$ is a projection operator. It can be shown that the original tensor $[A]$ can be expressed as a sum of its eigenvalues λ_k times the corresponding eigenprojectors $[P]_k$:

$$[A] = \sum_k \lambda_k [P]_k \qquad (3.165)$$

The summation ranges over the number of eigenvalues. The importance of this result will be far more clear when we get into tensor analysis. The term "eigenprojector" is used because each matrix $[P]_k$ can be regarded as a special operator that will project an arbitrary array $\{v\}$ into its part in the direction of the k^{th} eigenvector. The projection is oblique if the original matrix $[A]$ is non-symmetric. All of this will be more clear after you read about projection tensors later on.

Recall that we temporarily discussed eigenprojectors under the assumption that the eigenvalues were all distinct. When there are repeated eigenvalues, the same results still apply except that the index k in Eqs. (3.162) through (3.165) now ranges over the number of *distinct* eigenvalues and the tensor $[\zeta]_k$ is now zeros everywhere except a 1 in each location occupied by the k^{th} eigenvalue in the $[\Lambda]$ matrix. Thus, for example, a double root will have two entries of 1 on the diagonal. When there are double root eigenvalues, the associated eigenvectors are not unique, but their *span* is a unique plane. In this double root case, $[P]_k$ represents an operation that will project any vector onto that plane. Even though the eigenvectors are not unique, the eigenprojector *is* unique.

Finding eigenprojectors without finding eigenvectors. The eigenprojectors are truly the physically significant results of an eigenvalue analysis because they are unique (individual eigenvectors are not unique when there are repeated eigenvalues). You can construct the eigenprojector without ever having to obtain the eigenvectors by the following formula:

$$[P]_k = \frac{\displaystyle\prod_{\substack{i=1\\ i\neq k}}^{m} ([A] - \lambda_i [I])}{\displaystyle\prod_{\substack{i=1\\ i\neq k}}^{m} (\lambda_k - \lambda_i)} \qquad (3.166)$$

Here, m is the total number of *distinct* eigenvalues. For matrices of small dimensions, this formula can be quite useful because it eliminates the need to compute eigenvectors.

"When two men in business always agree, one of them is unnecessary." — *William Wrigler, Jr.*

4. Vector/tensor notation

"Ordinary" engineering vectors

We use the term "ordinary" to indicate the notion of a vector as it is introduced in elementary calculus texts — namely, something in 3D physical space that has length and orientation. Physical examples include velocity, electric field, and angular rotation. This "definition" is horribly ambiguous, but it is adequate for reviewing basic concepts and we promise to atone for this transgression later. In general, we use the term "engineering" whenever a quantity is defined from a physical rather than mathematical perspective. A more correct *mathematician's* definition of the term "vector" can be found on page 227, where we will also introduce **abstract vectors** in higher-dimensional spaces.

Engineering "laboratory" base vectors

Let $\{\underset{\sim}{\mathbf{e}}_1, \underset{\sim}{\mathbf{e}}_2, \underset{\sim}{\mathbf{e}}_3\}$ denote a set of mutually perpendicular vectors of unit magnitude, which we will refer to as the "**laboratory basis**" or the "**laboratory triad**." Elementary calculus books often denote these vectors by $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$, but practicing researchers just can't spare that many letters of the alphabet. The unit base vectors are always presumed to exist as a reference from which all directional quantities may be described. The orientation of the laboratory triad should be regarded as mutually agreed upon by all observers (e.g., all observers might agree that they point in the same directions as the edges of the walls and floor in one corner of their laboratory, or they might define the directions based on the stars — it doesn't really matter what the choice is, so long as everyone agrees on it). The laboratory base vectors are presumed to be fixed in time and space.

Other choices for the base vectors

We insist that the laboratory basis must exist and all observers must agree upon its definition. However, other bases can be used as well. Different observers can, if they wish, opt to use *supplemental* basis triads. All observers will be able to communicate effectively with each other by transforming their own results into equivalent results expressed using the commonly agreed-upon lab basis. The rules governing how to accomplish these transformations from one basis to another are central to the study of vector and tensor analysis.

When explaining the meanings of various vector and tensor operations, this book limits its scope to definitions that apply only to mutually perpendicular vectors, each of unit magnitude. We also require that the orthonormal triad be "right-handed" (i.e., crossing the first lab base vector into the second one gives the third one*). Any basis that is both right-handed and orthonormal is called **regularized**. All vector and tensor operations presented in this book apply only to regularized bases; generalized definitions that apply to irregular and curvilinear bases are provided in a sequel book [6].

Basis expansion of a vector

Let $\underset{\sim}{\boldsymbol{v}}$ be an ordinary vector (like velocity or electric field). Its expansion in terms of an orthonormal basis $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ is

$$\underset{\sim}{\boldsymbol{v}} = v_1\underset{\sim}{\boldsymbol{e}}_1 + v_2\underset{\sim}{\boldsymbol{e}}_2 + v_3\underset{\sim}{\boldsymbol{e}}_3 = \sum_{i=1}^{3} v_i\underset{\sim}{\boldsymbol{e}}_i \tag{4.1}$$

where v_i are called the **components** of the vector with respect to the basis. The zero vector $\underset{\sim}{\boldsymbol{0}}$ is defined to be the vector whose components are all zero.

We may arrange the components of a vector in a column matrix as

$$\{v\} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{4.2}$$

and the expansion of Eq. (4.1) is analogous to writing

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = v_1\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + v_2\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + v_3\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.3}$$

We may alternatively arrange the components of a vector into a row matrix:

$$<v> = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \tag{4.4}$$

The row representation is just the transpose of the column representation:

$$\{v\}^T = <v>, \tag{4.5}$$

However there is no such thing as the transpose of a vector — $\underset{\sim}{\boldsymbol{v}}^T$ is meaningless! If $\{v\}$ are the components of a vector collected into a 3×1 matrix, then $\{v\}^T$ also represents the *same vector* even though the matrix $\{v\}^T$ happens to be 1×3.

* By this, we mean that you can point your index and middle finger of your right hand in the direction of the first and second base vectors, respectively, and your right thumb will then point in the direction of the third base vector.

Summation convention — details

The expression in Eq. (4.1) is written more compactly by using Einstein's summation convention as

$$\underset{\sim}{\mathbf{v}} = v_i \underset{\sim}{\mathbf{e}}_i \tag{4.6}$$

The summation convention is defined such that any index (in this case *i*) that is repeated exactly twice in a term is understood to be summed from 1 to 3. The summation ranges up to 3 because ordinary engineering vectors are always referenced to 3D physical space.

Later on, we will define quantities (like matrix components A_{ij}) that have more than one index. Then, for example, the expression $A_{ij}v_j$, for which the index j is repeated, would mean the same thing as

$$\sum_{j=1}^{3} A_{ij}v_j \tag{4.7}$$

In this expression, note that the index i occurs exactly once and is not repeated. Therefore the above expression is actually *three* expressions, corresponding to the index i taking the values from 1 to 3. For rectangular Cartesian components, the summation convention has two fundamental rules (extra rules that apply for irregular bases can be found in Ref. [6] and some less-common special-situation rules are discussed later):

1. An index that occurs *exactly* twice in a term is called a **dummy index**, and it is understood to be summed from 1 to 3, with the implied summation symbol applied *only to the term in which the repeated dummy index appears.*
2. An index that occurs *exactly* once in a term is called a **free index**, and it must also appear *exactly once in every other term*.

The following expressions *violate* the summation convention:

$$\begin{aligned} &a_i + b_j && \text{(violates rule 2)} \\ &a_i A_{ij} b_i && \text{(violates rule 1)} \end{aligned} \tag{4.8}$$

The following expressions all *satisfy* the summation convention:

$$a_i b_i \tag{4.9a}$$

$$b_k A_{ik} + a_i + w_{jk} U_{pkqi} T_{jp} v_q \tag{4.9b}$$

$$a_i b_j \tag{4.9c}$$

$$a_i A_{ij} b_k \tag{4.9d}$$

The number of free indices indicates the number of expressions contained implicitly in a single indicial expression. The first expression above has no free indices, so it corresponds to a 0^{th}-order tensor (*i.e.,* a scalar). The second expression has one free index, so it corresponds to a 1^{st}-order tensor (*i.e.,* a vector). The next expression has two free indices, so it corresponds to a 2^{nd}-order tensor. The last expression also has two free indices, so it also corresponds to a 2^{nd}-order tensor. We will precisely define what is meant by the term "tensor" later; for now, we are merely clarifying notation and terminology. Incidentally, whenever the term “tensor” is used, it usually refers to a *second-order* tensor. Many authors alternatively use the word “rank” or “valence” instead of “order.” Hence, a tensor of *rank 2* is the same thing as what we will call a tensor of *order 2*. We prefer the term *order* to avoid confusion with the meaning of *rank* from matrix analysis.

Incidentally, note that Eq. (4.9b) uses k as a dummy sum index in both the first and last terms. This is perfectly legitimate. The rule demanding that a dummy index must appear exactly twice applies to each individual *term*. Written out in conventional form using summation symbols, Eq. (4.9b) is

$$\sum_{k=1}^{3}(b_k A_{ik}) + a_i + \sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}(w_{jk}U_{pkqi}T_{jp}v_q) \tag{4.10}$$

where the free index i ranges from 1 to 3.

Of course, summation is commutative (you can sum over *i* and then *j*, or vice-versa, without changing the final result). Said differently, the summation signs can be applied in any order. For example, the above equation could be written instead as

$$\sum_{k=1}^{3}(b_k A_{ik}) + a_i + \sum_{q=1}^{3}\sum_{k=1}^{3}\sum_{p=1}^{3}\sum_{j=1}^{3}(w_{jk}U_{pkqi}T_{jp}v_q) \tag{4.11}$$

Moving the summation symbols from the *jkpq* order to this *qkpj* order has no impact on the result.

Don’t forget what repeated indices really mean. Newcomers to tensor analysis sometimes forget that the summation rules are really just a notational convenience. Sometimes it’s wise to go back to conventional notation to simplify an indicial expression. Recall, for example, the definition of the Kronecker delta:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases} \tag{4.12}$$

Now consider the indicial expression

$$s = \delta_{kk} \tag{4.13}$$

There are no free indices, so the result must be a scalar. New students almost always get burned by using Eq. (4.12) to say that δ_{kk} must be equal to 1. However, this conclusion is wrong. The index k is *repeated*, so it must be summed. In other words, Eq. (4.13) really says

$$s = \sum_{k=1}^{3}\delta_{kk} = \delta_{11} + \delta_{22} + \delta_{33} = 1 + 1 + 1 = 3 \tag{4.14}$$

Recalling the true meaning of the indicial notation is also essential for simplifying other expressions where δ_{ij} appears. Consider, for example,

$$A_{km}\delta_{mj} \tag{4.15}$$

It is wise to mentally realize that this is really

$$\sum_{m=1}^{3} A_{km}\delta_{mj} \tag{4.16}$$

Using the definition of the Kronecker delta, note that

$$A_{km}\delta_{mj} = \begin{cases} A_{kj} & \text{if } m=j \\ 0 & \text{if } m \neq j \end{cases}$$

*The index "m" is not intended to be summed in this particular equation** (4.17)

Consequently, only one of the three terms ($A_{k1}\delta_{1j}$, $A_{k2}\delta_{2j}$, or $A_{k3}\delta_{3j}$) in Eq. (4.16) will be nonzero (the one where $m=j$). Therefore,

$$A_{km}\delta_{mj} = A_{kj} \tag{4.18}$$

This result is just one example of how a Kronecker delta may be removed from an expression whenever one of its indices is a dummy sum index. See page 64 for further examples.

Further special-situation summation rules. Here are two additional rules to go with the two main rules given on page 57.:

3. When a repeated subscript appears in the *divisor* of a fraction, it is understood to be implicitly summed within that divisor (i.e., the summation sign applies to that divisor, not to the whole expression). Free indices do *not* normally appear in divisors. It is permissible for the same repeated index symbol to appear separately in the numerator and divisor of an expression. For example,

$$\frac{A_{im}w_m}{w_m w_m} \quad \text{means} \quad \frac{\displaystyle\sum_{m=1}^{3} A_{im}w_m}{\displaystyle\sum_{m=1}^{3} w_m w_m} \tag{4.19}$$

4. Exponent and inverse operations take higher precedence than summation rules. For example, A_{ij}^{-1} means the ij component of $\underset{\approx}{A}^{-1}$, not $1/A_{ij}$. Similarly, A_{kk}^{-1} means $\text{tr}(\underset{\approx}{A}^{-1})$, not $1/\text{tr}\underset{\approx}{A}$. Whenever there is potential for confusion, it's advisable to show parentheses or to use words to clarify the intended meaning.

* It's important to inform your readers when you wish to temporarily suspend the summation conventions as we have done here. Some writers indicate that they do *not* wish for an index to be summed by marking it with an overbar, as in $A_{k\bar{m}}\delta_{\bar{m}i}$, or by putting the index in parentheses, as in $A_{k(m)}\delta_{(m)i}$ or by typesetting non-summed indices with capital letters, as in $A_{kM}\delta_{Mj}$.

Indicial notation in derivatives. The summation convention also applies to derivatives. For example

$$\frac{\partial y_{ik}}{\partial z_i} \tag{4.20}$$

means the same thing as

$$\sum_{i=1}^{3} \frac{\partial y_{ik}}{\partial z_i} \tag{4.21}$$

Even though derivatives are denoted using a structure that *looks like* division, note that the indicial notation conventions for derivatives are different from those for ratios. In particular, repeated subscripts in derivatives require the summation sign to go outside the entire expression, not just around the "divisor-like" part.

BEWARE: avoid implicit sums as independent variables. When you employ the summation convention in derivatives, you should be extra careful to be clear about what the independent variable is. Consider for example

$$w = b_{11} + b_{22} + b_{33} \tag{4.22}$$

What would the expression $\partial w / \partial b_{kk}$ mean? There are two possible answers:

$$\frac{\partial w}{\partial b_{kk}} = \frac{\partial w}{\partial (b_{11} + b_{22} + b_{33})} = 1 \tag{4.23}$$

or

$$\frac{\partial w}{\partial b_{kk}} = \frac{\partial w}{\partial b_{11}} + \frac{\partial w}{\partial b_{22}} + \frac{\partial w}{\partial b_{33}} = 1 + 1 + 1 = 3 \tag{4.24}$$

The two answers are not the same, so we need a *precedence* rule. Our experience in reading the literature is that most authors intend for the expression $\partial w / \partial b_{kk}$ to be interpreted as Eq. (4.24). Thus, the precedence rule is to always apply summations *after* taking derivatives. In other words, imagine that w is a function of nine b_{ij} components. After finding all nine $\partial w / \partial b_{ij}$ derivatives, then $\partial w / \partial b_{kk}$ is obtained by summing the three derivatives corresponding to $i = j$. To minimize confusion, we recommend that you write

$$\mu_{kk}, \text{ where } \mu_{ij} = \frac{\partial w}{\partial b_{ij}} \tag{4.25}$$

If you really want a derivative to be interpreted according to Eq. (4.23), then you should write the expression more carefully as

$$\frac{\partial w}{\partial s}, \text{ where } s = b_{kk} \tag{4.26}$$

Reading index STRUCTURE, not index SYMBOLS

This section tries to give a little insight to new students about how to read indicial formulas to maximize *understanding* and to avoid a queasy "alphabet soup" feeling when viewing expressions with lots of indices.

The actual symbols selected to be free and dummy indices are entirely up to you, so long as your choices do not violate the summation rules. Beginning students often fail to read indicial formulas in the most instructive manner. Experienced professionals know to instead look at the *structure* of where the free and dummy indices appear. For example, to memorize some strange formula like

$$Y_{ijk} = \frac{\partial a_{jk}}{\partial x_i} + \frac{\partial a_{ki}}{\partial x_j} - \frac{\partial a_{ij}}{\partial x_k}, \qquad (4.27)$$

you would be wise to ignore the actual symbols used for the indices. Instead, you should work to deduce the overall structure of the index placement. For example, if you want to write down the formula for Y_{jim}, then you could always painstakingly convert Eq. (4.27) by replacing in all occurrences of i by j, j by i, and k by m. However, doing it that way is error prone (especially since you cannot do it step-by-step because you would violate the sum rules by having four j's in the expression after step 1 of replacing i by j). People who *really understand* indicial notation would construct the formula for Y_{jim} by spending a little time "smelling" the structure of the Eq. (4.27). If you look carefully at that defining equation, you will note that the "denominators" in the three terms on the right hand side have the same indices as those on the left hand side — and they also appear in the same order. Thus, your first step to constructing the expression for Y_{jim} would be to write a partial "skeleton" formula as

$$Y_{jim} = \frac{\partial a_{??}}{\partial x_j} + \frac{\partial a_{??}}{\partial x_i} - \frac{\partial a_{??}}{\partial x_m}, \qquad (4.28)$$

where the "$??$" stands for indices not yet inserted. Again looking at the structure of Eq. (4.27) you would note that the subscripts on each $a_{??}$ are simply the "other two" indices not already in the "denominator." Furthermore, those subscripts are placed in an order that is a positive permutation of the free indices moving clockwise from the index already placed in the denominator. Specifically, the positive permutations of jim are: jim, imj, and mji. Because the first term in Eq. (4.28) has x_j in the "denominator," you know that the "other two" indices for the first term must be "im". Similarly, the second term has x_i in the denominator, so the other two indices must be mj (not jm because you need to select the same ordering as the positive permutation), etc. Thus, the final expression is

$$Y_{jim} = \frac{\partial a_{im}}{\partial x_j} + \frac{\partial a_{mj}}{\partial x_i} - \frac{\partial a_{ji}}{\partial x_m} \qquad (4.29)$$

In continuum mechanics, another common index structure is of the following form

$$U_{ijkl} = T_{ikjl} + W_{iljk} \qquad (4.30)$$

To “smell” the structure of the index placement, you might find it useful to mentally replace the indices with numbers representing the order in which you should write them on the paper:

$$U_{1234} = T_{1324} + W_{1423} \tag{4.31}$$

Thus, for example, if you want to write the formula for U_{pqim}, then you should again start with a partially completed skeleton in which you place only the first two indices:

$$U_{pq??} = T_{p?q?} + W_{p?q?} \tag{4.32}$$

Then you fill out the remaining two indices im by placing them in that order on the first term and in the reverse order in the second term to obtain

$$U_{pqim} = T_{piqm} + W_{pmqi} \tag{4.33}$$

Aesthetic (courteous) indexing

Once you have derived a final result, it is often helpful to your readers if you change your choice of index symbols so that the free indices are ordered alphabetically on the left hand side of the equation. For example, if your final result is something weird like

$$u_{skm} = A_{si} B_{mik}, \tag{4.34}$$

then re-writing it in the equivalent form,

$$u_{ijk} = A_{in} B_{knj}, \tag{4.35}$$

is often appreciated by (certain) readers.

Suspending the summation convention

An equation that violates the summation convention is usually a mistake, often indicating a typographical error or perhaps an error in which a dummy index was not changed to a different symbol prior to a substitution. Occasionally, however, one produces a *legitimate* equation that violates the summation convention, in which case, a temporary suspension of the summation convention must be indicated. We have already encountered one instance of needing to suspend the convention in our Eq. (4.17).

The summation conventions do not allow an index to appear three or more times in a term. Thus, for example, the expression

$$\lambda_1 \underset{\sim}{p}_1(\underset{\sim}{p}_1 \bullet \underset{\sim}{a}) + \lambda_2 \underset{\sim}{p}_2(\underset{\sim}{p}_2 \bullet \underset{\sim}{a}) + \lambda_3 \underset{\sim}{p}_3(\underset{\sim}{p}_3 \bullet \underset{\sim}{a}) \tag{4.36}$$

would have to be written in traditional form by explicitly showing the summation sign:

$$\sum_{k=1}^{3} \lambda_k \underset{\sim}{p}_k(\underset{\sim}{p}_k \bullet \underset{\sim}{a}) \tag{4.37}$$

The summation symbol cannot be removed because the summation index k appears three times. Here we had a legitimate expression in which an index occurred three times and a summation *was* desired. Below we consider the case when an index appears exactly twice and a summation is *not* desired.

Sometimes a *legitimate* indicial expression cannot be written in a manner that satisfies the summation convention. In such a situation, the summation convention must be temporarily suspended. For example, the i^{th} eigenvector $\{p\}_i$ corresponding to the i^{th} eigenvalue λ_i of a matrix $[A]$ is defined such that

$$[A]\{p\}_i = \lambda_i\{p\}_i \qquad \textit{(no sum on i).} \tag{4.38}$$

The phrase *"no sum on i"* tells the reader that the author wishes the index i to be interpreted as a *free* index even though it appears exactly twice on the right-hand side.

In tensor analysis, exceptions to the summation convention are rare, so it is a very convenient notational tool, especially when an expression contains numerous implied summations, as was the case in Eq. (3.105).

Combining indicial equations

The following equations all satisfy the summation rules:

$$\begin{aligned} a_i &= A_{ij}v_j & &\Leftrightarrow & \{a\} &= [A]\{v\} \\ b_i &= B_{ij}w_j & &\Leftrightarrow & \{b\} &= [B]\{w\} \\ s &= a_ib_i & &\Leftrightarrow & s &= \{a\}^T\{b\} \end{aligned} \tag{4.39}$$

The expressions on the right show the equivalent matrix expression for the operations. Note, in the last equation, that a transpose of $\{a\}$ is required in the matrix equation. There is no need for a transpose on a_i in the indicial expression — a_i^T is meaningless.

In the first two equations, the dummy summation index is j and the free index is i; hence, those equations actually represent *three separate equations* for each value of the free index "i". In the last expression, the dummy summation index is i, and there are no free indices (indicating that the equation is just a *single equation for a single scalar*).

It might be tempting to directly substitute the first two expressions in Eq. (4.39) into the third equation to obtain $s = A_{ij}v_jB_{ij}w_j$. However, this combined equation violates the summation convention because there are now *four* occurrences of the index j on the right-hand-side. To properly combine the equations in (4.39), the dummy index j must be changed to something else in one of the equations *before* combining the equations. The symbol used in place of j is arbitrary, except that it must not be i because i is already being used as a free index. Replacing j by k in the first of Eq. (4.39) gives $a_i = A_{ik}v_k$, allowing the equations to be combined to give

$$s = A_{ik}v_kB_{ij}w_j \qquad \Leftrightarrow \qquad s = \{v\}^T[A]^T[B]\{w\} \tag{4.40}$$

which does not violate the summation convention. Written out in traditional form, this equation may be written

$$s = \sum_{i=1}^{3}\sum_{k=1}^{3}\sum_{j=1}^{3} A_{ik}v_k B_{ij}w_j \tag{4.41}$$

In the above analysis, we showed the equivalent matrix expressions in the right-hand column. The above analysis could have been done entirely using matrix manipulations, but the skill of changing index symbols becomes indispensable when working with operations that have no straightforward matrix analogs.

Index-changing properties of the Kronecker delta

The indicial form of Eq. (3.22) may be written

$$\delta_{ij}v_j = v_i \tag{4.42}$$

Note that the Kronecker delta δ_{ij} acts in a way that appears to have "changed" the dummy summation index j on v_j to become an i on v_i with the δ_{ij} removed. This "Kronecker removal" property holds because, in the summation over j, only one term is nonzero (the one where $i=j$).

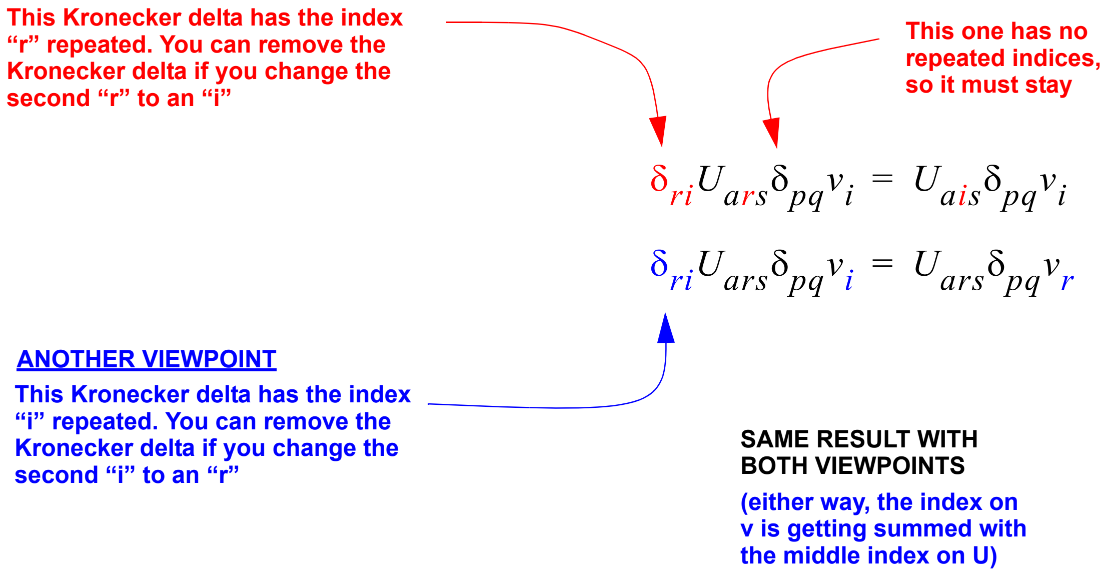
This index-changing property holds in general. For example, the expression $A_{ij}v_k\delta_{ip}$ can be simplified by noting that the subscript i on δ_{ip} is a dummy summation subscript. Therefore, δ_{ip} may be removed if the other occurrence of i is changed to a p. The simplified expression is therefore $A_{pj}v_k$. Conversely, given an expression $A_{pj}v_k$ for which you would prefer that the first index on A to be *"i"* instead of *"p"*, you can use a "Kronecker insertion rule" to replace the expression with $A_{ij}v_k\delta_{ip}$. These sorts of manipulations are routine, and this section covers in detail some "gotcha" precautions you must take to ensure that you don't violate the sum rules when changing indices.

We have already used the Kronecker-removal index-changing property when we simplified Eq. (3.86). The Kronecker-removal rule applies in more complicated expressions involving multiple Kronecker deltas. Consider, for example, the expression $T_{qms}\delta_{pk}v_q\delta_{km}\delta_{id}$. Here the subscript k appears exactly twice, so it is a dummy summation index, and the δ_{pk} may be removed from the expression if the other occurrence of k is changed to a p. This gives $T_{qms}v_q\delta_{pm}\delta_{id}$. Now note that the index m is a dummy sum index, so δ_{pm} may be removed if the other occurrence of m is changed to a p to give $T_{qps}v_q\delta_{id}$. The δ_{id} in this final expression may not be removed because both i and d are free indices, not summation indices.

Now consider the expression $\delta_{mk}A_{km}$. Noting that m is a summation index, this may be simplified by removing the Kronecker delta δ_{mk} if the other occurrence of m is replaced by k to give A_{kk}, which means $A_{11}+A_{22}+A_{33}$ and which is therefore equivalent to the trace operation of Eq. (3.51).

Kronecker Removal Rule

If one index on a Kronecker Delta is a dummy sum index (i.e., that index appears in two places), then the Kronecker delta may be removed if the *other* instance of the summed index is replaced by the second index on the Kronecker delta. This second index may be free or summed — it doesn't matter. (4.43)



This Kronecker delta has the index "r" repeated. You can remove the Kronecker delta if you change the second "r" to an "i"

This one has no repeated indices, so it must stay

$$\delta_{ri}U_{ars}\delta_{pq}v_i = U_{ais}\delta_{pq}v_i$$

$$\delta_{ri}U_{ars}\delta_{pq}v_i = U_{ars}\delta_{pq}v_r$$

ANOTHER VIEWPOINT
This Kronecker delta has the index "i" repeated. You can remove the Kronecker delta if you change the second "i" to an "r"

SAME RESULT WITH BOTH VIEWPOINTS
(either way, the index on v is getting summed with the middle index on U)

Above, we described how to simplify an expression by *removing* a Kronecker delta in which at least one index is summed. Specifically, Eq. (4.43) provided a way for you to *remove* the Kronecker delta from the expression. Frequently, however, you will want to use the index changing property in its converse by *reinserting* a Kronecker delta. This process is more tricky because you must first verify that the indices that you hope to use on the inserted Kronecker delta aren't already being used elsewhere in the equation or expression. If they are, you must be extra careful to ensure that your manipulations won't result in violation of the summation conventions. Consider, for example, an expression $s_n = B_{ins}A_{si} + A_{nm}v_m$. This expression is linear with respect to $[A]$, so we know[*] that there must exist coefficients C_{ijk} such that $s_i = C_{ijk}A_{jk}$. How do we go about finding these coefficients? By taking enough time to learn the index structure, you can probably slowly figure out that the final answer must be $C_{ijk} = B_{kij} + v_k\delta_{ij}$, but is there a step-by-step procedure that you can fall back on "slow-brain" days when a random walk through index changes just isn't producing what you seek? Such a methodical or algorithmic process would also be useful if you ever want to write a symbolic math program to do the index manipulations for you.

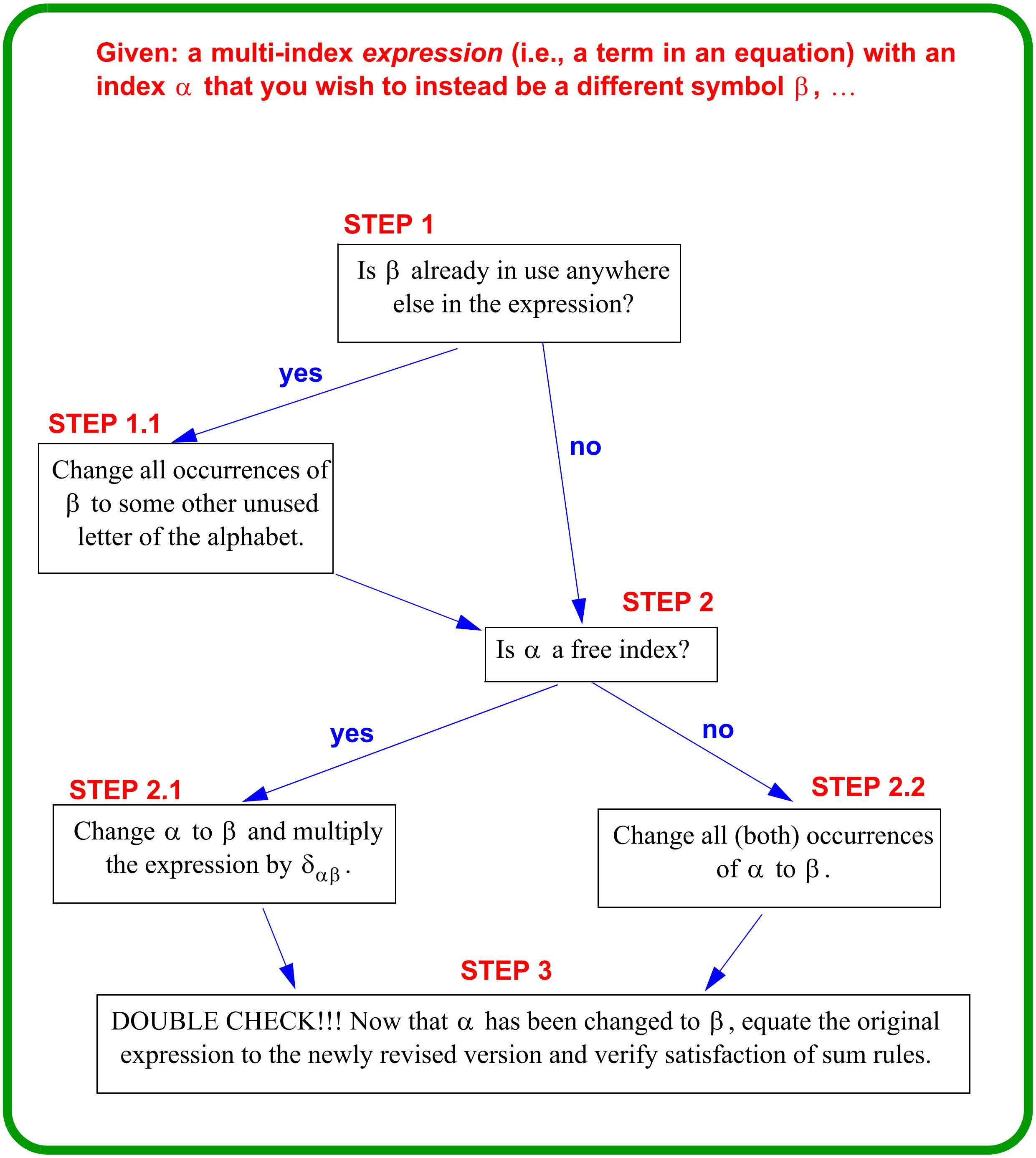
The first step in reformulating an *equation* is to look at the structure of the target equation, and compare that with the starting equation. If the target equation uses a different set or free indices, then your first order of business will be to re-cast the starting equation to use those free indices. In our example, we want to re-cast the equation $s_n = B_{ins}A_{si} + A_{nm}v_n$ into the new "target" form $s_i = C_{ijk}A_{jk}$. The free index in our target equation is *"i"*, but the free index in the starting equation is *"n"*. *WARNING: before simply changing "n" to "i", you must first change all other occurrences of "i" in the starting equation to any other unused letter of the alphabet.* Changing the *"i"* to, say, *"p"*, we get $s_n = B_{pns}A_{sp} + A_{nm}v_m$. Now that *"i"* is nowhere to be seen, we may change the free index *"n"* to the desired symbol *"i"*, giving $s_i = B_{pis}A_{sp} + A_{im}v_m$. When making these sorts of changes, remind yourself that a free index symbol will occur exactly once in *every* term, so don't forget to make the change *in every term*. Now that the free indices in our starting equation are the same as the free indices in the target equation, the new goal is to make adjustments in each individual *term*:

Given (as an illustrative example) $s_i = B_{pis}A_{sp} + A_{im}v_m$, (4.44a)

determine coefficients C_{ijk} such that $s_i = C_{ijk}A_{jk}$ (4.44b)

[*] from the Representation Theorem, covered later in Eq. 9.7.

Note from Eq. (4.44b) that the desired indices on $[A]$ are *"jk"*. However, the first term in Eq. (4.44a) has indices *"sp"* and the last term has indices *"im"*. By using the following flow chart individually on each term in Eq. (4.44a), you will be able to make the necessary index change adjustments.



Given: a multi-index *expression* (i.e., a term in an equation) with an index α that you wish to instead be a different symbol β, …

STEP 1: Is β already in use anywhere else in the expression?

STEP 1.1: Change all occurrences of β to some other unused letter of the alphabet.

STEP 2: Is α a free index?

STEP 2.1: Change α to β and multiply the expression by $\delta_{\alpha\beta}$.

STEP 2.2: Change all (both) occurrences of α to β.

STEP 3: DOUBLE CHECK!!! Now that α has been changed to β, equate the original expression to the newly revised version and verify satisfaction of sum rules.

You should apply this flow chart sequentially for each index that needs changing, making sure to always follow the steps using the latest revisions of the expressions.

Let's apply this flow chart to the first term on the right hand side of Eq. (4.44a), which is $B_{pis}A_{sp}$. Comparing with Eq. (4.44b), we desire the index *"s"* to instead be *"j"* (we also want *"p"* to instead be *"k"*, but the footnote of the flow chart instructs us to deal with only one index at a time, so let's first worry getting that *"s"* changed to a *"j"*). Using the flow chart,

STEP 1: Is *"j"* already in use anywhere else in the expression $B_{pis}A_{sp}$? ***No.***

STEP 2: Is *"s"* a free index? ***No***, it appears twice, so it is a dummy sum index.

STEP 2.2: Change both occurrences of *"s"* to *"j"* to obtain $B_{pij}A_{jp}$

STEP 3: Equate the starting expression (in step 1) to the final expression in step 2.2: $B_{pis}A_{sp} = B_{pij}A_{jp}$. Are there any sum rule violations? No. Both sides have the same free index (*"i"*) and all repeated indices appear *exactly* twice in their terms.

Now that we have completed one revision, let's work on the second subscript on $[A]$. Again comparing with Eq. (4.44b) with *our latest revision* $B_{pij}A_{jp}$, we see that we want that index *"p"* to instead be *"k"*. The path on the flow chart is similar to what was done above and the result is $B_{kij}A_{jk}$. This takes care of the first term in Eq. (4.44a). Now let's work on the second term, $A_{im}v_m$. Comparing with Eq. (4.44b), we want the index *"i"* to instead be *"j"*. Following the flow chart, here are the steps:

STEP 1: Is *"j"* already in use anywhere else in the expression $A_{im}v_m$? ***No.***

STEP 2: Is *"i"* a free index? ***YES***, because it appears exactly once.

STEP 2.1: Change *"i"* to *"j"* and multiply by δ_{ij} to obtain $A_{jm}v_m\delta_{ij}$

STEP 3: Equate the starting expression (in step 1) to the final expression in step 2.1: $A_{im}v_m = A_{jm}v_m\delta_{ij}$. Are there any sum rule violations? No. Both sides have the same free index (*"i"*) and all repeated indices appear *exactly* twice in their terms.

Working with the latest revision, $A_{jm}v_m\delta_{ij}$, again comparing with Eq. (4.44b) shows that we want the index *"m"* to instead be *"k"*. Following the flowchart again lets us revise the expression to read $A_{jk}v_k\delta_{ij}$. Recalling our revised version of the first term, we now have our final revision of Eq. (4.44):

$$s_i = B_{kij}A_{jk} + A_{jk}v_k\delta_{ij} \tag{4.45}$$

In this form, we can now compare with Eq. (4.44b) to assert that the coefficients we seek must be given by

$$C_{ijk} = B_{kij} + v_k\delta_{ij} \tag{4.46}$$

Note that step 2.1 of our flow chart used the following rule:

KRONECKER INSERTION RULE (FOR FREE INDICES ONLY)

If an expression has a FREE index α that you would like to instead be a different symbol β, then first change any existing dummy sum occurrences of β to any other unused letter. You may then change α to β if you multiply the expression by $\delta_{\alpha\beta}$.

(4.47)

Note that this rule (which is the converse of Eq. 4.43) is typically applied only to *expressions* (i.e., individual terms in an equation), and it can be applied only if occurrences of the desired index symbol are already present only as dummy sum indices, not as free indices. Note that application of this rule changes the index on a variable, without changing the free index. Specifically, after changing α to β and then multiplying by $\delta_{\alpha\beta}$, the index β will exist *twice*, so it is a dummy sum index and α is *still* the free index. After application of this rule, α will still be a free index, but it will now be a subscript on δ instead of on the original variable. Because the symbol α will still be the free index symbol after application of this rule, you will *not* generally apply this rule to *equations*. In equations, if you really want to change the letter of the alphabet used for a free index, you will need to make the change *in every single term of the equation*, making sure that the new symbol was not already in use prior to the change (if it was, remember to first change the other occurrences of the desired new symbol to any other letter of the alphabet).

Summing the Kronecker delta itself

Consider the expression $\delta_{ik}\delta_{ik}$. The index i is a dummy sum index on the first δ_{ik}, so *that* δ_{ik} (not the other one) may be removed if the second occurrence of i is changed to a k. The second occurrence of the k happens be on the second δ_{ik}. Thus $\delta_{ik}\delta_{ik}$ may be simplified to become δ_{kk}. Recalling that δ_{ij} equals 1 if $i=j$, one might be tempted to write $\delta_{kk}=1$, which would be wrong. The index k is still a summation index, and therefore

$$\delta_{kk} = \delta_{11} + \delta_{22} + \delta_{33} = 1 + 1 + 1 = 3 \tag{4.48}$$

Viewed differently, this says that the trace of the 3×3 identity matrix is equal to 3.

Our (unconventional) “under-tilde” notation

Conventional publishing practice typesets scalars in *italic* while vectors and tensors are set in ***bold***. As a teaching aid to add clarity to our discussion, we extend this practice by explicitly indicating the order of vectors and tensors by the number of under-tildes. Thus, for example, s denotes a scalar, $\underset{\sim}{\boldsymbol{v}}$ denotes a vector, $\underset{\approx}{\boldsymbol{T}}$ denotes a second-order tensor (to be defined later), $\underset{\approx\!\!\sim}{\boldsymbol{g}}$ denotes a third-order tensor, and so on.

Tensor invariant operations

Unless otherwise indicated, all vector and tensor components cited in this book are referenced to an orthonormal basis, which *could* be — but is not limited to — the laboratory basis. Lab components can be transformed to any other orthonormal basis by using the techniques covered on page __, where it is shown, for example, that components v_i of a vector $\underset{\sim}{\boldsymbol{v}}$ with respect to the lab basis $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ are related to the components $\hat{\underset{\sim}{v}}_j$ of the same vector with respect to a different orthonormal basis $\{\hat{\underset{\sim}{\boldsymbol{e}}}_1, \hat{\underset{\sim}{\boldsymbol{e}}}_2, \hat{\underset{\sim}{\boldsymbol{e}}}_3\}$ according to $v_i = Q_{ij}\hat{v}_j$ where $Q_{ij} = \underset{\sim}{\boldsymbol{e}}_i \bullet \hat{\underset{\sim}{\boldsymbol{e}}}_j$. The base vectors themselves are related by $\underset{\sim}{\boldsymbol{e}}_i = Q_{ij}\hat{\underset{\sim}{\boldsymbol{e}}}_j$. Importantly, it is shown on page __ that the $[Q]$ matrix is orthogonal (i.e., $Q_{ki}Q_{kj} = \delta_{ij}$).

Though probably not immediately obvious, this orthogonality property ensures that the component formulas for the vast majority of operations described in this book take precisely the same structural form in *any* orthonormal basis. For example, the vector dot product is defined for the lab basis as $a_k b_k$, and if you apply transformation formulas to substitute $a_k = Q_{ki}\hat{a}_i$ and $b_k = Q_{kj}\hat{a}_j$, then orthogonality of the $[Q]$ matrix shows that the dot product can be computed by $\hat{a}_k\hat{b}_k$, which is precisely of the same form as the formula that uses lab components. This sort of result is typical of tensor and vector operations. The linear transformation formula for $\underset{\sim}{\boldsymbol{y}} = \underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}}$ is computed by $y_i = F_{ij}x_j$ with respect to the lab basis and $\hat{y}_i = \hat{F}_{ij}\hat{x}_j$ with respect to a different "hat" basis (it is shown on page __ that the lab and "hat" components of the tensor $\underset{\approx}{\boldsymbol{F}}$ are related by $F_{ij} = Q_{im}Q_{jn}\hat{F}_{mn}$). Note that the component formulas are identical in structure. The *components* of the output vector are different in the two systems (i.e., $y_i \neq \hat{y}_i$). However, the output *vector* itself is computed by multiplying the components by the base vectors: $\underset{\sim}{\boldsymbol{y}} = y_i\underset{\sim}{\boldsymbol{e}}_i$ or $\underset{\sim}{\boldsymbol{y}} = \hat{y}_i\underset{\sim}{\boldsymbol{e}}_i$. Even though the components are different, this *product* of components times base vectors will give the *same* final result for $\underset{\sim}{\boldsymbol{y}}$. In fact, any operation whose component formula has the same structure in *any* orthonormal basis is referred to a **tensor invariant operation**. After multiplying components times base vectors, you will obtain *the same tensor (or vector)* regardless of which basis you used in your component calculations. Knowing that an operation is invariant is extremely helpful because it gives you licence to perform the operation *in any convenient orthonormal basis*, and you can rest assured that the resulting tensor (or vector) will be the same as if you had instead performed your component calculations in a different basis.

> *“Blessed are those who expect nothing,*
> *for they shall not be disappointed.”*
> *— Carl Sandburg*

5. Simple vector operations and properties

Dot product between two vectors

GOAL: Define, cite properties, show application to find angle between two vectors, show application to decide if a vector is zero.

Using the summation convention, the dot product between two vectors $\underset{\sim}{\mathbf{v}}$ and $\underset{\sim}{\mathbf{w}}$ is*

$$\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{w}} \equiv v_k w_k \tag{5.1}$$

Written out in full,

$$\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{w}} = v_1 w_1 + v_2 w_2 + v_3 w_3 \tag{5.2}$$

In matrix notation, this can be written

$$\begin{aligned}(\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{w}}) &= \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \{v\}^T\{w\} \\ &= \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = <v>\{w\}\end{aligned} \tag{5.3}$$

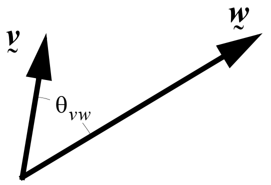
The magnitude of a vector $\underset{\sim}{\mathbf{v}}$ is given by

$$v = \|\underset{\sim}{\mathbf{v}}\| = +\sqrt{v_1^2 + v_2^2 + v_3^2} = +\sqrt{\underset{\sim}{\mathbf{v}} \bullet \underset{\sim}{\mathbf{v}}} \tag{5.4}$$

* This document presumes that all vectors are real. If however, you wish to generalize the inner product between two vectors $\underset{\sim}{\mathbf{v}}$ and $\underset{\sim}{\mathbf{w}}$ that might have complex components, then the dot product should be revised to $\underset{\sim}{\mathbf{v}} * \underset{\sim}{\mathbf{w}} = \bar{\underset{\sim}{\mathbf{v}}} \bullet \underset{\sim}{\mathbf{w}} = \bar{v}_k w_k$, where the overbar denotes the complex conjugate.

Geometrically, the dot product can be written

$$\underset{\sim}{v} \bullet \underset{\sim}{w} = vw \ \cos\theta_{vw} \ , \tag{5.5}$$



where v and w are the magnitudes of $\underset{\sim}{v}$ and $\underset{\sim}{w}$, respectively, and θ_{vw} is the angle between $\underset{\sim}{v}$ and $\underset{\sim}{w}$. The dot product is commutative:

$$\underset{\sim}{v} \bullet \underset{\sim}{w} = \underset{\sim}{w} \bullet \underset{\sim}{v} \tag{5.6}$$

It also satisfies the inner product positivity rule

$$\underset{\sim}{v} \bullet \underset{\sim}{v} > 0 \ \text{ if } \underset{\sim}{v} \neq \underset{\sim}{0}$$

$$\underset{\sim}{v} \bullet \underset{\sim}{v} = 0 \ \text{ if and only if } \underset{\sim}{v} = \underset{\sim}{0} \tag{5.7}$$

This positivity property implies that the magnitude of a vector, as defined in Eq. (5.4), will always be real.*

Dot product between orthonormal base vectors

If $\{\underset{\sim}{e}_1, \underset{\sim}{e}_2, \underset{\sim}{e}_3\}$ is an orthonormal basis, then

$$\underset{\sim}{e}_i \bullet \underset{\sim}{e}_j = \delta_{ij} , \tag{5.8}$$

where (recall) the symbol δ_{ij} is the Kronecker delta, defined

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{5.9}$$

A “quotient” rule (deciding if a vector is zero)

GOAL: Explain that you can’t define division by vectors, but there is an extended viewpoint that is similar.

Suppose the product of two *scalars* u and v is zero. Then we all know that you may conclude that $u=0$ and/or $v=0$. Furthermore, if $uv = 0$ *for all choices of* u, then you may conclude that v must be zero.

Now suppose that the dot product of two vectors $\underset{\sim}{u}$ and $\underset{\sim}{v}$ is zero. Then, based on the definition of the dot product, you may conclude that one of *three* possibilities exist: (1) $\underset{\sim}{u}$ is zero, (2) $\underset{\sim}{v}$ is zero, or (3) $\underset{\sim}{u}$ is perpendicular to $\underset{\sim}{v}$. More importantly...

$$\text{If } \underset{\sim}{u} \bullet \underset{\sim}{v} = \underset{\sim}{0} \ \textit{for all choices of } \underset{\sim}{u}, \text{ then } \underset{\sim}{v} = \underset{\sim}{0} . \tag{5.10}$$

A vector is zero if and only if all of its components are zero.

* Keep in mind that we are considering only ordinary engineering vectors having real components. If complex components were allowed, the inner product would be written $\bar{\underset{\sim}{v}} \bullet \underset{\sim}{w} \equiv \bar{v}_k w_k$, where the overbar denotes the complex conjugate.

Deciding if one vector equals another vector. Not surprisingly, two vectors $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{b}}$ are equal if and only if $\underset{\sim}{\boldsymbol{a}} - \underset{\sim}{\boldsymbol{b}} = \underset{\sim}{\boldsymbol{0}}$. Thus, the above tests for deciding if a vector is zero can be applied to tell if two vectors equal each other. Specifically, Eq. (5.10) can be written

$$\text{If } \underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{u}} = \underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{u}} \textit{ for all choices of } \underset{\sim}{\boldsymbol{u}}\text{, then } \underset{\sim}{\boldsymbol{a}} = \underset{\sim}{\boldsymbol{b}}. \tag{5.11}$$

Two vectors are equal if and only if corresponding components are equal. Without the proviso that Eq. (5.11) holds *for all choices of* $\underset{\sim}{\boldsymbol{u}}$, the best you would be able to conclude is that $\underset{\sim}{\boldsymbol{a}} - \underset{\sim}{\boldsymbol{b}}$ is perpendicular to the span of admissible choices for $\underset{\sim}{\boldsymbol{u}}$. If, for example, you were told that $\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{u}} = \underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{u}}$ for all vectors $\underset{\sim}{\boldsymbol{u}}$ in the 2-3 plane, then you could conclude that $\underset{\sim}{\boldsymbol{a}} - \underset{\sim}{\boldsymbol{b}}$ must point in the 1-direction. This would *not* imply that either $\underset{\sim}{\boldsymbol{a}}$ or $\underset{\sim}{\boldsymbol{b}}$ each individually point in the 1-direction; instead, it would imply that $a_2 = b_2$ and $a_3 = b_3$.

These statements seem self-evident in this simple 3D context, but become quite important in higher dimensions.

Finding the i-th component of a vector

GOAL: Show that the i^{th} component of a vector can be found by dotting that vector by the i^{th} base vector.

Recall that any vector $\underset{\sim}{\boldsymbol{v}}$ can be expressed as

$$\underset{\sim}{\boldsymbol{v}} = v_1 \underset{\sim}{\boldsymbol{e}}_1 + v_2 \underset{\sim}{\boldsymbol{e}}_2 + v_3 \underset{\sim}{\boldsymbol{e}}_3 \tag{5.12}$$

Dotting both sides with $\underset{\sim}{\boldsymbol{e}}_1$ gives $\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{e}}_1 = v_1$. Similarly, $\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{e}}_2 = v_2$ and $\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{e}}_3 = v_3$. Hence, the i^{th} component of a vector with respect to an *orthonormal* basis $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ can always be found by

$$v_i = \underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{e}}_i \tag{5.13}$$

This relationship can be derived using strictly indicial notation by noting that Eq. (5.12) is equivalent to

$$\underset{\sim}{\boldsymbol{v}} = v_k \underset{\sim}{\boldsymbol{e}}_k \tag{5.14}$$

Dotting both sides by $\underset{\sim}{\boldsymbol{e}}_i$, invoking Eq. (5.8), and finally using the index-changing property of the Kronecker delta (Eq. 4.43) gives

$$\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{e}}_i = v_k \underset{\sim}{\boldsymbol{e}}_k \bullet \underset{\sim}{\boldsymbol{e}}_i = v_k \delta_{ki} = v_i \tag{5.15}$$

which is equivalent to Eq. (5.13). This method for finding the component of a vector might seem at first glance to be trivial and of no obvious use. Suppose however, that $\{\underset{\sim}{\boldsymbol{e}}_1^*, \underset{\sim}{\boldsymbol{e}}_2^*, \underset{\sim}{\boldsymbol{e}}_3^*\}$ is a *different* basis. Further suppose that we *do* know the components of $\underset{\sim}{\boldsymbol{v}}$ with respect to the original (unstarred) basis, but we wish to determine the components of this vector with respect to this new (starred) basis. Then $v_i^* = \underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{e}}_i^*$. As a specific example, suppose that

$$\underset{\sim}{\boldsymbol{e}}_1^* = \tfrac{1}{5}(3\underset{\sim}{\boldsymbol{e}}_1 + 4\underset{\sim}{\boldsymbol{e}}_2), \quad \underset{\sim}{\boldsymbol{e}}_2^* = -\underset{\sim}{\boldsymbol{e}}_3^*, \quad \text{and} \quad \underset{\sim}{\boldsymbol{e}}_3^* = \tfrac{1}{5}(-4\underset{\sim}{\boldsymbol{e}}_1 + 3\underset{\sim}{\boldsymbol{e}}_2) \tag{5.16}$$

Suppose that we know that, with respect to the unstarred basis the vector $\underset{\sim}{\boldsymbol{v}}$ is given by

$$\underset{\sim}{v} = 2\underset{\sim}{e}_1 + 5\underset{\sim}{e}_2 \tag{5.17}$$

Then the components with respect to the unstarred basis are just $v_1 = 2$, $v_2 = 5$ and $v_3 = 0$, whereas the components of this vector with respect to the *starred* basis are

$$v_1^* = \underset{\sim}{v} \bullet \underset{\sim}{e}_1^* = (2\underset{\sim}{e}_1 + 5\underset{\sim}{e}_2) \bullet \tfrac{1}{5}(3\underset{\sim}{e}_1 + 4\underset{\sim}{e}_2) = \frac{6}{5} + \frac{20}{5} = \frac{26}{5} \tag{5.18}$$

$$v_2^* = \underset{\sim}{v} \bullet \underset{\sim}{e}_2^* = (2\underset{\sim}{e}_1 + 5\underset{\sim}{e}_2) \bullet (-\underset{\sim}{e}_3^*) = 0 \tag{5.19}$$

$$v_3^* = \underset{\sim}{v} \bullet \underset{\sim}{e}_3^* = (2\underset{\sim}{e}_1 + 5\underset{\sim}{e}_2) \bullet \tfrac{1}{5}(-4\underset{\sim}{e}_1 + 3\underset{\sim}{e}_2) = \frac{-8}{5} + \frac{15}{5} = \frac{7}{5} \tag{5.20}$$

The method presented in this section works only when the new (starred) basis is orthonormal; see page 88 to learn how to find the components of a vector with respect to an irregular (non-normalized and/or non-orthogonal) basis.

Even and odd vector functions

GOAL: Define, show function decomposition into even plus odd parts

If a function $f(\underset{\sim}{v})$ has the property that $f(-\underset{\sim}{v}) = f(\underset{\sim}{v})$, then the function is said to be **even**, and it is independent of the vector's directional sense. For example, the operation $\underset{\sim}{v} \bullet \underset{\sim}{v}$ is even. On the other hand, if $f(-\underset{\sim}{v}) = -f(\underset{\sim}{v})$, then the function is said to be **odd**.

Any function $f(\underset{\sim}{v})$ may be decomposed into its even part $f^e(\underset{\sim}{v})$ plus its odd part $f^o(\underset{\sim}{v})$ as

$$f(\underset{\sim}{v}) = f^e(\underset{\sim}{v}) + f^o(\underset{\sim}{v}) \tag{5.21}$$

where

$$f^e(\underset{\sim}{v}) \equiv \frac{1}{2}[f(\underset{\sim}{v}) + f(-\underset{\sim}{v})]$$

$$f^o(\underset{\sim}{v}) \equiv \frac{1}{2}[f(\underset{\sim}{v}) - f(-\underset{\sim}{v})] \tag{5.22}$$

Homogeneous functions

GOAL: Define, show identities

A function f is said to be **homogenous of degree k** if it satisfies

$$f(\alpha\underset{\sim}{v}) = \alpha^k f(\underset{\sim}{v}) \quad \text{for all positive scalars } \alpha. \tag{5.23}$$

For example, $f(x) = x^3$ is homogeneous of degree 3. The function $f(x) = x/|x|$ is homogeneous of degree zero. The function $f(x) = 1 + x$ is not homogeneous of any degree.

When a function is homogenous of degree k, then can be shown [__] that

$$f'(\underset{\sim}{v}) \bullet \underset{\sim}{v} = kf(\underset{\sim}{v}) \tag{5.24}$$

where $f'(\underset{\sim}{v}) = (\partial f / \partial v_i)\underset{\sim}{e}_i$. Furthermore, the function $f'(\underset{\sim}{v})$ is homogeneous of degree k–1. That is,

$$f'(\alpha\underset{\sim}{v}) = \alpha^{k-1} f'(\underset{\sim}{v}) \tag{5.25}$$

Vector orientation and sense

GOAL: Clarify terminology

The term "**orientation**" will be used to describe a line that is parallel to a given vector. The terms "**direction**" or "**sense**" will be used to describe the direction that a vector points along its orientational line. Thus, for example, the direction of $-\underset{\sim}{v}$ is opposite that of $\underset{\sim}{v}$ whereas the orientations of the two vectors are identical because they lie in parallel directions.

Suppose that a function $f(\underset{\sim}{v})$ is homogeneous of degree zero. In other words, putting k=0 in Eq. (5.23), we are supposing that a function satisfies the property that, for all scalars α, $f(\alpha\underset{\sim}{v}) = f(\underset{\sim}{v})$. Then the function $f(\underset{\sim}{v})$ depends only on the vector's orientation, not on its magnitude or sense.

Simple scalar components

GOAL: Find the scalar component of one vector in the direction of another.

The vector inner (dot) product is extremely useful for extracting parts of vectors (or tensors) in particular directions.

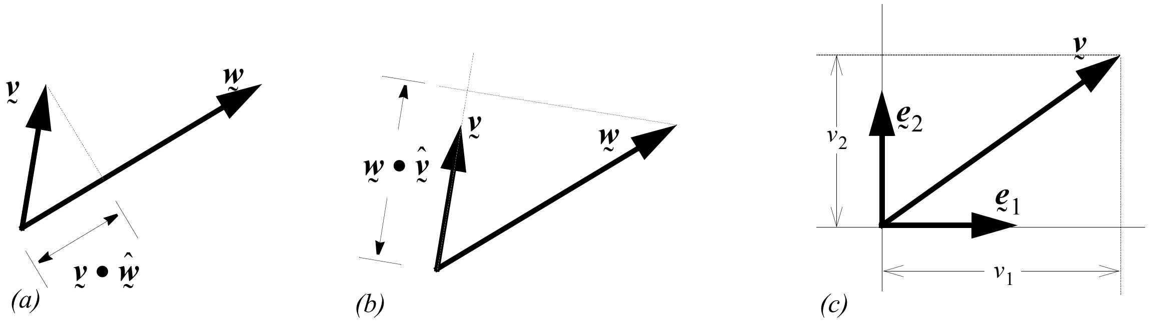


Figure 5.1. Finding components via projections. *(a)* scalar component of $\underset{\sim}{v}$ in the direction of $\underset{\sim}{w}$, *(b)* scalar component of $\underset{\sim}{w}$ in the direction of $\underset{\sim}{v}$, and *(c)* the scalar components of $\underset{\sim}{v}$ in the directions of the base vectors.

The operation

$$\underset{\sim}{v} \bullet \hat{\underset{\sim}{w}}, \qquad \text{where } \hat{\underset{\sim}{w}} \equiv \frac{\underset{\sim}{w}}{\|\underset{\sim}{w}\|} \tag{5.26}$$

gives the (rectangular) component of $\underset{\sim}{v}$ in the direction of a second vector $\underset{\sim}{w}$. Since we are only talking about the *direction* of $\underset{\sim}{w}$, the actual magnitude of $\underset{\sim}{w}$ is irrelevant — that's why the formula depends only on the unit vector $\hat{\underset{\sim}{w}}$ that points in the direction of $\underset{\sim}{w}$. The formula of Eq. (5.26) would not change if we were to multiply $\underset{\sim}{w}$ by any positive scalar α. The result changes sign if $\underset{\sim}{w}$ is replaced by $-\underset{\sim}{w}$.

Just as $\underset{\sim}{v}$ can be projected onto $\underset{\sim}{w}$, the reverse operation of projecting $\underset{\sim}{w}$ in the direction of $\underset{\sim}{v}$ gives the component of $\underset{\sim}{w}$ in the direction of $\underset{\sim}{v}$.

$$\underset{\sim}{w} \bullet \hat{\underset{\sim}{v}}, \qquad \text{where } \hat{\underset{\sim}{v}} \equiv \frac{\underset{\sim}{v}}{\|\underset{\sim}{v}\|} \tag{5.27}$$

This operation depends on the direction of $\underset{\sim}{v}$, but not its magnitude.

Eq. (5.13) is a special case of Eq. (5.26) in which $\underset{\sim}{w}$ is $\underset{\sim}{e}_i$. Thus, v_i is the projection of $\underset{\sim}{v}$ in the i^{th} coordinate direction.

Cross product

GOAL: Define, show identities, show how to find the area of the parallelogram formed by two vectors

The cross product between two vectors, $\underset{\sim}{a}$ and $\underset{\sim}{b}$ is a vector defined as:

$$\underset{\sim}{a} \times \underset{\sim}{b} = ab(\sin\theta_{ab})\hat{\underset{\sim}{n}}, \tag{5.28}$$

The unit vector $\hat{\underset{\sim}{n}}$ is perpendicular to both $\underset{\sim}{a}$ and $\underset{\sim}{b}$, with a direction defined by the right hand rule when sweeping from $\underset{\sim}{a}$ to $\underset{\sim}{b}$ through the angle θ_{ab}. Some books will tell you that the angle θ_{ab} must be less than $180°$, but that's not necessary. While it's true that the smaller angle between $\underset{\sim}{a}$ and $\underset{\sim}{b}$ is conventionally used, you can alternatively sweep through the larger angle, again by the right-hand rule; doing that will change the sign of *both* $\sin\theta_{ab}$ and $\hat{\underset{\sim}{n}}$, making the result in Eq. (5.28) unchanged.

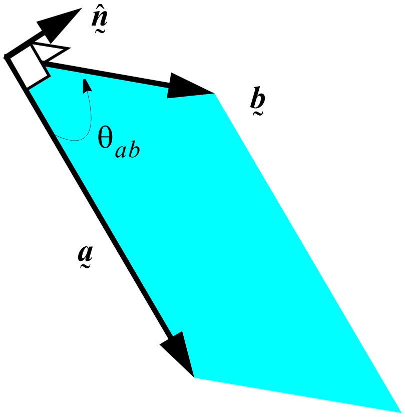


Figure 5.2. Cross product

The parallelogram in the illustration has an orientation perpendicular to $\underset{\sim}{a} \times \underset{\sim}{b}$. The area of the parallelogram equals the magnitude of $\underset{\sim}{a} \times \underset{\sim}{b}$.

If $\underset{\sim}{u} = \underset{\sim}{a} \times \underset{\sim}{b}$, then the components of $\underset{\sim}{u}$ are

$$\begin{aligned} u_1 &= a_2 b_3 - a_3 b_2 \\ u_2 &= a_3 b_1 - a_1 b_3 \\ u_3 &= a_1 b_2 - a_2 b_1 \end{aligned} \tag{5.29}$$

Hence,

$$\underset{\sim}{a} \times \underset{\sim}{b} = (a_2 b_3 - a_3 b_2)\underset{\sim}{e}_1 + (a_3 b_1 - a_1 b_3)\underset{\sim}{e}_2 + (a_1 b_2 - a_2 b_1)\underset{\sim}{e}_3 \tag{5.30}$$

Heuristically, this equation is often written as the determinant of a matrix:

$$\underset{\sim}{a} \times \underset{\sim}{b} \quad \text{“=”} \quad \det \begin{bmatrix} \underset{\sim}{e}_1 & \underset{\sim}{e}_2 & \underset{\sim}{e}_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \tag{5.31}$$

Cross product between orthonormal base vectors

GOAL: Cite important special-case of the cross product between base vectors.

Applying the definition of the cross product reveals that the nine possible ways to cross the orthonormal base vectors with themselves give

$$\begin{array}{lll} \underset{\sim}{e}_1 \times \underset{\sim}{e}_1 = \underset{\sim}{0} & \underset{\sim}{e}_1 \times \underset{\sim}{e}_2 = \underset{\sim}{e}_3 & \underset{\sim}{e}_1 \times \underset{\sim}{e}_3 = -\underset{\sim}{e}_2 \\ \underset{\sim}{e}_2 \times \underset{\sim}{e}_1 = -\underset{\sim}{e}_3 & \underset{\sim}{e}_2 \times \underset{\sim}{e}_2 = \underset{\sim}{0} & \underset{\sim}{e}_2 \times \underset{\sim}{e}_3 = \underset{\sim}{e}_1 \\ \underset{\sim}{e}_3 \times \underset{\sim}{e}_1 = \underset{\sim}{e}_2 & \underset{\sim}{e}_3 \times \underset{\sim}{e}_2 = -\underset{\sim}{e}_1 & \underset{\sim}{e}_3 \times \underset{\sim}{e}_3 = \underset{\sim}{0} \end{array} \tag{5.32}$$

These expressions may be written compactly using the summation notation as

$$\boxed{\underset{\sim}{e}_i \times \underset{\sim}{e}_j = \varepsilon_{ijk} \underset{\sim}{e}_k} \tag{5.33}$$

where ε_{ijk} is the permutation symbol defined in Eq. (3.76).

You can use Eq. (5.33) to deduce the component (indicial) expression of the cross product. Namely,

$$\underset{\sim}{a} \times \underset{\sim}{b} = (a_i \underset{\sim}{e}_i) \times (b_j \underset{\sim}{e}_j) = a_i b_j (\underset{\sim}{e}_i \times \underset{\sim}{e}_j) = a_i b_j \varepsilon_{ijk} \underset{\sim}{e}_k \tag{5.34}$$

The last step applied Eq. (5.33). Noting that the final result is the sum over k of an expression times $\underset{\sim}{e}_k$, the k^{th} component of the cross product operation must therefore be

$$\boxed{(\underset{\sim}{a} \times \underset{\sim}{b})_k = a_i b_j \varepsilon_{ijk}} \tag{5.35}$$

This formula relies on our previously stated assumption that all component formulas in this book are referenced to an orthonormal *right-handed* basis. If you use an orthonormal left-handed basis, then the above formula would be $(\underset{\sim}{a} \times \underset{\sim}{b})_k = -a_i b_j \varepsilon_{ijk}$. Some authors take this situation as an implication that the permutation symbol for left-handed base vectors should be defined as the negative of the right-handed definition. This is misleading and wrong-headed. What's really going on is that the cross product operation can be most generally written in structured notation as a special third-order *alternating* tensor $\underset{\approx}{\xi}$ operating on the vectors $\underset{\sim}{a}$ and $\underset{\sim}{b}$. [Namely, using notation to be defined later in this book, $\underset{\sim}{a} \times \underset{\sim}{b} = \underset{\approx}{\xi} \boldsymbol{:} (\underset{\sim}{a}\underset{\sim}{b})$.] The components of the alternating tensor can be proved to equal $\xi_{ijk} = \varepsilon_{ijk}$ if the basis is orthonormal and right-handed, while $\xi_{ijk} = -\varepsilon_{ijk}$ if the basis is orthonormal but left-handed. The ξ_{ijk} components have yet a different form if the basis is non-orthogonal or non-normalized.* Note that the permutation symbol does not change sign for a left-handed basis — only the components of the alternating *tensor* $\underset{\approx}{\xi}$ change upon a change of basis. We reiterate that, although the form of the component formulas for $\underset{\sim}{a} \times \underset{\sim}{b}$ depend on the nature of the basis, the final result for $\underset{\sim}{a} \times \underset{\sim}{b}$ is the same in all cases. In other words, $\underset{\sim}{a} \times \underset{\sim}{b}$ represents a particular and unique vector that does not depend on the underlying basis. Only the *method* that you must use to determine this unique vector varies depending on the choice of basis.

* Namely, if $\{\underset{\sim}{g}_1, \underset{\sim}{g}_2, \underset{\sim}{g}_3\}$ is a linearly independent set of vectors, then $\xi_{ijk} = \underset{\sim}{g}_i \bullet (\underset{\sim}{g}_j \times \underset{\sim}{g}_k)$.

Triple scalar product

GOAL: Define, cite properties, show application to deciding if three vectors are linearly independent.

Given three vectors, $\underset{\sim}{a}$, $\underset{\sim}{b}$, and $\underset{\sim}{c}$, the triple scalar product is defined

$$[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}] \equiv (\underset{\sim}{a} \times \underset{\sim}{b}) \bullet \underset{\sim}{c} \tag{5.36}$$

For rectangular Cartesian components, this operation can be computed by

$$[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}] = \det \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \tag{5.37}$$

The fully indicial form of the triple scalar product is

$$[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}] = \varepsilon_{ijk} a_i b_j c_k \tag{5.38}$$

The triple scalar product can be used to test whether or not three vectors are linearly independent:

Three vectors, $\underset{\sim}{a}$, $\underset{\sim}{b}$, and $\underset{\sim}{c}$, (in 3D space) are linearly independent if and only if $[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}] \neq 0$. (5.39)

It can be shown that the triple scalar product has the same value if the vectors are positively permuted and it changes sign if the vectors are negatively permuted. Specifically,

$$[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}] = [\underset{\sim}{b}, \underset{\sim}{c}, \underset{\sim}{a}] = [\underset{\sim}{c}, \underset{\sim}{a}, \underset{\sim}{b}] \tag{5.40}$$

and

$$[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}] = -[\underset{\sim}{c}, \underset{\sim}{b}, \underset{\sim}{a}] = -[\underset{\sim}{b}, \underset{\sim}{a}, \underset{\sim}{c}] = -[\underset{\sim}{a}, \underset{\sim}{c}, \underset{\sim}{b}] \tag{5.41}$$

Physically, the absolute value of $[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}]$ equals the volume of the parallelepiped formed by $\underset{\sim}{a}$, $\underset{\sim}{b}$, and $\underset{\sim}{c}$. The sign of $[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}]$ will be positive if the vectors form a "**right-handed**" triad, meaning that when the thumb of the right hand points in the direction of $\underset{\sim}{a}$ and the index finger points in the direction of $\underset{\sim}{b}$ then the middle finger will point *roughly* in the direction of $\underset{\sim}{c}$ (*i.e.*, if the middle finger were a vector, it would have a positive dot product with $\underset{\sim}{c}$). If the sign of $[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}]$ is negative, then the triad is said to be "**left-handed**."

Triple scalar product between orthonormal RIGHT-HANDED base vectors

GOAL: Cite the triple scalar product between right-handed base vectors and argue against redefining the permutation symbol for other types of basis triads (instead, advocate using different values for the permutation tensor components).

Dotting Eq. (5.33) by $\underset{\sim}{e}_m$ gives

$$(\underset{\sim}{e}_i \times \underset{\sim}{e}_j) \bullet \underset{\sim}{e}_m = \varepsilon_{ijk}\delta_{km} = \varepsilon_{ijm} \tag{5.42}$$

where ε_{ijk} is the permutation symbol defined in Eq. (3.76). Expressing the above result with the free index m replaced by k gives

$$\boxed{\varepsilon_{ijk} = [\underset{\sim}{e}_i, \underset{\sim}{e}_j, \underset{\sim}{e}_k]} \tag{5.43}$$

The triple scalar product $[\underset{\sim}{e}_i, \underset{\sim}{e}_j, \underset{\sim}{e}_k]$ is certainly well-defined if the basis is left-handed; in fact, the result is $[\underset{\sim}{e}_i, \underset{\sim}{e}_j, \underset{\sim}{e}_k] = -\varepsilon_{ijk}$. As mentioned earlier, this fact should not be used as justification for *redefining* the permutation symbol when using a left-handed basis. We recommend always defining the permutation symbol such that $\varepsilon_{123}{=}{+}1$ even if the basis is left-handed.

Even though the term "tensor" has not yet been defined, it's worth mentioning here (for future reference) that a straightforward formula exists for constructing a third-order permutation *tensor* in terms of any basis— including left-handed and even non-orthonormal. The permutation tensor components with respect to a non-orthonormal basis take yet a different form (covered in the more advanced Ref. [6]), and the process of computing the triple scalar product becomes more complicated. Nonetheless, the basic concept is the same: the triple scalar product of three vectors equals the *triple inner product* of the alternating tensor into the "dyadic" multiplication (defined later) of the three vectors. It just *happens* that the components of the alternating tensor equal the permutation symbol when the basis is orthonormal and right-handed. It is for this reason that we denote the alternating *tensor* by a symbol ξ_{ijk} that differs from its right-handed orthonormal components ε_{ijk}. A similar notational choice is also made when we denote the identity *tensor,* $\underset{\approx}{I}$, by a symbol that differs from its right-handed orthonormal components (the Kronecker delta δ_{ij}).

> *"I am always ready to learn, although*
> *I do not always like to be taught"*
> *— Winston Churchill*

6. Projections

Mathematically, a projection operation is any function $P(\underset{\sim}{x})$ for which

$$P(P(\underset{\sim}{x})) = P(\underset{\sim}{x}) \tag{6.1}$$

In other words, once the function P has been applied, then applying it a second time will not change the result. Below, we will discuss rank-1 projections that will return the part of a vector $\underset{\sim}{x}$ in the direction of some other vector. This is a projection because, if the first vector is already parallel to the second vector, the projection operation will simply return the first vector unchanged. We will also discuss rank-2 projections that return the part of a vector contained in a specified plane. This type of projection is like a formula for finding the shadow of a vector on the plane. If the vector is already in the plane, it's shadow is itself, making this operation a projection. As discussed below, projections can be linear or nonlinear.

Orthogonal (perpendicular) linear projections

GOAL: Show how to find the part of a vector in the direction of another vector (and clarify distinction between this and finding the scalar component in the direction of a vector).

Equation (5.26) shows how to find the *scalar* component of $\underset{\sim}{v}$ in the direction of $\underset{\sim}{w}$. If we were to multiply the result by $\hat{\underset{\sim}{w}}$, then we would obtain a *vector* that is in the direction of $\underset{\sim}{w}$ or, if $\underset{\sim}{v} \bullet \hat{\underset{\sim}{w}} < 0$, a vector that is in the opposite direction of $\underset{\sim}{w}$. This operation, in which we multiply the component of $\underset{\sim}{v}$ in the direction of $\hat{\underset{\sim}{w}}$ by $\hat{\underset{\sim}{w}}$ itself is called the *orthogonal vector projection of* $\underset{\sim}{v}$ *onto* $\underset{\sim}{w}$, and we will denote this vector by $\underset{\sim}{v}^{Dw}$ (which may be read as "the part of $\underset{\sim}{v}$ in the direction of $\underset{\sim}{w}$"). The mathematical definition is

$$\underset{\sim}{v}^{Dw} = (\underset{\sim}{v} \bullet \hat{\underset{\sim}{w}})\hat{\underset{\sim}{w}} \qquad \text{where } \hat{\underset{\sim}{w}} \equiv \frac{\underset{\sim}{w}}{\|\underset{\sim}{w}\|} \tag{6.2}$$

For some applications, it is prudent to recognize that the projection operation involves two vectors, $\underset{\sim}{v}$ and $\underset{\sim}{w}$, so it is sometimes advisable to instead define a binary* operator D such that $D(\underset{\sim}{v}, \underset{\sim}{w})$ denotes the part of $\underset{\sim}{v}$ in the direction of $\underset{\sim}{w}$. In terms of this alternative notation, Eq. (6.2) would read

* The term *binary* is merely a fancy way of saying that the operator has *two* arguments.

$$D(\underset{\sim}{v}, \underset{\sim}{w}) = \left(\frac{\underset{\sim}{v} \bullet \underset{\sim}{w}}{\underset{\sim}{w} \bullet \underset{\sim}{w}}\right)\underset{\sim}{w} \tag{6.3}$$

Clearly,

$$D(\underset{\sim}{v}, \underset{\sim}{w}) \neq D(\underset{\sim}{w}, \underset{\sim}{v}) \tag{6.4}$$

The part of the vector $\underset{\sim}{v}$ that is “left over” — i.e., $\underset{\sim}{v} - \underset{\sim}{v}^{Dw}$ — is the part of the vector $\underset{\sim}{v}$ that is perpendicular to $\underset{\sim}{w}$, which we will denote by $\underset{\sim}{v}^{Pw}$ or by $P(\underset{\sim}{v}, \underset{\sim}{w})$. It is defined by

$$\underset{\sim}{v}^{Pw} = \underset{\sim}{v} - (\underset{\sim}{v} \bullet \hat{\underset{\sim}{w}})\hat{\underset{\sim}{w}} \tag{6.5}$$

or

$$P(\underset{\sim}{v}, \underset{\sim}{w}) = \underset{\sim}{v} - \left(\frac{\underset{\sim}{v} \bullet \underset{\sim}{w}}{\underset{\sim}{w} \bullet \underset{\sim}{w}}\right)\underset{\sim}{w} \tag{6.6}$$

Note that $\underset{\sim}{v}^{Pw}$ lies in the plane perpendicular to $\underset{\sim}{w}$. Also note that

$$P(\underset{\sim}{v}, \underset{\sim}{w}) + D(\underset{\sim}{v}, \underset{\sim}{w}) = \underset{\sim}{v} \tag{6.7}$$

This equation shows more clearly that the operators P and D *decompose* the vector $\underset{\sim}{v}$ into two parts, one parallel to $\underset{\sim}{w}$ and the remainder perpendicular to $\underset{\sim}{w}$.

To make a sketch of $\underset{\sim}{v}^{Dw}$, simply draw a line parallel to $\underset{\sim}{w}$ that passes through the tail of $\underset{\sim}{v}$. Then draw a line perpendicular to $\underset{\sim}{w}$ (or, for 3D problems, draw a *plane* perpendicular to $\underset{\sim}{w}$) that passes through the tip of the vector $\underset{\sim}{v}$. These two lines (or, for 3D problems, the line and the plane) will intersect at a single point, which we will call A. Then $\underset{\sim}{v}^{Dw}$ is the vector extending from the tail of $\underset{\sim}{v}$ to A and $\underset{\sim}{v}^{Pw}$ is the vector extending from A to the *tip* of $\underset{\sim}{v}$. See Fig. 6.1.

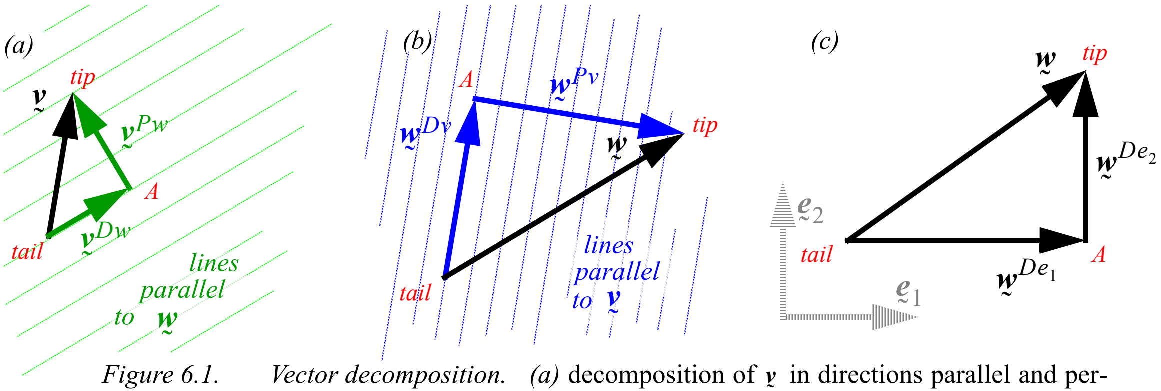

(a)
tip
$\underset{\sim}{v}$
$\underset{\sim}{v}^{Pw}$
A
$\underset{\sim}{v}^{Dw}$
tail
lines parallel to $\underset{\sim}{w}$
(b)
A
$\underset{\sim}{w}^{Pv}$
$\underset{\sim}{w}^{Dv}$
tip
$\underset{\sim}{w}$
lines parallel to $\underset{\sim}{v}$
tail
(c)
$\underset{\sim}{w}$
tip
$\underset{\sim}{w}^{De_2}$
$\underset{\sim}{e}_2$
tail
$\underset{\sim}{e}_1$
$\underset{\sim}{w}^{De_1}$
A

Figure 6.1. *Vector decomposition.* *(a)* decomposition of $\underset{\sim}{v}$ in directions parallel and perpendicular to $\underset{\sim}{w}$, *(b)* decomposition of $\underset{\sim}{w}$ into parts parallel and perpendicular to $\underset{\sim}{v}$, and *(c)* the projection of $\underset{\sim}{w}$ onto the unit base vectors.

As mentioned earlier, $D(\underset{\sim}{v}, \underset{\sim}{w}) \neq D(\underset{\sim}{w}, \underset{\sim}{v})$. Nonetheless. The vector $\underset{\sim}{w}$ may be similarly decomposed into parts that are in parallel and perpendicular to $\underset{\sim}{v}$ by the formulas

$$\underset{\sim}{w}^{Dv} = D(\underset{\sim}{w}, \underset{\sim}{v}) = (\underset{\sim}{w} \bullet \hat{\underset{\sim}{v}})\hat{\underset{\sim}{v}} \quad \text{and}$$
$$\underset{\sim}{w}^{Pv} = P(\underset{\sim}{w}, \underset{\sim}{v}) = \underset{\sim}{w} - (\underset{\sim}{w} \bullet \hat{\underset{\sim}{v}})\hat{\underset{\sim}{v}}$$
$$\text{where } \hat{\underset{\sim}{v}} \equiv \frac{\underset{\sim}{v}}{\|\underset{\sim}{v}\|} \tag{6.8}$$

Figure 6.1 clearly shows that this decomposition of $\underset{\sim}{w}$ results in completely different vectors as the decomposition of $\underset{\sim}{v}$.

The last sketch in Fig. 6.1 shows the decomposition of a vector $\underset{\sim}{w}$ into parts that are aligned with the base vectors. In particular, note that we labeled the vertical part of the vector as $\underset{\sim}{w}^{De_2}$ because, in three dimensions $\underset{\sim}{w}^{De_2} \neq \underset{\sim}{w}^{Pe_1}$. Suppose, for example, that the vector is $\underset{\sim}{w} = 6\underset{\sim}{e}_1 + 3\underset{\sim}{e}_2 + 7\underset{\sim}{e}_3$. Then

$$\underset{\sim}{w}^{De_2} = 3\underset{\sim}{e}_2 \tag{6.9}$$

whereas

$$\underset{\sim}{w}^{Pe_1} = 3\underset{\sim}{e}_2 + 7\underset{\sim}{e}_3 . \tag{6.10}$$

Rank-1 orthogonal projections

GOAL: Explain that finding the part of a vector in the direction of another vector is a projection operation, and explain why this projection has "rank 1".

The notation $\underset{\sim}{a}^{Db}$ symbolizes the orthogonal (perpendicular) projection of a vector $\underset{\sim}{a}$ in the direction of a second vector $\underset{\sim}{b}$. The symbol $\underset{\sim}{a}^{Db}$ may be referred to as "the part of $\underset{\sim}{a}$ in the direction of $\underset{\sim}{b}$." Since this operation involves two vectors, you might alternatively prefer to denote it in a more standard way by $D(\underset{\sim}{a}, \underset{\sim}{b})$. The result is always going to be a vector *parallel* to $\underset{\sim}{b}$. The adjective "orthogonal" or "perpendicular" is used to indicate that each point on the vector $\underset{\sim}{a}$ maps to the *closest* point on a line parallel to $\underset{\sim}{b}$; therefore, the *projection direction* is a family of lines that are perpendicular to $\underset{\sim}{b}$. The set of *all* vectors parallel to $\underset{\sim}{b}$ forms a *one-dimensional subspace* because any member of the set may be written in the form $\alpha\underset{\sim}{b}$, which involves only *one* arbitrary scalar α.

If $\underset{\sim}{n}$ is a unit vector, then any vector $\underset{\sim}{x}$ can be projected into a part that is parallel to $\underset{\sim}{n}$ and simultaneously perpendicular to the plane whose normal is $\underset{\sim}{n}$

$$\boxed{\underset{\sim}{p} = \underset{\sim}{n}(\underset{\sim}{n} \bullet \underset{\sim}{x})} \tag{6.11}$$

In other words, $\underset{\sim}{p}$ is just $\underset{\sim}{x}^{Dn}$. This projection has rank 1 because the target space is the one-dimensional line parallel to $\underset{\sim}{p}$. If one were to compute the average of $\underset{\sim}{x}$ over all rotations of $\underset{\sim}{x}$ about $\underset{\sim}{n}$, then the result would be the vector $\underset{\sim}{p}$.

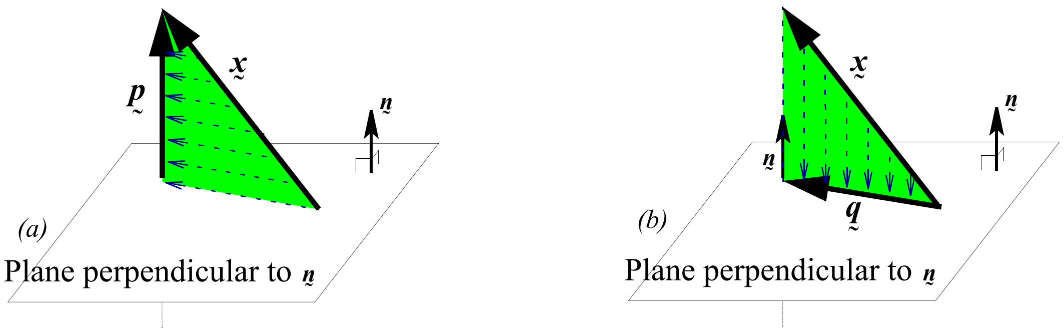


Figure 6.2. *(a) Rank-1 orthogonal projection, and (b) Rank-2 orthogonal projection.* The dashed lines show the projection direction. For the Rank-1 orthogonal projection, the dashed lines point from a location on $\underset{\sim}{x}$ to a target line that is parallel to the unit vector $\underset{\sim}{n}$ (the target line can be translated so that it passes through any point in the plane without changing the result for the projection vector $\underset{\sim}{p}$. For the Rank-2 projection, the dashed lines point from a location on $\underset{\sim}{x}$ toward the shortest distance to the plane *if the plane is positioned at the tail of the vector* $\underset{\sim}{x}$. The projections are orthogonal because the path from the vector $\underset{\sim}{x}$ to the target line or plane is the shortest distance.

Rank-2 orthogonal projections

GOAL: Show how to find the orthogonal (nearest) projection of a vector onto the plane perpendicular to another vector.

As sketched in Fig. 6.2, any vector $\underset{\sim}{x}$ can be projected onto a plane with unit normal $\underset{\sim}{n}$ by using the formula

$$\boxed{\underset{\sim}{q} = \underset{\sim}{x} - \underset{\sim}{n}(\underset{\sim}{n} \bullet \underset{\sim}{x})}. \tag{6.12}$$

In other words, $\underset{\sim}{q}$ is just $\underset{\sim}{x}^{Pn}$. This transformation is called an orthogonal projection because the projected vector $\underset{\sim}{q}$ represents the "shadow" cast by $\underset{\sim}{x}$ onto the plane perpendicular to $\underset{\sim}{n}$ when the light rays (formally called *level lines*) are coaxial with $\underset{\sim}{n}$. (Note: the term "coaxial" here means aligned, but not necessarily of the same sense.) The projection is second-rank because the target space to which the vector is being projected is two dimensional (a plane).

Basis interpretation of orthogonal projections

GOAL: Emphasize that the rank-1 and rank-2 projections effectively extract "pieces" of the starting vector that break it down into smaller parts relative to a locally aligned basis.

The vector $\underset{\sim}{n}$ can have any orientation. In order to better interpret equations like (6.11) and (6.12), it is often useful to set up a basis that has, say, its $\underset{\sim}{e}_3$ direction aligned with $\underset{\sim}{n}$. If we align $\underset{\sim}{e}_3$ with $\underset{\sim}{n}$, then the plane in Fig. 6.2, would be the 1-2 plane. Substituting $\underset{\sim}{n}=\underset{\sim}{e}_3$ into Eqs. (6.11) and (6.12) and applying Eq. (5.13) gives

$$\underset{\sim}{p} = \underset{\sim}{e}_3(\underset{\sim}{e}_3 \bullet \underset{\sim}{x}) = x_3\underset{\sim}{e}_3. \tag{6.13}$$

$$\underset{\sim}{q} = \underset{\sim}{x} - x_3\underset{\sim}{e}_3 = x_1\underset{\sim}{e}_1 + x_2\underset{\sim}{e}_2. \tag{6.14}$$

Thus, the projected vector $\underset{\sim}{p}$ is the part of $\underset{\sim}{x}$ that is in the 3-direction, and $\underset{\sim}{q}$ is the part of $\underset{\sim}{x}$ that is in the 1-2 plane. In many engineering applications, there exist multiple planes to which vectors are projected, so it is not always possible to merely set up a single basis that is lined up with all of the planes of interest — that's why our direct notation equations (6.11) and (6.12) are more convenient. Furthermore, the vector $\underset{\sim}{x}$ and the plane's normal $\underset{\sim}{n}$ might be changing in time. In this case, the direct notation formula of Eq. (6.12) is most convenient because it allows direct computation of rates.

The concept of a projection will now be generalized to permit us to find the "late afternoon shadow" of the vector $\underset{\sim}{x}$ when the "light rays" are oblique to the plane.

Rank-2 oblique linear projection

GOAL: Explain oblique (not nearest point) projections onto a plane.

Figure 6.2(b) showed how to find the shadow of a vector at "high noon," when the sun was directly overhead so that the light rays were perpendicular to the target plane. As sketched in Fig. 6.3, a more general kind of projection allows the "light rays" to intersect the plane at an *oblique* angle. Now we seek the formula for the shadow of $\underset{\sim}{x}$ in the late afternoon, when the sun is shining from an angle. This kind of projection can be characterized via two vectors $\underset{\sim}{a}$ and $\underset{\sim}{b}$. The target plane is defined to be perpendicular to $\underset{\sim}{b}$, and the light rays are parallel to $\underset{\sim}{a}$. The magnitudes of these vectors are inconsequential since they are only being used to define *orientations* (see page 75).

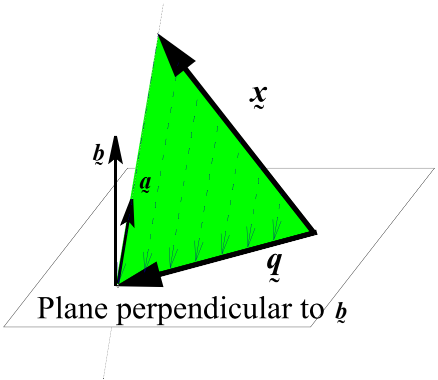


Figure 6.3. Oblique projection. The paths of projection (dashed lines) are all parallel to $\underset{\sim}{a}$, so they obliquely intersect the plane. The result for $\underset{\sim}{q}$ is unaffected by translation of the plane.

As before, we seek an expression for the vector $\underset{\sim}{q}$ that is the projection of $\underset{\sim}{x}$ onto the plane. This time, however, we want the projection direction (the "light rays") to be aligned with a different vector $\underset{\sim}{a}$. Referring to Fig. 6.3, we can see that there must exist a scalar coefficient such that the vector $\underset{\sim}{x}$ can be written as the sum of $\underset{\sim}{q}$ plus some (as yet) unknown multiple of $\underset{\sim}{a}$:

$$\underset{\sim}{x} = \underset{\sim}{q} + \eta \underset{\sim}{a}, \tag{6.15}$$

To find the value of η, we impose the condition that the vector $\underset{\sim}{q}$ must lie in the plane. In other words, $\underset{\sim}{b} \bullet \underset{\sim}{q}$ must be zero if we want $\underset{\sim}{q}$ to be perpendicular to $\underset{\sim}{b}$. Dotting both sides of Eq. (6.15) by $\underset{\sim}{b}$ (and setting $\underset{\sim}{b} \bullet \underset{\sim}{q} = 0$) gives

$$\underset{\sim}{b} \bullet \underset{\sim}{x} = 0 + \eta(\underset{\sim}{b} \bullet \underset{\sim}{a}). \tag{6.16}$$

Solving for η and substituting the result into (6.15) gives the desired formula for the oblique projection:

$$\underset{\sim}{q} = Q(\underset{\sim}{x})\text{, where } \boxed{Q(\underset{\sim}{x}) = \underset{\sim}{x} - \frac{\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{x})}{\underset{\sim}{a} \bullet \underset{\sim}{b}}}. \tag{6.17}$$

Naturally, Eq. (6.12) is a special case of the more general Eq. (6.17), obtained by choosing $\underset{\sim}{a}=\underset{\sim}{b}=\underset{\sim}{n}$. In other words, the projection is orthogonal only if $\underset{\sim}{a}$ is proportional to $\underset{\sim}{b}$.

Rank-1 oblique linear projection

GOAL: Explain oblique projections onto a second vector.

The remaining part of $\underset{\sim}{x}$ is obtained by a rank 1 projection. Namely, $\underset{\sim}{p} = \underset{\sim}{x} - \underset{\sim}{q}$. Thus,

$$\underset{\sim}{p} = P(\underset{\sim}{x})\text{, where } \boxed{P(\underset{\sim}{x}) = \frac{\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{x})}{\underset{\sim}{a} \bullet \underset{\sim}{b}}}. \tag{6.18}$$

The result of this operation will always be a vector that is simply a scalar multiple of $\underset{\sim}{a}$. This target space is one dimensional, so the projector P has rank 1.

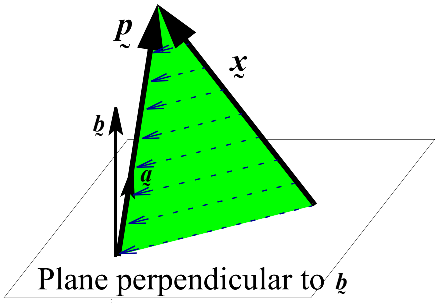


Figure 6.4. Rank-1 oblique projection. The path obliquely intersects the plane.

Note that the operator P has the property that $P(P(\underset{\sim}{x})) = P(\underset{\sim}{x})$. Physically, this means that projecting a vector that is *already in the target space* will just give you back that vector unchanged (your shadow has no shadow other than itself). This property is, in fact, used to define the term "projection." The operator Q has the similar property that $Q(Q(\underset{\sim}{x})) = Q(\underset{\sim}{x})$, so it is therefore a projector.

Degenerate (trivial) Rank-0 linear projection

GOAL: Explain that the zero operator is a degenerate projector.

Consider a function that does nothing but return the zero vector:

$$P(\underset{\sim}{x}) = \underset{\sim}{0} \tag{6.19}$$

As a special case, note that

$$P(\underset{\sim}{0}) = \underset{\sim}{0} \tag{6.20}$$

Though somewhat silly-seeming, we can then state that

$$P(\underset{\sim}{x}) = P(\underset{\sim}{0}) \tag{6.21}$$

Applying the zero operator a second time to both sides of Eq. (6.19) gives

$$P(P(\underset{\sim}{x})) = P(\underset{\sim}{0}) \tag{6.22}$$

or, using Eq. (6.21),

$$P(P(\underset{\sim}{x})) = P(\underset{\sim}{x}) \tag{6.23}$$

Hence, Eq. (6.12) permits us to assert that the uninteresting zero operator is a projection. By its definition, it is also a linear operator and its range space, trivially, contains only the zero vector. Hence the zero operator is a rank-0 projection because its range space is zero-dimensional (a single point in space).

Degenerate (trivial) Rank-3 projection in 3D space

GOAL: Explain that the identity operator is also a projection.

Consider the identity operator:

$$P(\underset{\sim}{x}) = \underset{\sim}{x} \tag{6.24}$$

Applying this operator a second time to both sides of this equation gives

$$P(P(\underset{\sim}{x})) = P(\underset{\sim}{x}) \tag{6.25}$$

Hence, trivially, the identity operation is a projection according to the definition in Eq. (6.12). Since we are considering vectors in three-dimensional space, the range of the identity operation is 3D space, making this a rank-3 projection.

Complementary projectors

GOAL: define, begin introducing concepts needed for the projection theorem

The operator P is called the complement of Q. In general, two operators P and Q are called complementary projectors if the following properties hold

$$P(P(\underset{\sim}{x})) = P(\underset{\sim}{x}) \tag{6.26}$$

$$Q(Q(\underset{\sim}{x})) = Q(\underset{\sim}{x}) \tag{6.27}$$

$$Q(P(\underset{\sim}{x})) = P(Q(\underset{\sim}{x})) = \underset{\sim}{0} \tag{6.28}$$

$$P(\underset{\sim}{x}) + Q(\underset{\sim}{x}) = \underset{\sim}{x} \tag{6.29}$$

The last property states that the sum of P and Q must be the identity operation (this is an abstract way of saying that the P and Q operators permit an additive *decomposition* of the vector $\underset{\sim}{x}$). The rank of P plus the rank of Q must sum to equal the dimension of the space, which is 3 for ordinary physical vectors.

Normalized versions of the projectors

GOAL: Generalize the "aligned" basis description of a projector to show that the structure is very similar for oblique projections except that the "aligned" basis is non-orthonormal.

The projectors Q and P in Eqs. (6.17) and (6.18) are homogeneous of degree zero with respect to $\underset{\sim}{a}$ and/or $\underset{\sim}{b}$. (*i.e.,* multiplying either of these vectors by any nonzero scalar does not affect the outcome of the operations). This property does *not* mean that the projection formulas depend on only the angle between $\underset{\sim}{a}$ and $\underset{\sim}{b}$. The scaling property simply implies that the formulas are insensitive to the magnitudes and senses of $\underset{\sim}{a}$ and $\underset{\sim}{b}$ — only their *orientations* matter (see page 75).

Because the magnitudes of $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{b}}$ are arbitrary, it is convenient in certain applications to replace them with $\hat{\underset{\sim}{\boldsymbol{a}}}$ and $\hat{\underset{\sim}{\boldsymbol{b}}}$, which have the same orientations as $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{b}}$, but whose magnitudes satisfy

$$\hat{\underset{\sim}{\boldsymbol{a}}} \bullet \hat{\underset{\sim}{\boldsymbol{b}}}=1 \qquad (6.30)$$

One possible way to construct $\hat{\underset{\sim}{\boldsymbol{a}}}$ and $\hat{\underset{\sim}{\boldsymbol{b}}}$ is*

$$\hat{\underset{\sim}{\boldsymbol{a}}} = \frac{\underset{\sim}{\boldsymbol{a}}}{\sqrt{\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{b}}}} \quad \text{and} \quad \hat{\underset{\sim}{\boldsymbol{b}}} = \frac{\underset{\sim}{\boldsymbol{b}}}{\sqrt{\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{b}}}} \qquad (6.31)$$

When the property of Eq. (6.30) holds, $\hat{\underset{\sim}{\boldsymbol{a}}}$ and $\hat{\underset{\sim}{\boldsymbol{b}}}$ are called “dual” vectors. Using the dual vectors, Eq. (6.17) simplifies to

$$\boxed{\underset{\sim}{\boldsymbol{q}} = \underset{\sim}{\boldsymbol{x}} - \hat{\underset{\sim}{\boldsymbol{a}}}(\hat{\underset{\sim}{\boldsymbol{b}}} \bullet \underset{\sim}{\boldsymbol{x}})} \qquad (6.32)$$

The component form of Eq. (6.17) take a particularly simple form when expressed using a *nonorthogonal* basis having the covariant† base vector $\underset{\sim}{\boldsymbol{g}}_3 = \hat{\underset{\sim}{\boldsymbol{a}}}$ and the contravariant base vector $\underset{\sim}{\boldsymbol{g}}^3 = \hat{\underset{\sim}{\boldsymbol{b}}}$. Namely

$$\underset{\sim}{\boldsymbol{q}} = x^1\underset{\sim}{\boldsymbol{g}}_1 + x^2\underset{\sim}{\boldsymbol{g}}_2, \qquad (6.33)$$

This result is strongly analogous to Eq. (6.14), with the key difference being that the $\underset{\sim}{\boldsymbol{g}}_k$ base vectors are nonorthogonal and not generally of unit length. The projector Q is a rank-2 projector because its range space is two dimensional.

A vector that is pointing *directly* at the sun will have no shadow at all. The **null space** of the projector Q is the set of all $\underset{\sim}{\boldsymbol{x}}$ for which $Q(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{0}}$. Setting $Q(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{0}}$ in Eq. (6.17) reveals that the null space of that particular projector is the set of vectors that are proportional to $\underset{\sim}{\boldsymbol{a}}$. Thus, the null space of the projector is one-dimensional. The range space (i.e., the target plane) is two dimensional. Since P and Q are complementary projectors, the null space of one is the range space of the other! Later on, projectors will be defined for higher-dimensional spaces and the null spaces will be more interesting.

* This formula applies when $\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{b}} > 0$. For $\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{b}} < 0$, then $\underset{\sim}{\boldsymbol{b}}$ should be replaced by $-\underset{\sim}{\boldsymbol{b}}$ before applying this formula.

† When a basis $\{\underset{\sim}{\boldsymbol{g}}_1, \underset{\sim}{\boldsymbol{g}}_2, \underset{\sim}{\boldsymbol{g}}_3\}$ is nonorthonormal, there naturally arises a useful companion or “dual” basis denoted $\{\underset{\sim}{\boldsymbol{g}}^1, \underset{\sim}{\boldsymbol{g}}^2, \underset{\sim}{\boldsymbol{g}}^3\}$ which is defined such that $\underset{\sim}{\boldsymbol{g}}^i \bullet \underset{\sim}{\boldsymbol{g}}_j = \delta^i_j$, where δ^i_j is the Kronecker delta of Eq. (5.9). Thus, for example, $\underset{\sim}{\boldsymbol{g}}^1$ is perpendicular to the plane formed by $\underset{\sim}{\boldsymbol{g}}_2$ and $\underset{\sim}{\boldsymbol{g}}_3$. A vector can be expressed in terms of either basis as $\underset{\sim}{\boldsymbol{v}} = v^i\underset{\sim}{\boldsymbol{g}}_i = v_i\underset{\sim}{\boldsymbol{g}}^i$. Subscripted quantities are called “**covariant**” and superscripted quantities are “**contravariant**.” (A cute mnemonic is “co-go-below”). Non-orthonormal tensor analysis is reviewed in Refs. [7,6].

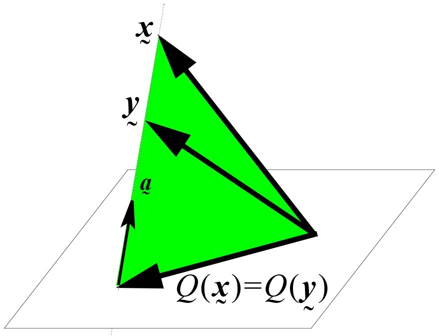


Figure 6.5. Projections of two vectors along a an obliquely oriented line.

Figure (6.5) shows two vectors, $\underset{\sim}{x}$ and $\underset{\sim}{y}$, that fall on the line defined by $\underset{\sim}{a}$. More precisely, for some scalar β,

$$\underset{\sim}{x} = \underset{\sim}{y} + \beta\underset{\sim}{a}\,. \qquad (6.34)$$

As seen in the sketch (or as verified by direct substitution into Eq. (6.17)),

$$Q(\underset{\sim}{x})=Q(\underset{\sim}{y})\,. \qquad (6.35)$$

Conversely, if (6.35) holds, then so does (6.34). This property of projections is the key to validating radial or oblique return algorithms in the field of plasticity.

Expressing a vector as a linear combination of three arbitrary (not necessarily orthonormal) vectors.

GOAL: outline most straightforward process, refine notation for the process to be more consistent with curvilinear notation.

Any vector $\underset{\sim}{x}$ can always be expressed as a linear combination of three other vectors $\underset{\sim}{a}$, $\underset{\sim}{b}$, and $\underset{\sim}{c}$ *as long as those three vectors are linearly independent (i.e., so long as the triple scalar product is nonzero:* $[\underset{\sim}{a}, \underset{\sim}{b}, \underset{\sim}{c}] \neq 0$*)*. If so, then we know there exist scalar multiples α, β, and γ such that

$$\underset{\sim}{x} = \alpha\underset{\sim}{a} + \beta\underset{\sim}{b} + \gamma\underset{\sim}{c} \qquad (6.36)$$

The brute force simplest way to determine values of the scalar coefficients is to simply dot both sides of this equation by $\underset{\sim}{a}$ to obtain

$$\underset{\sim}{x} \bullet \underset{\sim}{a} = \alpha(\underset{\sim}{a} \bullet \underset{\sim}{a}) + \beta(\underset{\sim}{b} \bullet \underset{\sim}{a}) + \gamma(\underset{\sim}{c} \bullet \underset{\sim}{a}) \qquad (6.37)$$

We can similarly dot both sides of Eq. (6.36) by $\underset{\sim}{b}$ and then separately by $\underset{\sim}{c}$ to obtain two additional equations which, with Eq. (6.37) forms a set of three equations and three unknowns (the $\{\alpha, \beta, \gamma\}$ coefficients). The system is solvable if and only if the original three vectors are linearly independent. We will now discuss this identical process using a slight change of notation so that it become more clear that each component actually corresponds to a rank-1 *oblique* projection of the vector $\underset{\sim}{x}$ onto the three vectors. To simplify the notation, we will now denote $\underset{\sim}{a}$, $\underset{\sim}{b}$, and $\underset{\sim}{c}$ by three different symbols, $\{\underset{\sim}{g}_1, \underset{\sim}{g}_2, \underset{\sim}{g}_3\}$.

The right-hand side of Fig. 6.1 showed how an arbitrary vector $\underset{\sim}{\boldsymbol{x}}$ may be decomposed into the sum of parts that are aligned with the *orthonormal* $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ basis. However, a basis does not *have* to be orthonormal — any three linearly independent vectors $\{\underset{\sim}{\boldsymbol{g}}_1, \underset{\sim}{\boldsymbol{g}}_2, \underset{\sim}{\boldsymbol{g}}_3\}$ may be alternatively used as a basis. We wish to additively decompose a vector $\underset{\sim}{\boldsymbol{x}}$ into parts aligned with these three non-orthonormal base vectors. Stated differently, we wish to discover a path from the tail to the tip of $\underset{\sim}{\boldsymbol{x}}$ that always travels parallel (or anti-parallel) to these three base vectors (See Fig. 6.6).

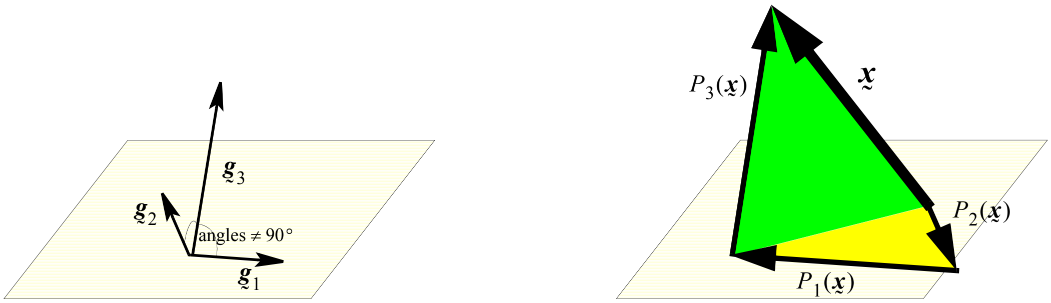

$\underset{\sim}{\boldsymbol{g}}_3$
$\underset{\sim}{\boldsymbol{g}}_2$
angles ≠ 90°
$\underset{\sim}{\boldsymbol{g}}_1$
$P_3(\underset{\sim}{\boldsymbol{x}})$
$\underset{\sim}{\boldsymbol{x}}$
$P_2(\underset{\sim}{\boldsymbol{x}})$
$P_1(\underset{\sim}{\boldsymbol{x}})$

Figure 6.6. Three oblique projections. The indicated plane contains $\underset{\sim}{\boldsymbol{g}}_1$ and $\underset{\sim}{\boldsymbol{g}}_2$, while $\underset{\sim}{\boldsymbol{g}}_3$ forms an oblique angle to the plane. The dual vector $\underset{\sim}{\boldsymbol{g}}^3$ is proportional to $\underset{\sim}{\boldsymbol{g}}_1 \times \underset{\sim}{\boldsymbol{g}}_2$, so it may be used to define the normal to the plane.

First, in order for the three $\{\underset{\sim}{\boldsymbol{g}}_1, \underset{\sim}{\boldsymbol{g}}_2, \underset{\sim}{\boldsymbol{g}}_3\}$ vectors to be linearly independent (and therefore form a basis), they must have a positive triple scalar product:

$$J_o \equiv [\underset{\sim}{\boldsymbol{g}}_1, \underset{\sim}{\boldsymbol{g}}_2, \underset{\sim}{\boldsymbol{g}}_3] = (\underset{\sim}{\boldsymbol{g}}_1 \times \underset{\sim}{\boldsymbol{g}}_2) \bullet \underset{\sim}{\boldsymbol{g}}_3 \quad > 0 \tag{6.38}$$

Corresponding to the $\{\underset{\sim}{\boldsymbol{g}}_1, \underset{\sim}{\boldsymbol{g}}_2, \underset{\sim}{\boldsymbol{g}}_3\}$ base vectors, there exists a unique “dual” basis $\{\underset{\sim}{\boldsymbol{g}}^1, \underset{\sim}{\boldsymbol{g}}^2, \underset{\sim}{\boldsymbol{g}}^3\}$ such that

$$\underset{\sim}{\boldsymbol{g}}_i \bullet \underset{\sim}{\boldsymbol{g}}^j = \delta_i^j \qquad \text{(where } \delta_i^j \text{ is an alternative symbol for the Kronecker delta).} \tag{6.39}$$

In other words, $\underset{\sim}{\boldsymbol{g}}^1$ must be perpendicular to $\underset{\sim}{\boldsymbol{g}}_2$ and $\underset{\sim}{\boldsymbol{g}}_3$, so it must be proportional to $(\underset{\sim}{\boldsymbol{g}}_2 \times \underset{\sim}{\boldsymbol{g}}_3)$. The constant of proportionality must be set so that $\underset{\sim}{\boldsymbol{g}}_1 \bullet \underset{\sim}{\boldsymbol{g}}^1 = 1$. Thus, as can be verified by direct substitution, the dual basis must be given by

$$\underset{\sim}{\boldsymbol{g}}^1 = \frac{1}{J_o}(\underset{\sim}{\boldsymbol{g}}_2 \times \underset{\sim}{\boldsymbol{g}}_3),\ \underset{\sim}{\boldsymbol{g}}^2 = \frac{1}{J_o}(\underset{\sim}{\boldsymbol{g}}_3 \times \underset{\sim}{\boldsymbol{g}}_1),\ \underset{\sim}{\boldsymbol{g}}^3 = \frac{1}{J_o}(\underset{\sim}{\boldsymbol{g}}_1 \times \underset{\sim}{\boldsymbol{g}}_2) \tag{6.40}$$

Without proof, we claim that the part of $\underset{\sim}{\boldsymbol{x}}$ in the direction of $\underset{\sim}{\boldsymbol{g}}_k$ is given by an oblique projection operation P_k defined by

$$P_k(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{g}}_k(\underset{\sim}{\boldsymbol{g}}^k \bullet \underset{\sim}{\boldsymbol{x}}),\ \text{with no summation on } k. \tag{6.41}$$

The three P_k operators are complementary projections because they satisfy the following properties:

$$P_i(P_j(\underset{\sim}{\boldsymbol{x}})) = \begin{cases} P_i(\underset{\sim}{\boldsymbol{x}}) & \text{if } i{=}j \\ \underset{\sim}{\boldsymbol{0}} & \text{if } i \neq j \end{cases} \tag{6.42}$$

and

$$P_1(\underset{\sim}{x}) + P_2(\underset{\sim}{x}) + P_3(\underset{\sim}{x}) = \underset{\sim}{x} \tag{6.43}$$

Generalized projections

GOAL: Reiterate the mathematician’s definition of the term “projection” (idempotent)

A transformation P is called a projection (or “idempotent”) if and only if

$$P(P(\underset{\sim}{x})) = P(\underset{\sim}{x}), \tag{6.44}$$

which merely states that projecting a vector that has already been projected will give you the same projected vector right back. Consider, for example, the function of Eq. (6.17):

$$P(\underset{\sim}{x}) = \underset{\sim}{x} - \frac{\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{x})}{\underset{\sim}{a} \bullet \underset{\sim}{b}} \tag{6.45}$$

Physically, we know that this function is a projector. To prove it rigorously, we operate on both sides of Eq. (6.45) by P to obtain

$$P(P(\underset{\sim}{x})) = \left[\underset{\sim}{x} - \frac{\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{x})}{\underset{\sim}{a} \bullet \underset{\sim}{b}}\right] - \frac{\underset{\sim}{a}\left(\underset{\sim}{b} \bullet \left[\underset{\sim}{x} - \frac{\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{x})}{\underset{\sim}{a} \bullet \underset{\sim}{b}}\right]\right)}{\underset{\sim}{a} \bullet \underset{\sim}{b}} \tag{6.46}$$

Simplification shows that the right hand side reduces back to $P(\underset{\sim}{x})$. Hence, Eq. (6.44) holds and P is indeed a projection.

Linear projections

GOAL: Define very special (linear) class of projectors, explain that oblique projectors are <u>also</u> linear.

A projection is linear if and only if

$$\begin{aligned} &\textit{(i)}\quad P(\alpha \underset{\sim}{x}) = \alpha P(\underset{\sim}{x}), \text{ for all } \alpha, \text{ and} \\ &\textit{(ii)}\quad P(\underset{\sim}{x}_1 + \underset{\sim}{x}_2) = P(\underset{\sim}{x}_1) + P(\underset{\sim}{x}_2) \text{ for all } \underset{\sim}{x}_1 \text{ and } \underset{\sim}{x}_2. \end{aligned} \tag{6.47}$$

The projection of Eq. (6.45) is easily seen to be linear even though it is oblique. Likewise, Eq. (6.41) is linear. Physically, a projection is linear if one can take the projection of a linear combination of vectors and the result is the same as you would get if you instead *first* projected the individual vectors and *then* took their linear combination.

Nonlinear projections

GOAL: To clarify <u>linear</u> projections, give examples of some <u>nonlinear</u> projections.

The vector “signum” function. An example of a *nonlinear* projection is

$$P(\underset{\sim}{x}) = \frac{\underset{\sim}{x}}{\sqrt{\underset{\sim}{x} \bullet \underset{\sim}{x}}} \qquad (\text{or } P(\underset{\sim}{x}) = \underset{\sim}{0} \text{ if } \underset{\sim}{x} = \underset{\sim}{0}) \tag{6.48}$$

This function is indeed a projection because Eq. (6.44) holds. The first linearity test *(i)* in Eq. (6.47) fails because, for example, $P(2\underset{\sim}{x}) \neq 2P(\underset{\sim}{x})$, so this projection is nonlinear. Geometrically, the above operator transforms a vector to a unit vector in the same direction. Hence, the range set for this transformation consists of vectors pointing from the origin to a point on the unit sphere. The transformation of Eq. (6.48) is non-linear because the range (the target surface of the sphere) is geometrically curvilinear. The level lines (i.e., the "light rays") are straight lines that emanate radially from the origin.

The vector signum operation always returns a unit vector in the direction of the starting vector, or the zero vector if the starting vector is zero. The signum operation is sometimes more compactly denoted with a "hat" so that

$$\hat{\underset{\sim}{x}} = \frac{\underset{\sim}{x}}{\sqrt{\underset{\sim}{x} \bullet \underset{\sim}{x}}} \qquad \text{(or } \hat{\underset{\sim}{x}}=\underset{\sim}{0} \text{ if } \underset{\sim}{x}=\underset{\sim}{0}\text{)} \tag{6.49}$$

For lengthy operands, the "hat" signum operation is sometimes abbreviated "sgn" or is indicated by a "hat" as a superscript on the end of an expression. For example,

$$\operatorname{sgn}(\underset{\sim}{a}+\underset{\sim}{b}) = \frac{\underset{\sim}{a}+\underset{\sim}{b}}{\|\underset{\sim}{a}+\underset{\sim}{b}\|} \qquad \text{or} \qquad \operatorname{sgn}(\underset{\sim}{a}+\underset{\sim}{b})=\underset{\sim}{0} \text{ if } (\underset{\sim}{a}+\underset{\sim}{b})=\underset{\sim}{0} \tag{6.50}$$

Gravitational (distorted light ray) projections. Nonlinear projectors can also be constructed for which the range is a linear manifold (i.e., a line or a plane) but the *path* to this range space is curvilinear. For example, the transformation

$$P(\underset{\sim}{x}) = \frac{[\underset{\sim}{x} \bullet \underset{\sim}{a} - (\underset{\sim}{x} \bullet \underset{\sim}{b})^2]\underset{\sim}{a}}{\underset{\sim}{a} \bullet (\underset{\sim}{a} - \underset{\sim}{b})} \tag{6.51}$$

is a projection because $P(P(\underset{\sim}{x})) = P(\underset{\sim}{x})$. It has a linear range space (vectors parallel to $\underset{\sim}{a}$), but the projection is nonlinear because the paths to the range space (i.e., the level lines) are curvilinear. The light rays are *bending* as they approach the target space.

A very important nonlinear projection transformation from the field of continuum mechanics is the polar decomposition (see page 206). Rather than being additive, this decomposition is multiplicative.

Self-adjoint projections

GOAL: Show that orthogonal (nearest point) projections are self-adjoint, whereas oblique projections are not. Set stage for later showing that orthogonal projection tensors are symmetric, whereas oblique projection tensors are not.

A projection P is self-adjoint if and only if

$$\underset{\sim}{u} \bullet P(\underset{\sim}{v}) = \underset{\sim}{v} \bullet P(\underset{\sim}{u}) \text{ for all } \underset{\sim}{u} \text{ and } \underset{\sim}{v}. \tag{6.52}$$

Applying this definition to Eq. (6.45) gives

$$\underset{\sim}{u} \bullet \left[\underset{\sim}{v} - \frac{\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{v})}{\underset{\sim}{a} \bullet \underset{\sim}{b}}\right] = \underset{\sim}{v} \bullet \left[\underset{\sim}{u} - \frac{\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{u})}{\underset{\sim}{a} \bullet \underset{\sim}{b}}\right] \tag{6.53}$$

Simplification gives

$$(\underset{\sim}{u} \bullet \underset{\sim}{a})(\underset{\sim}{b} \bullet \underset{\sim}{v}) = (\underset{\sim}{v} \bullet \underset{\sim}{a})(\underset{\sim}{b} \bullet \underset{\sim}{u}) \text{ for all } \underset{\sim}{u} \text{ and } \underset{\sim}{v}. \tag{6.54}$$

This condition can be true only if $\underset{\sim}{a}$ is proportional to $\underset{\sim}{b}$,* in which case the projection reduces to the orthogonal (nearest point, high noon) projection of Eq. (6.12). The P_k projectors defined in Eq. (6.41) are *not* self-adjoint unless the $\{\underset{\sim}{g}_k\}$ basis is orthonormal. Physically, self-adjoint projections correspond to projections to the *nearest point* in the target space (like a shadow at high-noon), whereas non-self-adjoint projections are oblique (like a shadow in late afternoon).

Later on, we will see that linear projectors can be represented through the use of projection tensors. Then we will find that a linear projector is self-adjoint only if its projection tensor is symmetric.

Gram-Schmidt orthogonalization

GOAL: show how to convert a set of vectors into a minimal orthonormal basis that will span the same space as the original set of vectors.

Suppose that you are working on a problem involving a collection of ordinary engineering vectors. For illustration, suppose that your collection contains four vectors $\{\underset{\sim}{v}_1, \underset{\sim}{v}_2, \underset{\sim}{v}_3, \underset{\sim}{v}_4\}$. Our goal is to construct an orthonormal basis for the span of any collection of vectors, regardless of whether or not they are linearly independent. One process for doing this is called Gram-Schmidt orthogonalization, and it involves simply a sequence of projections and normalizations as outlined in the following step-by-step sequence of calculations:

$$\underset{\sim}{n}_1 = \text{sgn}[\underset{\sim}{v}_1] \tag{6.55}$$

$$\underset{\sim}{n}_2 = \text{sgn}[\underset{\sim}{v}_2 - (\underset{\sim}{v}_2 \bullet \underset{\sim}{n}_1)\underset{\sim}{n}_1] \tag{6.56}$$

$$\underset{\sim}{n}_3 = \text{sgn}[\underset{\sim}{v}_3 - (\underset{\sim}{v}_3 \bullet \underset{\sim}{n}_1)\underset{\sim}{n}_1 - (\underset{\sim}{v}_3 \bullet \underset{\sim}{n}_2)\underset{\sim}{n}_2] \tag{6.57}$$

$$\underset{\sim}{n}_4 = \text{sgn}[\underset{\sim}{v}_4 - (\underset{\sim}{v}_4 \bullet \underset{\sim}{n}_1)\underset{\sim}{n}_1 - (\underset{\sim}{v}_4 \bullet \underset{\sim}{n}_2)\underset{\sim}{n}_2 - (\underset{\sim}{v}_4 \bullet \underset{\sim}{n}_3)\underset{\sim}{n}_3] \tag{6.58}$$

Recalling the definition of the vector signum function “sgn” given in Eq. (6.50), we note that the first of these equations normalizes the first vector $\underset{\sim}{v}_1$ (or simply gives zero if $\underset{\sim}{v}_1$ is zero). Note the similarity between Eq. (6.56) and Eq. (6.12), which shows that $\underset{\sim}{n}_2$ is simply the (normalized) part of $\underset{\sim}{v}_2$ perpendicular to $\underset{\sim}{n}_1$. The remaining equations cited are similar; they systematically create vectors that have a zero dot product with the preceding vectors. Since our example involved *four* starting vectors, we know that at *least* one — possibly more — of the above $\underset{\sim}{n}_k$ vectors will turn out to be zero. Even if the starting collection had contained only three vectors, it’s still possible for one of the $\underset{\sim}{n}_k$ vectors to come out to be zero. The appearance of a zero $\underset{\sim}{n}_k$ vector is merely an inconsequential byproduct that occurs whenever the starting collection not linearly independent. The *non-zero* $\underset{\sim}{n}_k$ vectors form a basis for the span of the original starting set of vectors. Therefore, the number of nonzero $\underset{\sim}{n}_k$ vectors equals the dimension of this span.

* To prove this claim, take $\underset{\sim}{u} \perp \underset{\sim}{b}$ and $\underset{\sim}{v} = \underset{\sim}{b}$ to obtain $(\underset{\sim}{u} \bullet \underset{\sim}{a})(\underset{\sim}{b} \bullet \underset{\sim}{b}) = 0$ or, since $\underset{\sim}{b} \neq \underset{\sim}{0}$ by implied premise, $\underset{\sim}{u} \bullet \underset{\sim}{a} = 0$. In other words, $\underset{\sim}{u} \perp \underset{\sim}{a}$ for all $\underset{\sim}{u} \perp \underset{\sim}{b}$. Thus, $\underset{\sim}{a}$ must be proportional to $\underset{\sim}{b}$.

Special case: orthogonalization of two vectors. One distasteful aspect of the Gram-Schmidt orthogonalization process is that the resulting basis depends on the ordering the starting vectors. There's nothing particularly wrong with this. To orthogonalize *two* vectors, $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$ in a less biased manner, you might consider the following alternative approach:

$$\text{Define } \underset{\sim}{\boldsymbol{m}} = \frac{\underset{\sim}{\boldsymbol{u}}}{||\underset{\sim}{\boldsymbol{u}}||} \text{ and } \underset{\sim}{\boldsymbol{n}} = \frac{\underset{\sim}{\boldsymbol{v}}}{||\underset{\sim}{\boldsymbol{v}}||} \tag{6.59}$$

Then an orthonormal basis can be defined as

$$\underset{\sim}{\boldsymbol{e}}_1 \equiv \frac{\underset{\sim}{\boldsymbol{m}} + \underset{\sim}{\boldsymbol{n}}}{||\underset{\sim}{\boldsymbol{m}} + \underset{\sim}{\boldsymbol{n}}||} \qquad \text{and} \qquad \underset{\sim}{\boldsymbol{e}}_2 \equiv \frac{\underset{\sim}{\boldsymbol{m}} - \underset{\sim}{\boldsymbol{n}}}{||\underset{\sim}{\boldsymbol{m}} - \underset{\sim}{\boldsymbol{n}}||} \tag{6.60}$$

There remains a small amount of bias here because exchanging $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$ will change the direction of $\underset{\sim}{\boldsymbol{e}}_2$.

The projection theorem

Some of the most profoundly useful engineering theorems merely state intuitively obvious concepts in a more useful (but less obvious) mathematical form.*

The projection theorem is another mathematical statement of a physically "obvious" concept. Namely, you can always break a vector into a part in a desired direction plus *whatever is left over.* Now let's state this idea mathematically.

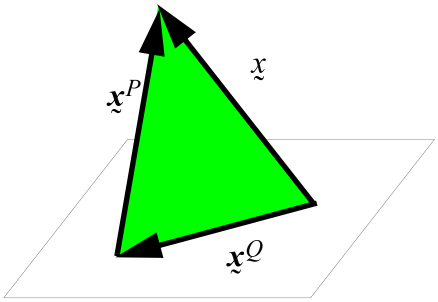

$\underset{\sim}{x}$
$\underset{\sim}{\boldsymbol{x}}^P$
$\underset{\sim}{\boldsymbol{x}}^Q$

Figure 6.7. Oblique projection. The path obliquely intersects the plane.

Given a projector P, the projection theorem states that any vector $\underset{\sim}{\boldsymbol{x}}$ can always be decomposed uniquely as

$$\underset{\sim}{\boldsymbol{x}} = \underset{\sim}{\boldsymbol{x}}^P + \underset{\sim}{\boldsymbol{x}}^Q \tag{6.61}$$

such that $\underset{\sim}{\boldsymbol{x}}^P$ is in the range space of the projector P and $\underset{\sim}{\boldsymbol{x}}^Q$ is in the null-space of the projector P. By this we mean that $\underset{\sim}{\boldsymbol{x}}^P$ and $\underset{\sim}{\boldsymbol{x}}^Q$ will satisfy

$$P(\underset{\sim}{\boldsymbol{x}}^P) = \underset{\sim}{\boldsymbol{x}}^P \tag{6.62}$$

$$P(\underset{\sim}{\boldsymbol{x}}^Q) = \underset{\sim}{\boldsymbol{0}} \tag{6.63}$$

The vector $\underset{\sim}{\boldsymbol{x}}^P$ is the projection of $\underset{\sim}{\boldsymbol{x}}$, and $\underset{\sim}{\boldsymbol{x}}^Q$ is simply the part of $\underset{\sim}{\boldsymbol{x}}$ that is "left over" after taking away $\underset{\sim}{\boldsymbol{x}}^P$. Specifically,

$$\underset{\sim}{\boldsymbol{x}}^P = P(\underset{\sim}{\boldsymbol{x}}) \tag{6.64}$$

$$\underset{\sim}{\boldsymbol{x}}^Q = Q(\underset{\sim}{\boldsymbol{x}}), \tag{6.65}$$

* For example, the principle of conservation of mass merely asserts that "what goes in must come out or stay there." The mathematical representation of this statement is more useful, but less obvious.

where

$$Q(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{x}} - P(\underset{\sim}{\boldsymbol{x}}) \tag{6.66}$$

The function Q is *itself* a projector. Some key properties of these "**complementary**" projectors are

$$\begin{aligned} &P(P(\underset{\sim}{\boldsymbol{x}})) = P(\underset{\sim}{\boldsymbol{x}}) \qquad Q(Q(\underset{\sim}{\boldsymbol{x}})) = Q(\underset{\sim}{\boldsymbol{x}}) \\ &P(Q(\underset{\sim}{\boldsymbol{x}})) = Q(P(\underset{\sim}{\boldsymbol{x}})) = \underset{\sim}{\boldsymbol{0}} \\ &P(\underset{\sim}{\boldsymbol{x}}) + Q(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{x}} \end{aligned} \tag{6.67}$$

Note that $P(Q(\underset{\sim}{\boldsymbol{x}}))$ is *not* the same thing as $\underset{\sim}{\boldsymbol{x}}^P \bullet \underset{\sim}{\boldsymbol{x}}^Q$. If the projection is oblique, then $\underset{\sim}{\boldsymbol{x}}^P \bullet \underset{\sim}{\boldsymbol{x}}^Q$ will be nonzero, but $P(Q(\underset{\sim}{\boldsymbol{x}}))$ will always be zero for complementary projectors. The transformation $P(\underset{\sim}{\boldsymbol{x}})$ is an orthogonal projection if and only if P is self-adjoint, in which case $\underset{\sim}{\boldsymbol{x}}^P \bullet \underset{\sim}{\boldsymbol{x}}^Q$ will also be zero.

The projection theorem is just the mathematical way to describe what is fairly obvious from looking at Fig. 6.7 (*i.e.,* a vector can be broken into its shadow plus whatever is left over). The projection theorem becomes quite useful (and less obvious) in higher dimensional spaces. Later in this book, we will introduce the notion of an abstract vector. Under this definition, we will find that matrices, differentiable functions and many other seemingly unrelated things behave just like vectors. The fact that a scalar function $f(x)$ can be written as the sum of its even plus odd parts is an application of the projection theorem. The fact that a matrix can be written as the sum of its symmetric plus skew-symmetric parts is an application of the projection theorem. The fact that a continuously differential function can be expressed as a Taylor series is the projection theorem (the basis is the infinite set $1, x, x^2, x^3, \ldots$).

Most material constitutive laws are expressible in terms of projections. For example, the projection theorem allows us to decompose stress and strain tensors into their deviatoric plus isotropic parts. Hooke's law of linear elasticity states that these individual parts are proportional to each other (i.e., pressure is proportional to volumetric strain and the stress deviator is proportional to the strain deviator). The proportionality constant is not necessarily equal for the two parts, so stress itself is not proportional to strain.

"No matter how much cats fight, there always seem to be plenty of kittens."
— Abraham Lincoln

7. Tensors

If you're sitting at a cocktail party with a bunch of engineers, physicists, and mathematicians, and you want to start a heated debate, just ask out loud: "What is a tensor?" One person will say that, for all practical purposes, a tensor is just a fancy word for a matrix.* Then someone else will pipe up indignantly and insist that a tensor is a linear transformation from vectors to vectors. Yet another person will say that a tensor is an ordered set of numbers that transform in a particular way upon a change of basis. Other folks (like us) will start babbling about "dyads" and "dyadics." In this chapter, we will touch on each of these various perspectives and urge you to adopt whatever definition of the term tensor you feel most comfortable with. As long as your definition *implies* the other guy's definition and vice versa, then either definition is acceptable.†

You can't study tensors without eventually (sometimes secretly) becoming enamoured of one particular definition of tensors. We will ease you into the notion of tensors by discussing how *engineering* tensors arise naturally when working with linear vector transformations. We will then describe our personal favorite (and less frequently adopted) definition of a tensor as being a sum of new abstract "objects" called "dyads." Admittedly, the dyad definition is not very satisfying or particularly intuitive, but we like it anyway because this definition requires no mention of a basis, *per se*, but it leads directly into the supplemental concept of a tensor basis, and it generalizes seamlessly to abstract, higher-dimensional, settings. From there, we will be poised to then cover the definition of an engineering tensor in terms of basis transformation rules.

* This is true for *some* practical purposes, but not for *all* practical purposes.

† Often, determining if one definition implies the other is a difficult task. The person who defines a tensor according to basis transformation rule is actually defining a *particular* class of tensors, whereas the definition in terms of linear transformations has broader abstract applicability, but becomes bogged down in the more subtle question: what is a vector? Answering this question for *engineering mechanics applications* eventually also comes around to the need to introduce basis transformation criteria, so in this arena, both definitions are equivalent. See page 227.

Analogy between tensors and other (more familiar) concepts

GOAL: Explain how vectors share several axiomatic properties in common with smooth scalar-valued functions. Then demonstrate that tensors are quite similar to smooth functions of two variables.

The mathematician's abstract definition of a vector, discussed on page 230, is elegantly crafted to call attention to the fact that certain seemingly unrelated sets of objects often share common properties. By studying sets with these properties in an abstract setting, mathematicians are able to derive numerous general results. Once done, they *know* that the results apply to any sets objects that obey those properties — no matter how unrelated those sets might appear to be at first glance.

This section will step through several observations about continuous smooth functions that you probably already know from your elementary algebra and calculus classes. Then, the parallel or analogous observations about vectors and tensors will be made in order to show the similarity between smooth functions and vectors. The goal here is to de-mystify the subject of tensor analysis by showing that it is very similar to other branches of mathematics where you are likely to be already comfortable.

Applying the mathematician's definition of a vector (page 230), it can be shown that the set of all real smooth[*] functions of *one* variable is itself a vector space, and we will denote it by C^1_∞. The subscript "∞" is used to tell you that this space is infinite dimensional[†] and the superscript "1" is telling you that the space is a first-order (i.e., vector) space. Any member of C^1_∞ is usually written $f(x)$, where x is permitted to take any value in the set of real numbers, and the actual symbol used for it is inconsequential. By analogy, we will denote the set of ordinary engineering vectors by the symbol V^1_3, where the subscript "3" indicates that engineering vectors are three-dimensional, and the superscript "1" indicates that the set V^1_3 is a *first*-order space (vectors). A member of V^1_3 is usually denoted in indicial form by v_i, where "i" is an index that is permitted to take any integer value from 1 to 3, and the actual symbol used for it is inconsequential. Note the analogy: the indicial notation v_i is analogous to the function notation $f(x)$. The vector index i, which takes values from 1 to 3, is analogous to the independent variable x, which takes values from $-\infty$ to ∞.

Applying the mathematician's definition (page 233), we can define the inner product between two smooth functions, $f \in C^1_\infty$ and $g \in C^1_\infty$ by

$$f \cdot g \equiv \int f(x) g(x) dx , \tag{7.1}$$

where the integration over the dummy variable x extends over the permissible range of x (namely from $x=-\infty$ to $x=+\infty$). By analogy, the inner product between two engineering vectors, $\underset{\sim}{f} \in V^1_3$ and $\underset{\sim}{g} \in V^1_3$ is defined

$$\underset{\sim}{f} \bullet \underset{\sim}{g} \equiv \sum f_i g_i , \tag{7.2}$$

[*] infinitely differentiable

[†] It's infinite dimensional because, if you tried to describe a function such as $y = x^2$ using a table of (x, y) pairs, your table would have to have an infinite number of entries.

where the summation over the dummy index "i" extends over the permissible range of "i" (namely from $i=1$ to $i=3$). Note the similarity between Eqs. (7.1) and (7.2). One involves a continuous summation (i.e., integration) over the independent variable and the other involves a discrete summation over the index.

We know that there exist coefficients α_i such that any smooth function can be expressed in the form of a power (Taylor) series:

$$f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \ldots \tag{7.3}$$

Viewed differently, this tells us that the set of smooth functions $\{x^k\}$, for $k = 0, 1, 2, \ldots$, forms a *basis* for C^1_∞. Any smooth function can be written as a linear combination of this basis. The number of members of the basis equals the dimension of the space (infinite). Analogously, we know in 3D vector analysis that there exist coefficients f_i such that any vector $\underset{\sim}{\mathbf{f}}$ can be expressed in the form

$$\underset{\sim}{\mathbf{f}} = f_1 \underset{\sim}{\mathbf{e}}_1 + f_2 \underset{\sim}{\mathbf{e}}_2 + f_3 \underset{\sim}{\mathbf{e}}_3 \tag{7.4}$$

where the set $\{\underset{\sim}{\mathbf{e}}_i\}$, for i = 1, 2, and 3, forms a basis for V^1_3. Any engineering vector can be written as a linear combination of this basis. The number of base vectors equals the dimension of the space (three).

Any smooth function can alternatively be expressed in the form of a Fourier series, for which the associated basis consists of trig functions. The coefficients in a Fourier expansion are *not* equal to the coefficients that appear in Eq. (7.3), although knowledge of those coefficients can be used to deduce the correct coefficients in the Fourier expansion. Likewise, any engineering vector can alternatively be expressed as an expansion in terms of some alternative basis $\{\underset{\sim}{\mathbf{e}}^*_i\}$, for which the coefficients are *not* equal to those in Eq. (7.4), although knowledge of those coefficients can be used to deduce the correct coefficients with respect to the new basis. Note the analogy: changing from a power series to a trig series expansion of a smooth function is the same (in spirit) as performing a change of basis in 3D vector analysis.

When discussing the overall properties of a smooth function $f(x)$, people will often refer to it as simply "f"; this removal of the independent variable helps readers stay focused on the fact that the function *itself* is being discussed, not the result of applying the function to an independent argument. Of course, actual *calculations* or *applications* of the function f will require reintroduction of the independent argument. Similarly, when people speak about vectors, they often refer to them in structured (AKA symbolic, direct, or "Gibbs") notation as $\underset{\sim}{\mathbf{v}}$; this removal of the explicit presence of the component index "i" helps readers focus on the true physical meaning of the vector. Actual calculations involving that vector will, of course, require reversion back to index notation.

The concept of smooth functions having only a *single*-argument (i.e., members of C^1_∞) can be extended to consider *binary* (two argument) smooth functions. We denote this new class of function by C^2_∞, where the superscript "2" indicates that members of this class of "objects" have *two* arguments. Members of C^2_∞ will be written in the form $F(x, y)$, where the arguments, x and y, are permitted to take on real values from $-\infty$ to $+\infty$, and the symbols that we use for these arguments are inconsequential (i.e., $F(r, s)$ would denote the *same* function as $F(x, y)$). Analogously, we will introduce in this section the concept of an engineering *tensor*. We will denote this new class of "object" by V_3^2, where the superscript "2" indicates that we are discussing *second-order* tensors and the subscript 3 is again denoting the dimension of the underlying physical space. Members of V_3^2 will be written indicially in the form F_{ij}, where the indices, *i* and *j*, are permitted to take on integer values from 1 to 3, and the symbols that we use for them are inconsequential (i.e., F_{rs} would denote the *same* tensor as F_{ij}). Note the analogy: second-order engineering tensors are similar in spirit to functions of two variables.

A binary function $F(x, y)$ can be used to transform a *single-argument* function $v(y)$ to become a new function $u(x)$ through the special operation

$$u(x) = \int F(x, y)v(y)dy . \qquad (7.5)$$

where the integration extends over the permissible values of the second argument y (i.e., the integral goes from $y=-\infty$ to $y=+\infty$). The binary function $F(x, y)$ fully dominates and characterizes the nature of the transformation. Analogously, we will find that the dominant role of engineering tensors is to characterize a primitive operation that transforms a single engineering vector $\underset{\sim}{v}$ to become a new vector $\underset{\sim}{u}$. In particular, this operation is written in index form as

$$u_i = \sum F_{ij} v_j \qquad (7.6)$$

where the summation extends over the permissible values of the second index "*j*" (i.e., the sum goes from $j=1$ to $j=3$). Note the analogy between Eqs. (7.5) and (7.6); one integrates over the second dummy integration variable while the other sums over the second dummy summation index.

Recognizing where analogies do and don't exist between seemingly unrelated fields of study can be indispensable. Talented researchers are often able to use what they know about a one subject to help them learn a new subject. Careful researchers are diligent to spend as much time identifying where the analogies *break down* as where they *apply*.

Linear operators (transformations)

GOAL: Set stage for the "linear transformation" definition of a tensor by showing how a 3×3 matrix arises naturally to characterize all information needed to compute how a linear vector-to-vector operation transforms its independent vector argument. Emphasize that this "component" matrix contains numbers that are tied to the choice of basis. Set stage for both the dyadic definition of a tensor and the component transformation definition. Introduce the notion of basis dyads.

Let f denote a vector-valued function of a vector. The symbol f denotes a set of *instructions*. The operator is not *itself* a vector — its *output* is a vector. Let $\underset{\sim}{x}$ be the vector supplied as an argument to the function, and let $\underset{\sim}{y}$ be the result of applying the function. Then we would write, $\underset{\sim}{y} = f(\underset{\sim}{x})$.

The function is linear if and only if

$$f(\alpha_1 \underset{\sim}{x}_1 + \alpha_2 \underset{\sim}{x}_2) = \alpha_1 f(\underset{\sim}{x}_1) + \alpha_2 f(\underset{\sim}{x}_2)$$

Equivalently, the function $\underset{\sim}{f}$ is called "**linear**" if and only if

$$\begin{array}{llll} & (i) & f(\alpha \underset{\sim}{v}) = \alpha f(\underset{\sim}{v}) & \text{for all scalars } \alpha \\ \text{and} & (ii) & f(\underset{\sim}{u} + \underset{\sim}{v}) = f(\underset{\sim}{u}) + f(\underset{\sim}{v}) & \text{for all vectors } \underset{\sim}{u} \text{ and } \underset{\sim}{v} \end{array} \tag{7.7}$$

Linearity gives you freedom to apply the operator to a linear combination of vectors in either of two equivalent ways: (1) you can take the linear combination of the vectors and directly apply the operator f or (2) you can apply the operator separately to each of starting vectors and *then* take the linear combination of the individual results.

If we know how a linear function transforms the three *base* vectors, then linearity lets us quickly determine how the function transforms an arbitrary vector. To see why this is so, let's define three $\underset{\sim}{f}_i$ vectors to be the transformations of the orthonormal basis vectors:

$$\underset{\sim}{f}_1 \equiv f(\underset{\sim}{e}_1), \qquad \underset{\sim}{f}_2 \equiv f(\underset{\sim}{e}_2), \qquad \text{and} \qquad \underset{\sim}{f}_3 \equiv f(\underset{\sim}{e}_3) \tag{7.8}$$

Importantly, these three vectors can be computed once and saved for all time. Usually these vectors are stored as columns of a 3×3 matrix $[F]$ so that

$$[F] = [\{\underset{\sim}{f}_1\}\{\underset{\sim}{f}_2\}\{\underset{\sim}{f}_3\}] \tag{7.9}$$

In other words, F_{ij} is the i^{th} component of $\underset{\sim}{f}_j$. More precisely,

$$F_{ij} = \underset{\sim}{e}_i \bullet \underset{\sim}{f}_j \qquad \Leftrightarrow \qquad \underset{\sim}{f}_j = F_{ij} \underset{\sim}{e}_i \tag{7.10}$$

Recall that an arbitrary vector $\underset{\sim}{x}$ may be written

$$\underset{\sim}{x} = x_1 \underset{\sim}{e}_1 + x_2 \underset{\sim}{e}_2 + x_3 \underset{\sim}{e}_3 \tag{7.11}$$

Operating on this vector by the linear function f and applying the linearity property, shows that the action of the linear transformation on an arbitrary vector $\underset{\sim}{v}$ can be rapidly computed by

$$\begin{aligned} f(\underset{\sim}{x}) &= x_1 f(\underset{\sim}{e}_1) + x_2 f(\underset{\sim}{e}_2) + x_3 f(\underset{\sim}{e}_3) \\ &= x_1 \underset{\sim}{f}_1 + x_2 \underset{\sim}{f}_2 + x_3 \underset{\sim}{f}_3 \end{aligned} \tag{7.12}$$

Using the summation convention,

$$f(\underset{\sim}{\boldsymbol{x}}) = x_j\underset{\sim}{\boldsymbol{f}}_j \tag{7.13}$$

Substituting Eq. (7.10) into Eq (7.13) gives

$$f(\underset{\sim}{\boldsymbol{x}}) = x_j(F_{ij}\underset{\sim}{\boldsymbol{e}}_i) = (F_{ij}x_j)\underset{\sim}{\boldsymbol{e}}_i \tag{7.14}$$

If we let the vector $\underset{\sim}{\boldsymbol{y}}$ denote the result of $f(\underset{\sim}{\boldsymbol{x}})$, then we write

$$\underset{\sim}{\boldsymbol{y}} = f(\underset{\sim}{\boldsymbol{x}}) = y_i\underset{\sim}{\boldsymbol{e}}_i\,. \tag{7.15}$$

Thus, the i^{th} component of $f(\underset{\sim}{\boldsymbol{x}})$ is obtained by the simple matrix multiplication, $y_i = F_{ij}x_j$, or

$$\{y\} = [F]\{x\} \tag{7.16}$$

The matrix $[F]$ is called the matrix of components of the “tensor” associated with the linear transformation f. These components are relative to the basis $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ that we used to compute the $\underset{\sim}{\boldsymbol{f}}_k$ vectors. Using a different basis will result in a different $[F]$ component matrix.

For any linear function, there exists an associated tensor, and vice versa.* Consequently, many authors [e.g., 20] define a tensor to be any linear transformation of vectors to vectors.† We prefer to keep the two ideas separate in our presentation. After all, we frequently know the linear transformation (i.e., the set of instructions) *before* we know the tensor itself.‡ For example, when $\underset{\sim}{\boldsymbol{b}}$ is a known constant vector, the cross product operation $f(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{b}} \times \underset{\sim}{\boldsymbol{x}}$ is linear with respect to $\underset{\sim}{\boldsymbol{x}}$. However, the *tensor* associated with this operation is not immediately obvious. The operation is linear, so we know that a tensor (representable by a matrix) must exist, but what is the tensor component matrix that corresponds to $f(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{b}} \times \underset{\sim}{\boldsymbol{x}}$? Not obvious.

It is critical to recognize that the components of the matrix F_{ij} are, by construction, referenced your chosen basis. Thus, implicitly, a tensor must consist of *both* a matrix of components *and* an associated basis. If the basis changes, then the component matrix changes in a specific way (explained later), which is why some people [e.g., 19] define a tensor to be a set of components that transform in the necessary way upon a change of basis.

The intimate dependence of the F_{ij} components on the underlying basis $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ is well emphasized by using the following basis notation for tensors:

* This claim is called the *Representation Theorem*, and it will be discussed in more detail on page 122. For now, take this statement as an unproved truth.

† As clarified by Simmonds, “...To say that we are given a 2nd order tensor $\boldsymbol{T}$ means that we are told $\boldsymbol{T}$’s action on (i.e., where it sends) any vector $\boldsymbol{v}$. Thus two 2nd order tensors $\boldsymbol{S}$ and $\boldsymbol{T}$ are said to be equal if their action on all vectors is the same...”

‡ The distinction is analogous to the unique correspondence between animals and their DNA sequences. Disregarding cloning and identical twins, there is exactly one animal for each DNA sequence and vice versa, but this does not mean that animals and DNA sequences are the same. Likewise, in tensor analysis, one often has a well-characterized linear transformation *without* having an explicit expression for the associated tensor (even though we know it exists).

$$\begin{aligned} \underset{\approx}{\boldsymbol{F}} = \quad & F_{11}(\underset{\sim}{\boldsymbol{e}}_1 \otimes \underset{\sim}{\boldsymbol{e}}_1) + F_{12}(\underset{\sim}{\boldsymbol{e}}_1 \otimes \underset{\sim}{\boldsymbol{e}}_2) + F_{13}(\underset{\sim}{\boldsymbol{e}}_1 \otimes \underset{\sim}{\boldsymbol{e}}_3) \\ + \quad & F_{21}(\underset{\sim}{\boldsymbol{e}}_2 \otimes \underset{\sim}{\boldsymbol{e}}_1) + F_{22}(\underset{\sim}{\boldsymbol{e}}_2 \otimes \underset{\sim}{\boldsymbol{e}}_2) + F_{23}(\underset{\sim}{\boldsymbol{e}}_2 \otimes \underset{\sim}{\boldsymbol{e}}_3) \\ + \quad & F_{31}(\underset{\sim}{\boldsymbol{e}}_3 \otimes \underset{\sim}{\boldsymbol{e}}_1) + F_{32}(\underset{\sim}{\boldsymbol{e}}_3 \otimes \underset{\sim}{\boldsymbol{e}}_2) + F_{33}(\underset{\sim}{\boldsymbol{e}}_3 \otimes \underset{\sim}{\boldsymbol{e}}_3) \end{aligned} \tag{7.17}$$

The dyadic multiplication “⊗” symbol is defined below. For now, the presence of $\underset{\sim}{\boldsymbol{e}}_i \otimes \underset{\sim}{\boldsymbol{e}}_j$ next to F_{ij} should be regarded as a way to indicate that the F_{ij} matrix of components is defined relative to the $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ basis. In this form, $\underset{\approx}{\boldsymbol{F}}$ is often called a “**dyadic**,” but we will simply continue to use the term “tensor.”

The above expansion may be written compactly using the summation convention as

$$\underset{\approx}{\boldsymbol{F}} = F_{ij}(\underset{\sim}{\boldsymbol{e}}_i \otimes \underset{\sim}{\boldsymbol{e}}_j) \tag{7.18}$$

We will usually omit the “⊗” symbol for dyadic multiplication so that two vectors written side by side are *understood* to be multiplied dyadically, and the above equations would be written more compactly as

$$\begin{aligned} \underset{\approx}{\boldsymbol{F}} = \quad & F_{11}\underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1 + F_{12}\underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_2 + F_{13}\underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_3 \\ + \quad & F_{21}\underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_1 + F_{22}\underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2 + F_{23}\underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_3 \\ + \quad & F_{31}\underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_1 + F_{32}\underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_2 + F_{33}\underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_3 \end{aligned} \tag{7.19}$$

and

$$\underset{\approx}{\boldsymbol{F}} = F_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \tag{7.20}$$

If $\underset{\approx}{\boldsymbol{F}}$ is the tensor associated with a vector-valued linear transformation of vectors, $f(\underset{\sim}{\boldsymbol{x}})$, then we introduce the following notation:

$$\boxed{\underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}} \quad \text{means the same thing as} \quad f(\underset{\sim}{\boldsymbol{x}})} \tag{7.21}$$

Most authors do not use our double underline (under-tilde) convention to indicate the order of a tensor, so you will typically see tensors typeset simply in bold, ***F***. Furthermore, many authors do not use the raised dot notation (hence, to them, $\boldsymbol{Fx}$ means the same thing as what we write as $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}}$. As will be further explained later, our notation lends itself better to heuristic “self-explanatory” interpretations of the intended meanings of operations, which becomes indispensable when working with higher-order tensors in modern material modeling applications.

Recall that many people define a tensor to be a linear operation taking vectors to vectors. The basis dyad $\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j$ is itself a tensor, and it has a component matrix that contains all zeros except for a 1 at the ij position. Thus, the matrix form of Eq. (7.19) is simply

$$\begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} = F_{11}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{12}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{13}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
$$+ F_{21}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{22}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{23}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$
$$+ F_{31}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + F_{32}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + F_{33}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.22}$$

This basis/matrix correspondence is completely analogous to the act of writing a vector in basis form as

$$\underset{\sim}{\boldsymbol{v}} = v_1\underset{\sim}{\boldsymbol{e}}_1 + v_2\underset{\sim}{\boldsymbol{e}}_2 + v_3\underset{\sim}{\boldsymbol{e}}_3 \tag{7.23}$$

and then interpreting this expression using matrix notation as

$$\begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} = v_1\begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} + v_2\begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} + v_3\begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \tag{7.24}$$

Dyads and dyadic multiplication

GOAL: Define, cite properties

The "⊗" symbol represents "dyadic multiplication," also frequently referred to as a "dyadic product" [24]. A dyad [35] between two vectors $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{b}}$ is an abstract mathematical construction denoted $\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}$ that takes on meaning when it operates on an arbitrary vector $\underset{\sim}{\boldsymbol{v}}$ as follows:

$$(\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}) \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}}) \quad \text{for all vectors } \underset{\sim}{\boldsymbol{v}} \tag{7.25}$$

A dyad has physical meaning only when operating on a vector. Alone, it is a mathematical bookkeeping device that can be independently manipulated but has no inherent meaning in its own right.

Two dyads $\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}$ and $\underset{\sim}{\boldsymbol{r}} \otimes \underset{\sim}{\boldsymbol{s}}$ are said to be "equal" if and only if

$$(\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}) \bullet \underset{\sim}{\boldsymbol{v}} = (\underset{\sim}{\boldsymbol{r}} \otimes \underset{\sim}{\boldsymbol{s}}) \bullet \underset{\sim}{\boldsymbol{v}} \quad \text{for all vectors } \underset{\sim}{\boldsymbol{v}} \tag{7.26}$$

Note that the right hand side of Eq. (7.25) equals the vector $\underset{\sim}{\boldsymbol{a}}$ times a scalar, $(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}})$. Thus, $(\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}) \bullet \underset{\sim}{\boldsymbol{v}}$ is proportional to $\underset{\sim}{\boldsymbol{a}}$. Similarly, $(\underset{\sim}{\boldsymbol{b}} \otimes \underset{\sim}{\boldsymbol{a}}) \bullet \underset{\sim}{\boldsymbol{v}}$ would be proportional to $\underset{\sim}{\boldsymbol{b}}$. Thus, dyadic multiplication does not commute. That is,

$$\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}} \neq \underset{\sim}{\boldsymbol{b}} \otimes \underset{\sim}{\boldsymbol{a}} \quad \text{in general} \tag{7.27}$$

Applying the definition of a dyad (namely, Eq. (7.25)) to both sides of Eq. (7.26), note that two dyads $\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}$ and $\underset{\sim}{\boldsymbol{r}} \otimes \underset{\sim}{\boldsymbol{s}}$ are equal if and only if

$$\underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}}) = \underset{\sim}{\boldsymbol{r}}(\underset{\sim}{\boldsymbol{s}} \bullet \underset{\sim}{\boldsymbol{v}}) \quad \text{for all vectors } \underset{\sim}{\boldsymbol{v}} \tag{7.28}$$

It is straightforward (but somewhat tedious) to show that, *for nonzero vectors*,

$$\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}} = \underset{\sim}{\boldsymbol{r}} \otimes \underset{\sim}{\boldsymbol{s}} \text{ only if } \underset{\sim}{\boldsymbol{r}} = \alpha\underset{\sim}{\boldsymbol{a}} \text{ and } \underset{\sim}{\boldsymbol{s}} = \beta\underset{\sim}{\boldsymbol{b}} \text{ where } \alpha\beta = 1\,. \tag{7.29}$$

In other words, the vector $\underset{\sim}{\boldsymbol{r}}$ must be proportional to $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{s}}$ must be proportional to $\underset{\sim}{\boldsymbol{b}}$. The condition on the proportionality constant is equivalent to requiring that $\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{b}} = \underset{\sim}{\boldsymbol{r}} \bullet \underset{\sim}{\boldsymbol{s}}$. If $\underset{\sim}{\boldsymbol{a}}$ and/or $\underset{\sim}{\boldsymbol{b}}$ is zero, then $\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}} = \underset{\sim}{\boldsymbol{r}} \otimes \underset{\sim}{\boldsymbol{s}}$ only if $\underset{\sim}{\boldsymbol{r}}$ and/or $\underset{\sim}{\boldsymbol{s}}$ is zero.

The above discussion reiterates that dyadic multiplication does not commute; i.e.,

$$\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}} \neq \underset{\sim}{\boldsymbol{b}} \otimes \underset{\sim}{\boldsymbol{a}} \tag{7.30}$$

Referring to Eq. (7.29), we obtain the necessary and sufficient condition for $\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}$ to commute:

$$\underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}} = \underset{\sim}{\boldsymbol{b}} \otimes \underset{\sim}{\boldsymbol{a}} \text{ if and only if } \underset{\sim}{\boldsymbol{b}} = \alpha\underset{\sim}{\boldsymbol{a}} \text{ for some } \alpha\,. \tag{7.31}$$

In other words, the two vectors in the dyad would have to be multiples of each other in order for a dyadic multiplication to commute.

Simpler "no-symbol" dyadic notation

GOAL: Advocate in favor of not using the dyadic multiplication symbol.

Whenever two vectors are written side-by-side with no "⊗" symbol between them, they are understood to be multiplied dyadically. Thus,

$$\boxed{\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}} \text{ means the same as } \underset{\sim}{\boldsymbol{a}} \otimes \underset{\sim}{\boldsymbol{b}}\,.} \tag{7.32}$$

In addition to being a more compact notation, this convention for dyadic multiplication also has the appealing property that the definition of Eq. (7.25) can be written

$$(\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}) \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}})\,, \tag{7.33}$$

which allows us to drop the parentheses without ambiguity.* We advocate retaining the explicit "⊗" symbol only when it is needed for clarity.

Dyads are like defensive linemen in football — essential, but rarely of much interest *per se*. Dyads are merely abstract mathematical objects, which (up to this point) have no apparent practical use. After we define addition and scalar multiplication for dyads, we will see that *tensors* (which are of great practical importance) are always expressible as a linear combination of the nine possible dyads between the orthonormal base vectors. These nine basis dyads, $\{\underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_2, \ldots, \underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_3\}$, form a basis for tensors just as the three base vectors $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ form a basis for ordinary vectors. With the "no symbol" notation, Eq. (7.18) may be written in basis notation as

$$\underset{\approx}{\boldsymbol{F}} = F_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\,, \tag{7.34}$$

In this expression, the indices i and j are summed from 1 to 3. Thus, the above expression is a linear combination of dyads, the meaning of which is described in more detail below.

The matrix associated with a dyad

GOAL: Show that a dyad has an associated 3×3 matrix that is equivalent to the outer product of the vectors. Set stage for making connection between dyads and tensors — both have associated matrices. Dyads are special kinds of tensors.

The right-hand side of Eq. (7.33) is a vector whose i^{th} component is

$$a_i(b_j v_j) \tag{7.35}$$

We can define a 3×3 matrix whose ij components are $a_i b_j$. Then the expression in Eq. (7.35) can be written

$$[a_i b_j] v_j \tag{7.36}$$

* Some people strenuously object to this side-by-side notation for dyadic multiplication. They argue in favor of the "⊗" symbol because dyadic multiplication is "different" from scalar multiplication. By the same logic, however, the identity $(\gamma + \beta)\underset{\sim}{\boldsymbol{v}} = \gamma\underset{\sim}{\boldsymbol{v}} + \beta\underset{\sim}{\boldsymbol{v}}$ should also be notationally objectionable because addition between vectors is "different" from addition between scalars. Likewise, the notation dy/dx would be objectionable because derivatives are not really fractions. In both mathematics and engineering, we routinely overload operations for good reason: *heuristic notational advantages*. The meanings of the overloaded operations are implied by the nature of the arguments.

This can be written as a matrix equation

$$\begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix}, \tag{7.37}$$

which is the matrix representation of the left-hand side of Eq. (7.33). Using conventional matrix notation, note that

$$[\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}] = \begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}^T = \{a\}\langle b\rangle \tag{7.38}$$

The result is a square matrix. You should contrast this operation with the similar-looking *scalar-valued* operation of Eq. (5.3) in which the transpose merely occurs in a different place!

The operation in Eq. (7.33) is a *linear* transformation of the vector $\underset{\sim}{\boldsymbol{v}}$ resulting in a new vector, $\underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}})$. Consequently, as discussed on page 99, this operation has an associated 3×3 matrix, which it turns out to be identical to the matrix in Eq. (7.38). For example, Eq. (7.9) showed that the first column of the matrix associated with a linear transformation $f(\underset{\sim}{\boldsymbol{x}})$ is given by $f(\underset{\sim}{\boldsymbol{e}}_1)$. In the present context, our linear function is $f(\underset{\sim}{\boldsymbol{v}}) = \underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}})$, so the first column of the associated matrix should contain $\underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{e}}_1)$, and (recalling Eq. 5.13), we see that this is indeed the first column of the matrix in Eq. (7.38).

The sum of dyads

GOAL: define, cite properties

The sum of two dyads, $\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}$ and $\underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}}$, is an abstract notion defined such that

$$(\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}} + \underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}}) \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}}) + \underset{\sim}{\boldsymbol{c}}(\underset{\sim}{\boldsymbol{d}} \bullet \underset{\sim}{\boldsymbol{v}}) \quad \text{for any vector } \underset{\sim}{\boldsymbol{v}} \tag{7.39}$$

The sum of three dyads, $\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}} + \underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}} + \underset{\sim}{\boldsymbol{e}}\underset{\sim}{\boldsymbol{f}}$ is defined such that

$$(\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}} + \underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}} + \underset{\sim}{\boldsymbol{e}}\underset{\sim}{\boldsymbol{f}}) \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}}) + \underset{\sim}{\boldsymbol{c}}(\underset{\sim}{\boldsymbol{d}} \bullet \underset{\sim}{\boldsymbol{v}}) + \underset{\sim}{\boldsymbol{e}}(\underset{\sim}{\boldsymbol{f}} \bullet \underset{\sim}{\boldsymbol{v}}) \quad \text{for all } \underset{\sim}{\boldsymbol{v}} \tag{7.40}$$

The sum of two or more dyads takes on meaning only when operating on a vector. The sum of dyads also has a matrix interpretation. Specifically, the matrix $[\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}} + \underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}}]$ associated with the sum of two dyads is obtained by simply adding the matrices for $[\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}]$ and $[\underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}}]$.

A sum of two or three dyads is *NOT* (generally) reducible

GOAL: Show that the sum of two or three dyads cannot always be rearranged to become just a single dyad — the sum of dyads is itself a new object. It is a superset (i.e., the set of all dyads is a subset of the set of all sums of two dyads, but not vice versa).

By applying the definition of Eq. (7.39) it's straightforward to prove that

$$\underset{\sim}{a}\underset{\sim}{p} + \underset{\sim}{a}\underset{\sim}{q} = \underset{\sim}{a}(\underset{\sim}{p} + \underset{\sim}{q}) \tag{7.41}$$

This is a *very special* situation in which the sum of two dyads can be written as a single dyad (between the vector $\underset{\sim}{a}$ and the vector $\underset{\sim}{p} + \underset{\sim}{q}$).

In general, the sum of two dyads *cannot* be expressed as a single dyad. One *cannot* generally find two vectors $\underset{\sim}{r}$ and $\underset{\sim}{s}$ such that $\underset{\sim}{a}\underset{\sim}{b} + \underset{\sim}{c}\underset{\sim}{d} = \underset{\sim}{r}\underset{\sim}{s}$. Thus, the sum of two dyads is itself a distinct new abstract object. The sum of three dyads is also a distinct mathematical abstraction that in general cannot be written as either a single dyad or the sum of two dyads. We will later demonstrate that the sum of four or more dyads can always be reduced to the sum of three or fewer dyads (provided that the vectors belong to ordinary 3D space). Thus, the sum of four or more dyads is *not* a new object.

The set of all "objects" that are expressible as a sum of two dyads is a "superset" of the set of all dyads because any *single* dyad can be written in the form $\underset{\sim}{a}\underset{\sim}{b} + \underset{\sim}{0}\underset{\sim}{0}$. Likewise, objects expressible as the sum of *three* dyads is a superset of objects expressible as the sum of two dyads.

Scalar multiplication of a dyad

GOAL: Define this operation, cite properties, emphasize that scalar multiplication can act on any of the individual vectors forming a dyad.

The multiplication of a scalar γ with the dyad $\underset{\sim}{a}\underset{\sim}{b}$ is a new dyad denoted $\gamma(\underset{\sim}{a}\underset{\sim}{b})$, and is naturally defined such that

$$[\gamma(\underset{\sim}{a}\underset{\sim}{b})] \bullet \underset{\sim}{v} = \gamma\underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{v}) \qquad \text{for all } \underset{\sim}{v} \tag{7.42}$$

Using the dyad symbol for clarity, we note that the scalar multiple may be regarded as an external multiplier or it may be absorbed into any one of the vectors in the dyad:

$$\gamma(\underset{\sim}{a} \otimes \underset{\sim}{b}) = (\gamma\underset{\sim}{a}) \otimes \underset{\sim}{b} = (\underset{\sim}{a}\gamma) \otimes \underset{\sim}{b} = \underset{\sim}{a} \otimes (\gamma\underset{\sim}{b}) = \underset{\sim}{a} \otimes (\underset{\sim}{b}\gamma) \tag{7.43}$$

No matter where the scalar is placed (on the right side of the dyad or between the two vectors), it can always be moved to the left side. In other words,

$$\underset{\sim}{a}\gamma\underset{\sim}{b} \text{ and } \underset{\sim}{a}\underset{\sim}{b}\gamma \text{ mean the same thing as } \gamma(\underset{\sim}{a}\underset{\sim}{b}). \tag{7.44}$$

Thus, scalar multiplication requires no parentheses.

Scalar multiplication of a *sum* of dyads is defined as you might expect. Namely,

$$\begin{aligned} \gamma(\underset{\sim}{a}\underset{\sim}{b} + \underset{\sim}{c}\underset{\sim}{d}) &= (\gamma\underset{\sim}{a}\underset{\sim}{b} + \gamma\underset{\sim}{c}\underset{\sim}{d}) \\ \gamma(\underset{\sim}{a}\underset{\sim}{b} + \underset{\sim}{c}\underset{\sim}{d} + \underset{\sim}{e}\underset{\sim}{f}) &= (\gamma\underset{\sim}{a}\underset{\sim}{b} + \gamma\underset{\sim}{c}\underset{\sim}{d} + \gamma\underset{\sim}{e}\underset{\sim}{f}) \end{aligned} \tag{7.45}$$

The sum of four or more dyads *is* reducible! (not a superset)

GOAL: Show that the sum of more than three dyads (in 3D) can always be reduced to the sum of three or fewer dyads.

The sum of four or more dyads is defined by natural extension of the definitions of Eqs. (7.41) and (7.40). Thus, for example, the sum of four dyads is defined so that, for all vectors $\underset{\sim}{\boldsymbol{v}}$,

$$(\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}} + \underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}} + \underset{\sim}{\boldsymbol{e}}\underset{\sim}{\boldsymbol{f}} + \underset{\sim}{\boldsymbol{g}}\underset{\sim}{\boldsymbol{h}}) \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}}) + \underset{\sim}{\boldsymbol{c}}(\underset{\sim}{\boldsymbol{d}} \bullet \underset{\sim}{\boldsymbol{v}}) + \underset{\sim}{\boldsymbol{e}}(\underset{\sim}{\boldsymbol{f}} \bullet \underset{\sim}{\boldsymbol{v}}) + \underset{\sim}{\boldsymbol{g}}(\underset{\sim}{\boldsymbol{h}} \bullet \underset{\sim}{\boldsymbol{v}}) \tag{7.46}$$

In typical engineering applications, vectors are ordinary vectors in 3D physical space. Consequently, any set of four vectors must be linearly dependent. Thus, at least one of the vectors in the set, $\{\underset{\sim}{\boldsymbol{b}}, \underset{\sim}{\boldsymbol{d}}, \underset{\sim}{\boldsymbol{f}}, \underset{\sim}{\boldsymbol{h}}\}$, can be written as a linear combination of the other vectors. Suppose, for illustration, that the dependent vector is $\underset{\sim}{\boldsymbol{h}}$. Then there exist α_k scalars such that

$$\underset{\sim}{\boldsymbol{h}} = \alpha_1\underset{\sim}{\boldsymbol{b}} + \alpha_2\underset{\sim}{\boldsymbol{d}} + \alpha_3\underset{\sim}{\boldsymbol{f}} \tag{7.47}$$

Hence, the dyad $\underset{\sim}{\boldsymbol{g}}\underset{\sim}{\boldsymbol{h}}$ can be written

$$\underset{\sim}{\boldsymbol{g}}\underset{\sim}{\boldsymbol{h}} = (\alpha_1\underset{\sim}{\boldsymbol{g}})\underset{\sim}{\boldsymbol{b}} + (\alpha_2\underset{\sim}{\boldsymbol{g}})\underset{\sim}{\boldsymbol{d}} + (\alpha_3\underset{\sim}{\boldsymbol{g}})\underset{\sim}{\boldsymbol{f}} \tag{7.48}$$

and therefore,

$$\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}} + \underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{d}} + \underset{\sim}{\boldsymbol{e}}\underset{\sim}{\boldsymbol{f}} + \underset{\sim}{\boldsymbol{g}}\underset{\sim}{\boldsymbol{h}} = (\underset{\sim}{\boldsymbol{a}} + \alpha_1\underset{\sim}{\boldsymbol{g}})\underset{\sim}{\boldsymbol{b}} + (\underset{\sim}{\boldsymbol{c}} + \alpha_2\underset{\sim}{\boldsymbol{g}})\underset{\sim}{\boldsymbol{d}} + (\underset{\sim}{\boldsymbol{e}} + \alpha_3\underset{\sim}{\boldsymbol{g}})\underset{\sim}{\boldsymbol{f}} \tag{7.49}$$

The left-hand side is the sum of four dyads. The right hand side is the sum of three dyads. This proves that any sum of four or more dyads can always be reduced to three or fewer dyads. Consequently, the sum of four or more dyads is not a new abstract object! The sum of any number of dyads is generally referred to as a **dyadic**, but we will use the term “**tensor**.”

The dyad definition of a second-order tensor

GOAL: Use result of previous section to define a “tensor” to be any dyad or sum of dyads.

The function $f[\underset{\sim}{\boldsymbol{v}}] = \underset{\sim}{\boldsymbol{a}}(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{v}})$ is linear in $\underset{\sim}{\boldsymbol{v}}$. The dyad $\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}$ is the tensor associated with this linear transformation. Dyads are the most rudimentary tensors. As emphasized earlier, the sum of two dyads generally cannot be reduced to a single dyad. Likewise, the sum of three dyads cannot be reduced to the sum of fewer dyads. However, the sum of four or more dyads can always be reduced to the sum three or fewer dyads. Consequently, a second-order tensor may be defined as any sum of dyads. The term “**dyadic**” is also used to mean the same thing. This definition is equivalent to the more traditional definition of a tensor in terms of linear transformations from vectors to vectors in the sense that one definition implies the other. The “sum of dyads” definition is more useful in certain settings, especially when considering a tensor as a higher-dimensional vector.

For clarity, we typeset second-order tensors in bold using two under-tildes. Thus, $\underset{\approx}{\mathbf{A}}$, $\underset{\approx}{\mathbf{B}}$, $\underset{\approx}{\mathbf{U}}$ would denote second-order tensors. Tensors are extremely useful in physical applications. For example, tensors represent stress, strain, rotational inertia, electrical permittivity, and many other important physical quantities. Once a physical quantity is proved to be representable by a tensor, a wealth of theorems from abstract tensor analysis then apply to the tensor, furthering its physical interpretation.

Expansion of a second-order tensor in terms of basis dyads

GOAL: show that expanding an ordinary engineering tensor as a linear combination of the NINE possible ways to form basis dyads is similar in spirit to expanding an ordinary engineering vector as a linear combination of the laboratory orthonormal basis.

A dyad is the most rudimentary form of a tensor. Many people think of a tensor as simply a 3×3 matrix. Most of the time, that view is adequate. However, *the components of a tensor's matrix are always referenced to a particular basis. If the basis changes, then the matrix changes.* As discussed later, the matrix corresponds to a tensor if and only if the matrix changes in a particular manner upon a change of basis. Since the preceding statement contained the phrase "if and only if" the component transformation property is sometimes taken as a definition of the term tensor. For now, however, we shall continue with our equivalent view of a tensor as a sum of dyads.

For a three-dimensional vector space (i.e., for a space V_3^1 where vectors have three components), the sum of *two* dyads *cannot* generally be written as a single dyad. Likewise, the sum of three dyads generally cannot be reduced to the sum of fewer dyads. However, as proved earlier, any sum of more than three dyads can always be reduced to the sum of three or fewer dyads. From this observation, we offered the very abstract definition of a "second-order tensor" to be any sum of dyads. Knowing that such a sum can always be written as the sum of at most three dyads, one might be tempted to *wrongly* conclude that tensors form a three dimensional space; however the three dyads are not the same for all tensors — in fact, they are not even unique for any given tensor. To determine the dimension of the space of tensors, we need to determine how many scalars are needed to uniquely define a tensor.

Given a tensor $\underset{\approx}{\mathbf{A}}$ (*i.e.,* given a sum of dyads), we know that there exist[*] vectors such that

$$\underset{\approx}{\mathbf{A}} = \underset{\sim}{\mathbf{a}}\underset{\sim}{\mathbf{b}} + \underset{\sim}{\mathbf{c}}\underset{\sim}{\mathbf{d}} + \underset{\sim}{\mathbf{e}}\underset{\sim}{\mathbf{f}} \qquad (7.50)$$

Each of the vectors may be expanded in terms of a basis ($\underset{\sim}{\mathbf{a}} = a_i \underset{\sim}{\mathbf{e}}_i$, etc.), in which case, the first term would become $\underset{\sim}{\mathbf{a}}\underset{\sim}{\mathbf{b}} = a_i \underset{\sim}{\mathbf{e}}_i b_j \underset{\sim}{\mathbf{e}}_j$. Using the property of Eq. (7.44), this can be written

$$\underset{\sim}{\mathbf{a}}\underset{\sim}{\mathbf{b}} = a_i b_j \underset{\sim}{\mathbf{e}}_i \underset{\sim}{\mathbf{e}}_j \qquad (7.51)$$

[*] Keep in mind, we only need to assert that these vectors exist. In practice, the appropriate vectors are almost never actually computed. Furthermore, they aren't even unique.

Performing similar expansions for the other terms in Eq. (7.50), we see that any sum of dyads (i.e. any tensor) can be written as a linear combination of the nine possible basis dyads $\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j$ for *i* and *j* ranging from 1 to 3. That is, for any tensor $\underset{\approx}{\mathbf{A}}$, there exist scalar coefficients (called components) A_{ij} such that

$$\underset{\approx}{\mathbf{A}} = A_{ij}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j \tag{7.52}$$

where there are an implied sums of *i* and *j* ranging from 1 to 3 for a total of nine terms. The principal advantage of the representation in Eq. (7.52) is that the A_{ij} components are unique for a given orthonormal basis, just as the components of an ordinary vector are unique for a given basis.

The A_{ij} components of tensors are often displayed in 3×3 arrays like this:

$$[\underset{\approx}{\mathbf{A}}] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{7.53}$$

Let $\underset{\approx}{\mathbf{A}}$ be a second-order tensor. In terms of an orthonormal basis $\{\underset{\sim}{\mathbf{e}}_1, \underset{\sim}{\mathbf{e}}_2, \underset{\sim}{\mathbf{e}}_3\}$, the tensor $\underset{\approx}{\mathbf{A}}$ may be written

$$\underset{\approx}{\mathbf{A}} = \sum_{i=1}^{3}\sum_{j=1}^{3} A_{ij}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j \tag{7.54}$$

or, using Einstein's summation convention,

$$\underset{\approx}{\mathbf{A}} = A_{ij}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j \tag{7.55}$$

It's crucial to realize that these components are referenced to a particular basis. When working with more than one basis, the component matrix might be subscripted with $\underset{\sim}{\mathbf{e}}\underset{\sim}{\mathbf{e}}$, as in the expression

$$[\underset{\approx}{\mathbf{A}}] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}_{\underset{\sim}{e}\underset{\sim}{e}} . \tag{7.56}$$

The $\underset{\sim}{\mathbf{e}}\underset{\sim}{\mathbf{e}}$ subscript tells you that the ij component of the matrix is the coefficient of $\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j$ in the basis expansion, $\underset{\approx}{\mathbf{A}} = A_{ij}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j$. Another way to indicate the associated basis is to append the name of the basis in angled brackets after the listing of the matrix. Thus, for example,

$$[\underset{\approx}{\mathbf{A}}] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad \text{<lab>} \tag{7.57}$$

would indicate to the reader that the components are referenced to the laboratory basis. Finally, the notation

$$[\underset{\approx}{A}] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \qquad <\underset{\sim}{e}_3=\underset{\sim}{n}> \tag{7.58}$$

would tell the reader that the component matrix $[\underset{\approx}{A}]$ is being displayed with respect to a special purpose basis in which the 3-direction has been aligned with a known unit vector $\underset{\sim}{n}$. This sort of display would be used when the component matrix is particularly simple with respect some particular (cleverly selected) basis. See, for example, our discussion of transverse isotropy on page 224.

Triads and higher-order tensors

GOAL: Introduce higher-order tensors

Recall that we introduced the concept of second-order tensors by discussing linear transformations from vectors to vectors (V_3^1 to V_3^1). From there, we introduced the primitive "new object" called a dyad, and defined a tensor to be any sum of dyads. That definition was really for *second-order* tensors. Any second-order tensor can be expressed as a linear combination of the nine basis dyads. This idea can be extended by introducing a **triad** to be a new object formed from three vectors $\underset{\sim}{a}\underset{\sim}{b}\underset{\sim}{c}$, defined such that

$$(\underset{\sim}{a}\underset{\sim}{b}\underset{\sim}{c}) \bullet \underset{\sim}{v} \equiv \underset{\sim}{a}\underset{\sim}{b}(\underset{\sim}{c} \bullet \underset{\sim}{v}) \text{ for all vectors } \underset{\sim}{v} \tag{7.59}$$

Addition and scalar multiplication of triads is defined similarly to that of dyads. Also, as was the case for dyads, you can show that scalar multiplication is vector-wise commutative, which means that multiplying a triad by a scalar will give the same result as multiplying any of its constituent vectors by that scalar:

$$\gamma(\underset{\sim}{a}\underset{\sim}{b}\underset{\sim}{c}) = (\gamma\underset{\sim}{a})\underset{\sim}{b}\underset{\sim}{c} = \underset{\sim}{a}(\gamma\underset{\sim}{b})\underset{\sim}{c} = \underset{\sim}{a}\underset{\sim}{b}(\gamma\underset{\sim}{c}) = (\underset{\sim}{a}\underset{\sim}{b}\underset{\sim}{c})\gamma \tag{7.60}$$

As was done for second-order tensors, we can recognize that there are *twenty-seven* ($3 \times 3 \times 3$) ways to form triads out of various combinations of the lab base vectors. A **third-order** tensor is defined to be any linear combination of the basis *triads*, and it can be characterized by a three-index $3 \times 3 \times 3$ matrix. We will indicate third-order tensors by using "under-tildes" and the basis expansion for a third-order tensor $\underset{\approx\!\!\sim}{H}$ can be written as

$$\underset{\approx\!\!\sim}{H} = H_{ijk}\underset{\sim}{e}_i\underset{\sim}{e}_j\underset{\sim}{e}_k \tag{7.61}$$

A third-order tensor characterizes the action of a linear operator that takes vectors to second-order tensors. If $\underset{\approx}{Y}$ is a second-order tensor given by a linear operation, $\underset{\approx}{Y} = H(\underset{\sim}{x})$, then, recalling the "advanced" discussion surrounding Eq. (9.17), there exists a third-order tensor $\underset{\approx\!\!\sim}{H}$ such that $Y_{ij} = H_{ijk}x_k$. Similarly, a third-order tensor characterizes a linear operator that takes second-order tensors to vectors. The alternating tensor (whose components with respect to a right-handed orthonormal basis equal the alternating symbol) is a good example. In rigid body mechanics, the alternating tensor can be used to convert the angular velocity *vector* into the angular velocity *tensor*. Conversely $1/2$ times the alternating tensor is the operator that will convert the angular velocity tensor back into the angular velocity vector.

These concepts extend to even higher-dimensional tensors. In materials modeling, *fourth-order* tensors play a dominant role because material models typically represent a set of rules for which a second-order tensor, such as the strain, is transformed to give a different second-order tensor, such as stress. Even if the transformation is non-linear, it will nevertheless still be linear in *rate* form, as will be discussed later. That's why it is so important to study linear operations. Virtually all nonlinear functions in real physical applications become linear in rate form, which therefore permits tapping into the rich literature available for solving linear problems.

Because a third-order tensor has three indices, each taking values from 1 to 3, a third order tensor has 27 components. A fourth-order tensor has 81 components.

Our V_m^n tensor "class" notation

GOAL: Define the meaning of V_m^n and show that any tensor of class V_m^n will have m^n components.

Recall that engineering scalars, vectors, second-order tensors, third-order tensors, and fourth-order tensors are specified by 1, 3, 9, 27, and 81 numbers, respectively. Scalars are often called 0^{th}-order tensors. Vectors are sometimes called 1^{st}-order tensors. In general, an n^{th} order engineering tensor has 3^n components, and we say that these tensors are of class V_3^n.

When solving a problem for which all tensors are have isotropic symmetry about some 2D plane embedded in 3D space, it is conventional to set up the basis so that the third base vector points perpendicular to that plane. Doing this permits the 3D problem to be reduced to a 2D problem where vectors now have only 2 nonzero components and second-order tensors are characterized by 2×2 matrices.

When working in two dimensions, an n^{th} order engineering tensor has 2^n components. Similarly, when working in an m-dimensional manifold (which is the higher dimensional version of a plane), an n^{th} order engineering tensor has m^n components, and we say that it is of class V_m^n.

It will be explained later that a second-order tensor of class V_m^2 is *also* a first order tensor of class $V_{m^2}^1$. For example, an ordinary second-order engineering tensor (class V_3^2) is also a first-order vector in a 9-dimensional space (class V_9^1). Just as you can speak of planes embedded in ordinary 3D space, you can also limit your attention to *subspaces* or *linear manifolds* with 9D tensor space. The set of all symmetric second-order engineering tensors, for example, is closed under tensor addition and scalar multiplication. By this we mean that any linear combination of symmetric tensors will be itself a symmetric tensor. Symmetric tensors (which have six independent components) can be regarded as a *six*-dimensional vector, and we say that they are of class V_6^1.

When you consider a problem in ordinary 3D space where most of the action occurs in a plane, then (unless that plane happens to be lined up with the laboratory basis) your analysis will undoubtedly simplify if you introduce a different orthonormal basis that is aligned with the plane. Similarly, when considering symmetric tensors embedded within 9D tensor space, your calculations will simplify if you switch away from the conventional $\underset{\sim}{e}_i\underset{\sim}{e}_j$ basis used in 9D space in favor of a different basis that is "aligned" with symmetric tensors. For example, instead of using $\underset{\sim}{e}_1\underset{\sim}{e}_2$ as one of your base-tensors, you would instead use $\underset{\sim}{e}_1\underset{\sim}{e}_2+\underset{\sim}{e}_2\underset{\sim}{e}_1$, which is symmetric. All of the other base tensors would need to be redefined as well if you want to switch to a basis that is aligned with symmetric tensors. To see how this change-of-basis would go, note that the component expansion for a general (not necessarily symmetric) tensor given in Eq. (7.22) can be written *equivalently* as

$$\begin{aligned}\begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} = {} & F_{11}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{22}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{33}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ & + \left(\frac{F_{23}+F_{32}}{2}\right)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + \left(\frac{F_{31}+F_{13}}{2}\right)\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \left(\frac{F_{12}+F_{21}}{2}\right)\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ & + \left(\frac{F_{32}-F_{23}}{2}\right)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + \left(\frac{F_{13}-F_{31}}{2}\right)\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} + \left(\frac{F_{21}-F_{12}}{2}\right)\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{7.62}$$

The matrices shown on the right-hand-side of this equation can be regarded as an alternative basis for 9D tensor space that is aligned with symmetric tensors. This basis is still capable of describing arbitrary non-symmetric tensors (just as a three-vector basis whose first two base vectors are contained within a plane in ordinary 3D space is still capable of describing all vectors, including those not in the plane). If $[F]$ happens to be symmetric, then $F_{23} = F_{32}$, $F_{31} = F_{13}$, and $F_{12} = F_{21}$, and the above expansion reduces to

$$\begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} = F_{11}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{22}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + F_{33}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + F_{23}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + F_{31}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + F_{12}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.63}$$

Thus, if you are dealing *exclusively* with symmetric tensors, then you only need six base tensors. The components with respect to these base tensors are called Voigt components. One disadvantage with the Voigt system is that the base tensors are not *unit* tensors, but that problem can be easily rectified by normalizing the Voigt basis to obtain what's called the Mandel basis. The key point here was that the set of all symmetric tensors really needs only *six* base tensors. Consequently, this set is of class V_6^1.

Fourth-order engineering tensors are of class V_3^4, but they are also of class V_9^2. In other words, a fourth-order tensor referenced to ordinary 3D space can be regarded as a *second*-order tensor referenced to an abstract 9D space. Consequently, they can be manipulated in computations by using a 9×9 matrix, with the indices ranging from 1 to 9 correspond to physical space index pairs 11, 22, 33, 23, 31, 12, 32, 13, 21. If you limit attention to fourth-order tensors that are minor symmetric, then the last three columns and last three rows of this 9×9 matrix will contain all zeros. In other words, you will be dealing only with the upper 6×6 part of the matrix. Consequently, minor-symmetric fourth-order tensors are of class V_6^2 and they have at most 6^2, or 36, nonzero components.

An ability to change how you regard the class of a tensor is useful in materials mechanics. For example, in plasticity, the trial elastic stress rate is found by assuming that a material is behaving elastically. If it is found that this assumption would move the stress into a "forbidden" region that violates the yield condition, then plastic flow must be occurring. The set of admissible *elastic* stresses is defined by a yield function such that $f(\underset{\approx}{\sigma}) < 0$. When $\underset{\approx}{\sigma}$ is regarded as a *vector* of class V_6^1, then $f(\underset{\approx}{\sigma}) = 0$ defines a yield *surface* in 6D space. For example, just as the equation $\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}} - R^2 = 0$ defines a sphere of radius R in ordinary 3D space, the equation $\underset{\approx}{\sigma} \boldsymbol{:} \underset{\approx}{\sigma} - R^2 = 0$ would define a *hypersphere* in 6D stress space. When the trial assumption of elastic behavior is found to move the stress into inadmissible stress states (i.e., those for which $f(\underset{\approx}{\sigma}) > 0$), then [8] the equations governing classical nonhardening plasticity can be used to show that the *actual* stress rate is obtained by projecting the trial elastic stress rate onto the yield surface. The projection operation that is similar in structure to the projection shown in Fig. 6.3 except that the vector dot product is replaced by the tensor inner product ($\boldsymbol{:}$). The outward "normal" $\underset{\approx}{B}$ that defines the target plane is the gradient of the yield function (i.e., $B_{ij} = \partial f / \partial \sigma_{ij}$. These sorts of statements are implicitly regarding stress (and the yield surface normal) as tensors of class V_6^1.

Comment

So far, we have covered two important definitions of the term "tensor." We will delay discussing the definition that is cast in terms of basis transformation rules until we can cover some basic tensor operations using the definitions we have given *so far*. This discussion will lead us into a discussion of how a tensor's matrix is related to the underlying basis, at which time, the basis transformation rules should make enough sense to understand why many people define tensors according to these rules.

> *“Seriousness is the favorite refuge of the shallow.”* — *Oscar Wilde*

8. Tensor operations

Dotting a tensor from the *right* by a vector

Keep in mind that a tensor is a mathematical bookkeeping device, which takes on physical meaning only when operating on a vector. The notation $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}}$ represents the linear transformation associated with the tensor. Specifically, applying the definition of a linear combination of dyads, the notation $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}}$ must be interpreted as

$$\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}} = (A_{ij}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j) \bullet \underset{\sim}{\mathbf{v}} = (A_{ij}\underset{\sim}{\mathbf{e}}_i)(\underset{\sim}{\mathbf{e}}_j \bullet \underset{\sim}{\mathbf{v}}) = (A_{ij}\underset{\sim}{\mathbf{e}}_i)v_j = (A_{ij}v_j)\underset{\sim}{\mathbf{e}}_i \tag{8.1}$$

In the second-to-last step, we have used Eq. (5.13) to write $\underset{\sim}{\mathbf{e}}_j \bullet \underset{\sim}{\mathbf{v}} = v_j$. Comparing the far left and far right-hand sides of Eq. (8.1) we conclude that

$$\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}} \text{ is a vector whose } i^{th} \text{ component is } A_{ij}v_j. \tag{8.2}$$

Thus, the components of $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}}$ may be obtained by the matrix multiplication $[A]\{v\}$.

The transpose of a tensor

The transpose of a tensor $\underset{\approx}{\mathbf{A}}$ is a new tensor denoted $\underset{\approx}{\mathbf{A}}^T$ and defined in direct notation such that

$$\underset{\sim}{\mathbf{u}} \bullet (\underset{\approx}{\mathbf{A}}^T \bullet \underset{\sim}{\mathbf{v}}) = \underset{\sim}{\mathbf{v}} \bullet (\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{u}}) \text{ for all vectors } \underset{\sim}{\mathbf{u}} \text{ and } \underset{\sim}{\mathbf{v}} \tag{8.3}$$

The ij component of $\underset{\approx}{\mathbf{A}}^T$ is denoted A^T_{ij}. In indicial notation, the above definition becomes

$$u_iA^T_{ij}v_j = v_mA_{mn}u_n \tag{8.4}$$

We would like to change the dummy summation indices on the right hand side so that we may compare it to the left hand side for arbitrary vectors $\underset{\sim}{\mathbf{u}}$ and $\underset{\sim}{\mathbf{v}}$. That means we want the index on components of $\underset{\sim}{\mathbf{u}}$ and $\underset{\sim}{\mathbf{v}}$ to be the same on both sides. Replacing n by i and replacing m by j, the above equation becomes

$$u_iA^T_{ij}v_j = v_jA_{ji}u_i \tag{8.5}$$

Rearranging gives

$$u_i(A_{ij}^T - A_{ji})v_j = 0 \tag{8.6}$$

The only way this can hold for all vectors $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$ is if

$$\boxed{A_{ij}^T = A_{ji}} \tag{8.7}$$

The above equation is often cited as the definition of the transpose, but the direct notation definition of Eq. (8.3) is more general since it makes no assumption that the underlying basis is Cartesian.

It is straightforward to show that

$$(\alpha\underset{\approx}{\boldsymbol{A}})^T = \alpha\underset{\approx}{\boldsymbol{A}}^T \tag{8.8}$$

and

$$(\underset{\approx}{\boldsymbol{A}} + \underset{\approx}{\boldsymbol{B}})^T = \underset{\approx}{\boldsymbol{A}}^T + \underset{\approx}{\boldsymbol{B}}^T \tag{8.9}$$

Thus, the transpose operation is linear. The transpose of a dyad $\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}$ is obtained by simply swapping the order of the vectors. Namely,

$$(\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}})^T = \underset{\sim}{\boldsymbol{b}}\underset{\sim}{\boldsymbol{a}} \tag{8.10}$$

If a tensor is written in the form of Eq. (7.55), the transpose simply swaps the base vectors. Specifically,

$$\underset{\approx}{\boldsymbol{A}}^T = (A_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j)^T = A_{ij}(\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j)^T = A_{ij}\underset{\sim}{\boldsymbol{e}}_j\underset{\sim}{\boldsymbol{e}}_i = A_{ji}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \tag{8.11}$$

swap

in last step, change dummy index i to j and j to i

In writing the final form, we have merely emphasized that any symbol may be used for the dummy subscripts. Namely, we may swap the symbols i and j without loss. The ji component of $\underset{\approx}{\boldsymbol{A}}^T$ is the coefficient of $\underset{\sim}{\boldsymbol{e}}_j\underset{\sim}{\boldsymbol{e}}_i$; so the second-to-last expression says that $A_{ji}^T = A_{ij}$. The final expression says that the ij component of $\underset{\approx}{\boldsymbol{A}}^T$ is the coefficient of $\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j$, namely A_{ji}. *Both statements are equivalent — they just use different free index symbols to describe the same result!*

Dotting a tensor from the *left* by a vector

We will define the notation $\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}}$ to be a vector defined such that

$$(\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}}) \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{u}} \bullet (\underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{v}}) \text{ for all vectors } \underset{\sim}{\boldsymbol{v}} \tag{8.12}$$

Following an analysis similar to Eq. (8.1) we write

$$\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} = \underset{\sim}{\boldsymbol{u}} \bullet (A_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j) = A_{ij}(\underset{\sim}{\boldsymbol{u}} \bullet \underset{\sim}{\boldsymbol{e}}_i)\underset{\sim}{\boldsymbol{e}}_j = A_{ij}(u_i)\underset{\sim}{\boldsymbol{e}}_j = (u_iA_{ij})\underset{\sim}{\boldsymbol{e}}_j \tag{8.13}$$

Thus,

$$\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} \text{ is a vector whose } j^{th} \text{ component is } u_iA_{ij}. \tag{8.14}$$

Equivalently, changing the symbols for the indices,

$$\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} \text{ is a vector whose } i^{th} \text{ component is } u_j A_{ji} \tag{8.15}$$

For both Eqs. (8.2) and (8.15), the index that is summed is on the same side of the [A] matrix as the dot product ($\underset{\sim}{\boldsymbol{u}}$ is dotting into $\underset{\approx}{\boldsymbol{A}}$ from the left, and the summed index on A_{ij} is the *left* index). The transpose of $\underset{\approx}{\boldsymbol{A}}$ is a new tensor $\underset{\approx}{\boldsymbol{A}}^T$ defined such that $(\underset{\approx}{\boldsymbol{A}}^T)_{ij} = A_{ji}$. Thus, we note that

$$\boxed{\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} = \underset{\approx}{\boldsymbol{A}}^T \bullet \underset{\sim}{\boldsymbol{u}}} \tag{8.16}$$

Likewise,

$$\boxed{\underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{u}} = \underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}}^T} \tag{8.17}$$

Dotting a tensor by vectors from both sides

The notation $\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{v}}$ is defined in a notationally consistent manner. Namely,

$$\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{v}} = (\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}}) \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{u}} \bullet (\underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{v}}) = u_i A_{ij} v_j \tag{8.18}$$

There are no free indices, so this result is a scalar.

For advanced analysis, it is often useful to recognize that

$$\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\approx}{\boldsymbol{A}} \boldsymbol{:} \underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}} = \underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}} \boldsymbol{:} \underset{\approx}{\boldsymbol{A}} \tag{8.19}$$

Where "**:**" is the tensor inner product defined later in Eq. (12.1).

Extracting a particular tensor component

Using techniques similar to those used to derive Eq. (5.13) one can prove that

$$A_{ij} = \underset{\sim}{\boldsymbol{e}}_i \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{e}}_j = \underset{\approx}{\boldsymbol{A}} \boldsymbol{:} \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \tag{8.20}$$

By virtue of Eq. (8.12), parentheses are not needed in this formula.

Dotting a tensor into a tensor (tensor composition)

Heuristically, you may infer the meaning of $\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}$ by expanding each tensor in its basis form as follows

$$\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}} = A_{ij} \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \bullet B_{kl} \underset{\sim}{\boldsymbol{e}}_k \underset{\sim}{\boldsymbol{e}}_l \tag{8.21}$$

Applying Eq. (7.44), this may be written

$$\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}} = A_{ij} B_{kl} \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \bullet \underset{\sim}{\boldsymbol{e}}_k \underset{\sim}{\boldsymbol{e}}_l \tag{8.22}$$

Applying Eq. (5.8), we note that $\underset{\sim}{\boldsymbol{e}}_j \bullet \underset{\sim}{\boldsymbol{e}}_k = \delta_{jk}$ so that

$$\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}} = A_{ij} B_{kl} \underset{\sim}{\boldsymbol{e}}_i \delta_{jk} \underset{\sim}{\boldsymbol{e}}_l \tag{8.23}$$

Again applying Eq. (7.44), this may be written

$$\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}} = A_{ij}B_{kl}\delta_{jk}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_l \tag{8.24}$$

Finally, using the Kronecker-removal rule (Eq. 4.43), we may eliminate the δ_{jk} if all occurrences of the summed index j are replaced with k, giving

$$\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}} = A_{ik}B_{kl}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_l \tag{8.25}$$

This result is a linear combination of $A_{ik}B_{kl}$ times the dyad $\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_l$. Therefore, $\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}$ must represent a tensor whose il component is given by $A_{ik}B_{kl}$. Later on, the linear transformation operator associated with $\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}$ will be seen to represent the composition of the linear transformations associated individually with $\underset{\approx}{\boldsymbol{A}}$ and $\underset{\approx}{\boldsymbol{B}}$.

“Everything comes to he who hustles while he waits.” — Thomas Edison

9. Tensor analysis primitives

Three kinds of vector and tensor notation

Three kinds of notations dominate the literature: direct, basis, and indicial. “Direct,” or **structured**, notation uses underlined symbols with no reference to a basis. Indicial notation shows strictly the component form of an expression. Basis notation is similar to indicial notation except that the associated basis is shown explicitly.

Rather than giving formal definitions, we will merely show the distinctions by examples:

Example 1: Representations of a vector

direct: $\underset{\sim}{v}$ (no indices, no base vectors)

basis: $v_i\underset{\sim}{e}_i$ (no free indices, one base vector → result is a vector)

indicial: v_i (one free index → result corresponds to a vector)

Example 2: A linear transformation of a vector

direct: $\underset{\approx}{A} \bullet \underset{\sim}{v}$ (no indices, no base vectors, abstract operation symbol)

basis: $(A_{ij}v_j)\underset{\sim}{e}_i$ (no free indices, one base vector → result is a vector)

indicial: $A_{ij}v_j$ (one free index → result corresponds to a vector)

Example 3: The composition of two tensors

direct: $\underset{\approx}{A} \bullet \underset{\approx}{B}$ (no indices, no base vectors, abstract operation symbol)

basis: $(A_{ij}B_{jk})\underset{\sim}{e}_i\underset{\sim}{e}_k$ (no free indices, two base vectors → result is a tensor)

indicial: $A_{ij}B_{jk}$ (two free indices → result corresponds to a tensor)

Example 4: The inner product of two vectors

direct: $\underset{\sim}{a} \bullet \underset{\sim}{b}$ (no indices, no base vectors, abstract operation symbol)

basis: a_kb_k (no free indices, no base vectors → result is a scalar)

indicial: a_kb_k (no free indices → result corresponds to a scalar)

When counting base vectors in an expression given using basis notation, it is important to understand that this count must occur after the expression has been simplified into a form that is identical to the indicial form except also multiplied by the base vectors corresponding to the free indices in the indicial expression. For example, $(a_i \underset{\sim}{\mathbf{e}}_i) \bullet (b_k \underset{\sim}{\mathbf{e}}_k)$ is expressed using basis notation, but the presence of two base vectors does not mean it corresponds to a tensor; the dot product between $\underset{\sim}{\mathbf{e}}_i$ and $\underset{\sim}{\mathbf{e}}_k$ becomes simply δ_{ik} so that the expression simplifies to $a_i b_k \delta_{ik}$ or simply $a_k b_k$, which has no base vectors and is therefore a scalar.

Direct notation is useful for conceptual discussions. As long as the abstract operations are well-defined, it is straightforward to convert from direct notation to any of the other notations. The principal *disadvantage* of direct notation is that the meanings of the abstract operations can vary widely in the literature. For example, many authors define the notation $\mathbf{A}\mathbf{v}$ to mean what we have opted to denote by $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}}$. Also, it is often more difficult to infer the order of the result when using direct notation. It is a bit easier, though, with our "under-tilde" convention. To infer the order of the result, first you count the number of under-tildes (for example, $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}}$ has three under-tildes); then you subtract from this number by an amount appropriate to operations in the expression. For example, the "single" dot operation will always reduce the count by 2. Three minus two equals one; therefore, $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{v}}$ must be a first-order tensor (i.e., a vector). The cross product operation will reduce the count by 1. The double dot operation "**:**" (see Eq. 3.60) reduces the count by four.

An extremely important use for direct notation arises in vector and tensor calculus, where the meaning of, say, the curl for an obscure curvilinear coordinate system can be simply "looked up" in a handbook. Likewise, the appropriate component form for, say, the dot product for a non-rectangular basis can be looked up in a handbook. You can perform all of your physical analyses in Cartesian indicial form and then simply express the final result in direct notation. As long as every step in your analysis involved valid vector and tensor operations (vector addition, dot products, cross products, dyadic products, gradient operations, etc.), then straightforward rules exist to convert your final *direct notation* result into any other notation or to any other coordinate system without error — it is not necessary to re-do your entire analysis when you switch notations or coordinates!

Direct notation formulas also tend to exhibit more obvious connections with similar formulas for scalars. For example, the time rate of the inverse of a tensor is given by

$$\frac{d(\underset{\approx}{\mathbf{F}}^{-1})}{dt} = -\underset{\approx}{\mathbf{F}}^{-1} \bullet \frac{d\underset{\approx}{\mathbf{F}}}{dt} \bullet \underset{\approx}{\mathbf{F}}^{-1}, \tag{9.1}$$

which is clearly analogous to the scalar formula

$$\frac{d\left(\frac{1}{x}\right)}{dt} = -\frac{dx/dt}{x^2} \tag{9.2}$$

Tensor gradients are another good example. We can define direct notation symbols for the derivative of scalars, vectors, or tensors with respect to other scalars, vectors, or tensors as follows:

$$\frac{ds}{d\underset{\sim}{\boldsymbol{x}}} \equiv \frac{\partial s}{\partial x_k}\underset{\sim}{\boldsymbol{e}}_k, \qquad \frac{d\underset{\sim}{\boldsymbol{u}}}{d\underset{\sim}{\boldsymbol{w}}} \equiv \frac{\partial u_i}{\partial w_j}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j, \qquad \frac{d\underset{\sim}{\boldsymbol{u}}}{d\underset{\approx}{\boldsymbol{A}}} \equiv \frac{\partial u_i}{\partial A_{jk}}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\underset{\sim}{\boldsymbol{e}}_k, \quad \text{etc.} \tag{9.3}$$

By using a notation such as this, direct notation expressions for the chain rule take forms that are very similar to what is already familiar for scalars. For example, it can be shown that

$$\left[\frac{d\underset{\sim}{\boldsymbol{u}}}{d\underset{\sim}{\boldsymbol{w}}}\right]^{-1} = \frac{d\underset{\sim}{\boldsymbol{w}}}{d\underset{\sim}{\boldsymbol{u}}} \tag{9.4}$$

Suppose, as another example, that a scalar s is given by a function of another scalar ψ, a vector $\underset{\sim}{a}$, and a tensor $\underset{\approx}{F}$. If each of these arguments is expressible as a function of position $\underset{\sim}{x}$ then the gradient of s with respect to position can be computed in direct notation by using the chain rule:

$$\frac{ds}{d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial s}{\partial\psi}\frac{d\psi}{d\underset{\sim}{\boldsymbol{x}}} + \frac{\partial s}{\partial\underset{\sim}{\boldsymbol{a}}} \bullet \frac{d\underset{\sim}{\boldsymbol{a}}}{d\underset{\sim}{\boldsymbol{x}}} + \frac{\partial s}{\partial\underset{\approx}{\boldsymbol{F}}} : \frac{d\underset{\approx}{\boldsymbol{F}}}{d\underset{\sim}{\boldsymbol{x}}} \tag{9.5}$$

where the "**:**" symbol denotes the second-order tensor inner product, defined later. This formula looks like an ordinary application of the chain rule except that the appropriate inner product is used between factors. The indicial form of the above equation would be

$$\frac{\partial s}{\partial x_i} = \frac{\partial s}{\partial\psi}\frac{\partial\psi}{\partial x_i} + \frac{\partial s}{\partial a_k}\frac{\partial a_k}{\partial x_i} + \frac{\partial s}{\partial F_{mn}}\,\frac{\partial F_{mn}}{\partial x_i} \tag{9.6}$$

Basis notation is useful when working with more than one basis, or as an intermediate notation when converting a direct notation expression to indicial form. Indicial notation is often the most clear, though an alphabet soup of indices can be distracting when discussing conceptual issues. Furthermore, the constant task of looking after the summation rules (changing j's to k's, etc.) can be wearisome and error-prone. There are times when even indicial notation is vague. For example, what does the expression $\partial f/\partial\sigma_{ii}$ mean? Is it $\partial f/\partial(\text{tr}\underset{\approx}{\sigma})$ or is it $\text{tr}[\partial f/\partial\underset{\approx}{\sigma}]$? The two are not the same. Likewise, does A^{-1}_{ij} mean $(\underset{\approx}{A}^{-1})_{ij}$ or $1/A_{ij}$. Sometimes these questions can be answered by defining operator precedence* or simply from context. Nonetheless, these issues are characteristic of potential flaws with indicial notation. The bottom line is that all three notations are useful, and none

* A good rule of thumb is that the indices take lowest precedence. Thus A^{-1}_{ij} would be found by taking the inverse of $\underset{\approx}{\boldsymbol{A}}$ and *then* finding the ij component.

is perfect. A competent analyst should learn all notations and pick the one that is clearest for the application at hand. A courteous analyst will always attempt to infer and use whichever notational system their audience (or the majority of their audience) will be most comfortable with.

REPRESENTATION THEOREM for linear forms

GOAL: Explain that there exists a unique tensor that characterizes each function that linearly transforms vectors to vectors.

If a vector-valued function $f(\underset{\sim}{x})$ is known to be linear, then there exists a second-order tensor $\underset{\approx}{F}$ which is independent of $\underset{\sim}{x}$ such that the function $f(\underset{\sim}{x})$ can be written in the form $\underset{\approx}{F} \bullet \underset{\sim}{x}$. Stated mathematically,

> If $f(\underset{\sim}{x})$ is a vector valued function that is linear in $\underset{\sim}{x}$, then there exists a tensor $\underset{\approx}{F}$ such that $f(\underset{\sim}{x}) = \underset{\approx}{F} \bullet \underset{\sim}{x}$, where $\underset{\approx}{F}$ will be independent of $\underset{\sim}{x}$. (9.7)

The converse is also true. Specifically, any expression of the form $\underset{\approx}{F} \bullet \underset{\sim}{x}$ can be proved to be linear with respect to $\underset{\sim}{x}$ if you can prove that $\underset{\approx}{F}$ is independent of $\underset{\sim}{x}$. This might seem like an obvious statement, but it can be subtle in certain applications.*

If $f(\underset{\sim}{x})$ is linear in $\underset{\sim}{x}$, then the equation $\underset{\sim}{y} = f(\underset{\sim}{x})$ can (and should) be written $\underset{\sim}{y} = \underset{\approx}{F} \bullet \underset{\sim}{x}$, which is expressed in indicial notation as

$$y_i = F_{ij} x_j \tag{9.8}$$

where the associated base vector $\underset{\sim}{e}_i$ is understood to be multiplied by these components. If the vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$ are written as 3×1 matrices and the tensor $\underset{\approx}{F}$ is written as a 3×3 matrix, Eq. (9.8) can be written

$$\begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \tag{9.9}$$

Incidentally, we have used the phrase "if $f(\underset{\sim}{x})$ is linear in $\underset{\sim}{x}$" as a sloppy way to indicate that the function f might depend additionally on other variables that we have not shown explicitly in the independent variable list. If, for example, you are dealing with a function $f(\underset{\sim}{x}, \phi, \underset{\sim}{c})$ where ϕ is a scalar and $\underset{\sim}{c}$, then this function is said to be linear in $\underset{\sim}{x}$ if

$$f(\alpha \underset{\sim}{u} + \beta \underset{\sim}{v}, \phi, \underset{\sim}{c}) = \alpha f(\underset{\sim}{u}, \phi, \underset{\sim}{c}) + \beta f(\underset{\sim}{v}, \phi, \underset{\sim}{c}) \tag{9.10}$$

* For example, if $\underset{\approx}{R}$ represents an orthogonal tensor defined such that $\underset{\approx}{R} \bullet \underset{\sim}{x}$ will rotate a vector $\underset{\sim}{x}$ by an angle α about an axis $\underset{\sim}{a}$, then you can use this operation in a description of torsion of a circular rod to describe the movement of points on the rod. However, for torsion, the angle of rotation increases with distance along the rod. Hence, since $\underset{\approx}{R}$ depends on the angle of rotation, it must also depend on position. Consequently, $\underset{\approx}{R} \bullet \underset{\sim}{x}$ would be a *nonlinear* transformation of $\underset{\sim}{x}$ despite the fact that it appears on the surface to be linear because the spatial dependence of $\underset{\approx}{R}$ is not shown explicitly. There's nothing wrong with this, but you do need to monitor possibilities like this.

for all scalars α and β and all vectors $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$. In this case, there exists a tensor $\underset{\approx}{\boldsymbol{F}}$ that will be independent of $\underset{\sim}{\boldsymbol{x}}$ but will generally depend on ϕ and $\underset{\sim}{\boldsymbol{c}}$ such that $f(\underset{\sim}{\boldsymbol{x}}, \phi, \underset{\sim}{\boldsymbol{c}}) = \underset{\approx}{\boldsymbol{F}}(\phi, \underset{\sim}{\boldsymbol{c}}) \bullet \underset{\sim}{\boldsymbol{x}}$. If the function f depends on ϕ and $\underset{\sim}{\boldsymbol{c}}$ in a nonlinear manner, then so will the tensor $\underset{\approx}{\boldsymbol{F}}$.

Representation theorem for vector-to-scalar linear functions.

Our statements about existence of tensors corresponding to linear functions were given above in the context of vector-to-vector transformations. However, the same idea applies to tensors of other orders as well. If, for example, $g(\underset{\sim}{\boldsymbol{x}})$ is a *scalar*-valued function that depends linearly on its vector argument $\underset{\sim}{\boldsymbol{x}}$, then you can assert existence of a *vector* $\underset{\sim}{\boldsymbol{g}}$ such that $g(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{g}} \bullet \underset{\sim}{\boldsymbol{x}}$.

> If $f(\underset{\sim}{\boldsymbol{x}})$ is a *scalar*-valued function that is linear in $\underset{\sim}{\boldsymbol{x}}$, then there exists a vector $\underset{\sim}{\boldsymbol{g}}$ such that $f(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{g}} \bullet \underset{\sim}{\boldsymbol{x}}$, where $\underset{\sim}{\boldsymbol{g}}$ will be independent of $\underset{\sim}{\boldsymbol{x}}$. (9.11)

Consider, for example, the equation $s = (\underset{\sim}{\boldsymbol{b}} \times \underset{\sim}{\boldsymbol{x}}) \bullet \underset{\sim}{\boldsymbol{a}}$, where $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{b}}$ are known constant vectors. The right-hand-side of this expression is linear with respect to $\underset{\sim}{\boldsymbol{x}}$. Consequently, we should be able to construct a vector $\underset{\sim}{\boldsymbol{g}}$ that depends on $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{b}}$ but not on $\underset{\sim}{\boldsymbol{x}}$ such that $s = \underset{\sim}{\boldsymbol{g}} \bullet \underset{\sim}{\boldsymbol{x}}$. To find $\underset{\sim}{\boldsymbol{g}}$, the simplest method is to write the requirement

$$(\underset{\sim}{\boldsymbol{b}} \times \underset{\sim}{\boldsymbol{x}}) \bullet \underset{\sim}{\boldsymbol{a}} = \underset{\sim}{\boldsymbol{g}} \bullet \underset{\sim}{\boldsymbol{x}} \qquad (9.12)$$

in indicial form. Namely,

$$\varepsilon_{ijk} b_j x_k a_i = g_k x_k \qquad (9.13)$$

Since this must hold for all x_k, we conclude that

$$g_k = \varepsilon_{ijk} b_j a_i \qquad (9.14)$$

or

$$\underset{\sim}{\boldsymbol{g}} = \underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{b}} \qquad (9.15)$$

In hindsight, you might look at this result to see that you could have done the entire proof in direct (structured) notation by noting that $(\underset{\sim}{\boldsymbol{b}} \times \underset{\sim}{\boldsymbol{x}}) \bullet \underset{\sim}{\boldsymbol{a}}$ is the triple-scalar-product, $[\underset{\sim}{\boldsymbol{b}}, \underset{\sim}{\boldsymbol{x}}, \underset{\sim}{\boldsymbol{a}}]$ and, by the cyclic property of the triple-scalar-product, you could assert that it must equal $[\underset{\sim}{\boldsymbol{a}}, \underset{\sim}{\boldsymbol{b}}, \underset{\sim}{\boldsymbol{x}}]$ or $(\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{b}}) \bullet \underset{\sim}{\boldsymbol{x}}$. It is more easy to compare this re-written expression with $\underset{\sim}{\boldsymbol{g}} \bullet \underset{\sim}{\boldsymbol{x}}$ to immediately recognize that $\underset{\sim}{\boldsymbol{g}} = \underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{b}}$. In practice, elegant solutions like this one might be difficult to see *a priori*, and doing it indicially is often easier.

Incidentally, note that the index "k" on the right side of Eq. (9.13) is only a dummy index. It was just fortunate happenstance that the same dummy index was on x_k on both sides of that equation. You must never go from an equation like (9.13) to a conclusion like (9.14) unless you have the dummy indices all matching on the variable you wish to "cancel". If we had been inattentive and written the indicial form of $\underset{\sim}{\boldsymbol{g}} \bullet \underset{\sim}{\boldsymbol{x}}$ as $g_i x_i$, then Eq. (9.13) would have read

$$\varepsilon_{ijk} b_j x_k a_i = g_i x_i \qquad (9.16)$$

This is a legitimate indicial equation (no summation conventions are violated), but it would be incorrect to simply remove the x's to write $\varepsilon_{ijk}b_ja_i = g_i$; doing that would violate the sum conventions because "i" would be a dummy sum index on the left side but a free index on the right side. When facing a situation like this one when the indices are not all identical on the variable you wish to "cancel" you can use the methods on page 67 to manipulate the indices to desired symbols. Before you "cancel" a variable x_k, however, you *must* assert that the equation holds *for all* x_k (see Eq. 5.11).

Advanced Representation Theorem (to be read once you learn about higher-order tensors and the $V_m{}^n$ class notation).

We have described the representation theorem for vector-to-vector transformations and for vector-to-scalar transformations. The representation theorem applies to any function that linearly transforms a vector or tensor of class V_3^m into V_3^n. The representation theorem ensures existence of a tensor of class V_3^{m+n} that characterizes the linear function. Specifically, the linear form can be written as this V_3^{m+n} tensor multiplied by the input argument using the inner product for the input space, V_3^m.

> If $f(\boldsymbol{x})$ is a V_3^n-valued function that is linear with respect to a V_3^m tensor $\boldsymbol{x}$, then there exists a V_3^{m+n} tensor $\boldsymbol{H}$ such that $f(\boldsymbol{x}) = \boldsymbol{H}*\boldsymbol{x}$, where $*$ is the inner product on V_3^m and $\boldsymbol{H}$ will be independent of $\boldsymbol{x}$. (9.17)

Suppose, for example that $h(\underset{\sim}{\boldsymbol{x}})$ is linear transformation of a vector $\underset{\sim}{\boldsymbol{x}}$ that produces a second-order tensor as its output. The input $\underset{\sim}{\boldsymbol{x}}$ is of class V_3^1 and the output (a second-order tensor) is of class V_3^2. Because the transformation is linear, you can assert the existence of a tensor of class V_3^{1+2}, or V_3^3 that characterizes the transformation using the inner product on the input space. The input space is V_3^1, which is just ordinary vectors, so the inner product there is just the single raised dot. Thus, if $h(\underset{\sim}{\boldsymbol{x}})$ is linear transformation of a vector $\underset{\sim}{\boldsymbol{x}}$ that produces a second-order tensor as its output, then there exists a *third-order* tensor $\underset{\underset{\sim}{\approx}}{\boldsymbol{H}}$ such that $h(\underset{\sim}{\boldsymbol{x}}) = \underset{\underset{\sim}{\approx}}{\boldsymbol{H}} \bullet \underset{\sim}{\boldsymbol{x}}$.

The argument to a linear function doesn't have to be a vector. Materials modeling, for example, deals with transformations that take tensors to tensors (e.g., given strain, the model will compute stress). In this case, we are considering linear transformations from V_3^2 to V_3^2, so these can be characterized by a tensor of class V_3^{2+2}, or fourth-order V_3^4 tensors, acting on the input space using the V_3^2 inner product. Thus, if $f(\underset{\approx}{\boldsymbol{X}})$ is a linear function that produces a second-order tensor as output, then you may assert existence of a *fourth-order* tensor $\underset{\underset{\approx}{\approx}}{\boldsymbol{E}}$ such that $f(\underset{\approx}{\boldsymbol{X}}) = \underset{\underset{\approx}{\approx}}{\boldsymbol{E}}:\underset{\approx}{\boldsymbol{X}}$. Note that we are no longer using the *single* dot product. We are using the double-dot product because that is the inner-product appropriate to the argument of the function. If $\underset{\approx}{\boldsymbol{Y}} = f(\underset{\approx}{\boldsymbol{X}})$, then $\underset{\approx}{\boldsymbol{Y}} = \underset{\underset{\approx}{\approx}}{\boldsymbol{E}}:\underset{\approx}{\boldsymbol{X}}$, which can be written in component form as $Y_{ij} = E_{ijkl}X_{kl}$. You can also consider scalar-valued functions of tensors. Consider, for example, the trace of a tensor $g(\underset{\approx}{\boldsymbol{X}}) = \text{tr}\underset{\approx}{\boldsymbol{X}}$. This function is linear in $\underset{\approx}{\boldsymbol{X}}$, so we may assert existence of a tensor $\underset{\approx}{\boldsymbol{G}}$ such that $g(\underset{\approx}{\boldsymbol{X}}) = \underset{\approx}{\boldsymbol{G}}:\underset{\approx}{\boldsymbol{X}}$. For this example, it turns out that $\underset{\approx}{\boldsymbol{G}}$ is the identity tensor, so we may therefore write $\text{tr}\underset{\approx}{\boldsymbol{X}} = \underset{\approx}{\boldsymbol{I}}:\underset{\approx}{\boldsymbol{X}}$.

Why all this fuss over linear functions when we know that Mother Nature plagues us with *non-linear* phenomena? The answer is that linear transformation theory is a generalization of what you learned in high-school about straight lines. The mathematics of straight lines forms the essential foundation for calculus, where nonlinear functions are handled by regarding them as an infinite collection of tiny *straight* line segments. Physical applications of tensors to nonlinear transformations employ similar principles, as will be discussed in Chapter 21.

Finding the tensor associated with a linear function

So far, we have only asserted that a tensor $\underset{\approx}{\boldsymbol{F}}$ exists whenever the function $f(\underset{\sim}{\boldsymbol{x}})$ is linear. In practical applications, the fastest way to construct the F_{ij} matrix is to recognize that the i^{th} column of $[\underset{\approx}{\boldsymbol{F}}]$ contains the components of the vector $f(\underset{\sim}{\boldsymbol{e}}_i)$.

For analytical applications, an indicial expression for the tensor is often desired, and this indicial expression can usually be cast in direct notation as well.

Method #1. The conventional way of finding the tensor associated with a linear transformation is to write out the "i^{th}" component of $f(\underset{\sim}{\boldsymbol{x}})$ in indicial form and then set the result equal to the indicial form of $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}}$, namely $F_{ij}x_j$. In this expression, note that the index "j" is a dummy summation index, and we could have equally well written $F_{ik}x_k$. You should take care to choose the same symbol for the dummy sum index on x everywhere so that you can then assert that the result must hold for all $\underset{\sim}{\boldsymbol{x}}$, permitting you to drop the x_k's from the indicial equation. This approach is used in Eqs. (11.21) and (11.22).

Method #2. The tensor $\underset{\approx}{\boldsymbol{F}}$ can be determined by simply differentiating the function $f(\underset{\sim}{\boldsymbol{x}})$ with respect to $\underset{\sim}{\boldsymbol{x}}$.

$$\left\{\begin{array}{c} \text{if } \underset{\sim}{\boldsymbol{y}} = f(\underset{\sim}{\boldsymbol{x}}) \\ \text{and } f(\underset{\sim}{\boldsymbol{x}}) \text{ is linear in } \underset{\sim}{\boldsymbol{x}} \end{array}\right\},$$

then there exists a tensor $\underset{\approx}{\boldsymbol{F}}$ such that $\underset{\sim}{\boldsymbol{y}} = \underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}}$

$$\text{where } \underset{\approx}{\boldsymbol{F}} = \frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial y_i}{\partial x_j}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \tag{9.18}$$

An example of this statement is given in Eq. (11.23). Incidentally, the derivative $\partial(\quad)/\partial x_j$ denotes the partial derivative with respect to x_j, *holding the other components of* $\underset{\sim}{\boldsymbol{x}}$ *constant.* When applying Eq. (9.7) for finding $\underset{\approx}{\boldsymbol{F}}$, it is often important to recognize that the derivative of a vector with respect to itself is the identity tensor $\underset{\approx}{\boldsymbol{I}}$, as defined below.

$$\frac{d\underset{\sim}{\boldsymbol{x}}}{d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial x_i}{\partial x_j}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j = \delta_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \equiv \underset{\approx}{\boldsymbol{I}} \tag{9.19}$$

This formula is applied, for example, in Eq. (11.25). Unlike method #1, this method does not require you to spend time trying to get the dummy summation index to be the same on all x_k, which can be difficult for complicated expressions.

Method #3. A related "slick" method for finding the tensor is to express the equation $f(\underset{\sim}{x}) = \underset{\approx}{F} \bullet \underset{\sim}{x}$ in rate form. Taking the rate of the right-hand side (permitting only $\underset{\sim}{x}$ to be regarded as time varying) gives $\underset{\approx}{F} \bullet \dot{\underset{\sim}{x}}$. The rate of $f(\underset{\sim}{x})$ can typically be easily rearranging into a form of "something" dotted into $\dot{\underset{\sim}{x}}$ and that "something" must be the tensor $\underset{\approx}{F}$.

EXAMPLE. Consider, for example, the vector-to-vector function $f(\underset{\sim}{x}) = (\underset{\sim}{c} \bullet \underset{\sim}{x})\underset{\sim}{c}$, where $\underset{\sim}{c}$ is a constant vector. This function is linear with respect to which is linear with respect to $\underset{\sim}{x}$, so you can assert existence of a tensor $\underset{\approx}{F}$ such that $f(\underset{\sim}{x}) = \underset{\approx}{F} \bullet \underset{\sim}{x}$.

METHOD #1: The i^{th} component of $f(\underset{\sim}{x})$ is $c_n x_n c_i$, and this result must be equated to the i^{th} component of $\underset{\approx}{F} \bullet \underset{\sim}{x}$, namely $F_{in}x_n$. Note that we used the dummy index symbol "n" on x_n to ensure that it would be the same as the index used in our first expression. Equating the two expressions gives $c_n x_n c_i = F_{in}x_n$. Asserting that this must be true for all x_n shows that $F_{in} = c_n c_i$.

METHOD #2: In indicial form, the equation $f(\underset{\sim}{x}) = \underset{\approx}{F} \bullet \underset{\sim}{x}$ can be written $c_n x_n c_i = F_{ik}x_k$. This time, we didn't bother to get the dummy sum index the same on both x's. Differentiating both sides with respect to x_p gives $c_n \delta_{np} c_i = F_{ik}\delta_{kp}$. Using the index-changing property of the Kronecker delta, this simplifies to $c_p c_i = F_{ip}$, which is the same as what we got using method 1.

METHOD #3: For our example function, the equation $f(\underset{\sim}{x}) = \underset{\approx}{F} \bullet \underset{\sim}{x}$ is $(\underset{\sim}{c} \bullet \underset{\sim}{x})\underset{\sim}{c} = \underset{\approx}{F} \bullet \underset{\sim}{x}$. Taking rates gives $(\underset{\sim}{c} \bullet \dot{\underset{\sim}{x}})\underset{\sim}{c} = \underset{\approx}{F} \bullet \dot{\underset{\sim}{x}}$ rearranging so that the $\dot{\underset{\sim}{x}}$ is on the trailing end of both expressions gives $\underset{\sim}{c}(\underset{\sim}{c} \bullet \dot{\underset{\sim}{x}}) = \underset{\approx}{F} \bullet \dot{\underset{\sim}{x}}$. Asserting this must hold for all $\dot{\underset{\sim}{x}}$ shows that $\underset{\approx}{F} = \underset{\sim}{c}\underset{\sim}{c}$, which is the direct notation version of the result found using methods 1 and 2. This example was a bit trivial, so the power of this method is not readily apparent.

The identity tensor

Consider the following function

$$f(\underset{\sim}{x}) = \underset{\sim}{x} \tag{9.20}$$

Applying Eq. (7.7), this function is seen to be linear in $\underset{\sim}{x}$. Therefore, applying Eq. (9.7), there must exist a second-order tensor, which we will denote $\underset{\approx}{I}$, such that

$$\underset{\approx}{I} \bullet \underset{\sim}{x} = \underset{\sim}{x} \text{ for all } \underset{\sim}{x} \tag{9.21}$$

Recalling Eq. (9.19), the ij components of $\underset{\approx}{I}$ with respect to any orthonormal basis are simply δ_{ij}.

The 3×3 matrix for the second-order identity tensor $\underset{\approx}{I}$ is just the identity matrix

$$[\underset{\approx}{I}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{with respect to any orthonormal basis.} \tag{9.22}$$

Thus, in basis notation, the identity tensor may be written

$$\underset{\approx}{I} = \delta_{ij}\underset{\sim}{e}_i\underset{\sim}{e}_j \tag{9.23}$$

or, expanded out explicitly,

$$\underset{\approx}{I} = \underset{\sim}{e}_1\underset{\sim}{e}_1 + \underset{\sim}{e}_2\underset{\sim}{e}_2 + \underset{\sim}{e}_3\underset{\sim}{e}_3 = \underset{\sim}{e}_k\underset{\sim}{e}_k \tag{9.24}$$

An alternative way to derive Eq. (9.22) is to recall that the i^{th} column of the tensor must contain $f(\underset{\sim}{e}_i)$. Thus, the first column of the tensor must contain the component array for $f(\underset{\sim}{e}_1) = \underset{\sim}{e}_1$. The component array for $\underset{\sim}{e}_1$ is

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{9.25}$$

so this must be the first column of the matrix. The second and third columns are the component arrays for $\underset{\sim}{e}_2$ and $\underset{\sim}{e}_3$, respectively. Thus we obtain the same matrix as in Eq. (9.22).

Tensor associated with composition of two linear transformations

Recall[*] that for any linear vector-to-vector transformation, $f(\underset{\sim}{x})$, there exists a tensor $\underset{\approx}{F}$ such that

$$f(\underset{\sim}{x}) = \underset{\approx}{F} \bullet \underset{\sim}{x} \tag{9.26}$$

Likewise, for any linear vector-to-vector transformation, $g(\underset{\sim}{x})$, there exists a tensor $\underset{\approx}{G}$ such that

$$g(\underset{\sim}{x}) = \underset{\approx}{G} \bullet \underset{\sim}{x} \tag{9.27}$$

Therefore, the composition of the two transformations must be given by

$$f(g(\underset{\sim}{x})) = \underset{\approx}{F} \bullet (\underset{\approx}{G} \bullet \underset{\sim}{x}) \tag{9.28}$$

The composition of two linear functions can be readily verified to be itself a linear function. Hence, recalling Eq. (9.7), there must exist a tensor, which we will denote $\underset{\approx}{F} \bullet \underset{\approx}{G}$ such that

$$\underset{\sim}{f}(\underset{\sim}{g}(\underset{\sim}{x})) = (\underset{\approx}{F} \bullet \underset{\approx}{G}) \bullet \underset{\sim}{x} \qquad \text{for all vectors } \underset{\sim}{x} \tag{9.29}$$

The components of $\underset{\approx}{F} \bullet \underset{\approx}{G}$ are

[*] from the Representation Theorem, Eq. (9.7)

$$(\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}})_{ij} = F_{ik}G_{kj} \tag{9.30}$$

Hence, the matrix for $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}}$ may be found by the simple matrix multiplication, $[F][G]$.

Eqs. (9.28) and (9.29) must both hold simultaneously for all $\underset{\sim}{\boldsymbol{x}}$ vectors. Hence, the direct notation statement,

$$(\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}}) \bullet \underset{\sim}{\boldsymbol{x}} = \underset{\approx}{\boldsymbol{F}} \bullet (\underset{\approx}{\boldsymbol{G}} \bullet \underset{\sim}{\boldsymbol{x}}) \quad \text{for all } \underset{\sim}{\boldsymbol{x}} \tag{9.31}$$

serves as a definition of what is meant by $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}}$.

The power of heuristically consistent notation

In Eq. (9.31), we have *chosen* a heuristically appealing notation for $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}}$ so that we may drop the parentheses to write

$$\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}} \bullet \underset{\sim}{\boldsymbol{x}} \tag{9.32}$$

without ambiguity. Many authors prefer the more-easily typeset notation $\boldsymbol{FG}$ to denote the composition, so they would not be able to drop the parentheses. Our notation is more consistent because you can always heuristically decipher what it means. For example, recalling that

$$\underset{\approx}{\boldsymbol{F}} = F_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \text{ and } \underset{\approx}{\boldsymbol{G}} = G_{kl}\underset{\sim}{\boldsymbol{e}}_k\underset{\sim}{\boldsymbol{e}}_l, \tag{9.33}$$

you can heuristically write

$$\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}} = (F_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j) \bullet (G_{kl}\underset{\sim}{\boldsymbol{e}}_k\underset{\sim}{\boldsymbol{e}}_l) = F_{ij}G_{kl}(\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j) \bullet (\underset{\sim}{\boldsymbol{e}}_k\underset{\sim}{\boldsymbol{e}}_l) \tag{9.34}$$

Using Eq. (5.8), the user can note that $\underset{\sim}{\boldsymbol{e}}_j \bullet \underset{\sim}{\boldsymbol{e}}_k = \delta_{jk}$, so that the above equation would become

$$\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}} = F_{ij}G_{kl}\delta_{jk}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_l \tag{9.35}$$

Finally, noting that δ_{jk} is zero unless, $j=k$, this becomes

$$\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}} = F_{ik}G_{kl}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_l \tag{9.36}$$

The remaining dyad is $\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_l$. Hence, the il components of $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}}$ must be

$$(\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{G}})_{il} = F_{ik}G_{kl} \tag{9.37}$$

Aside from our choice of dummy and free indices, this result is identical to (9.30). Throughout this book, we define our notation in such a manner that heuristic analyses like these will *always* give the correct interpretation of the notation. Thus, for example, the cross product between a tensor and a vector would be interpreted

$$\begin{aligned} \underset{\approx}{\boldsymbol{A}} \times \underset{\sim}{\boldsymbol{v}} &= (A_{ij}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j) \times (v_k\underset{\sim}{\boldsymbol{e}}_k) \\ &= (A_{ij}v_k\underset{\sim}{\boldsymbol{e}}_i)(\underset{\sim}{\boldsymbol{e}}_j \times \underset{\sim}{\boldsymbol{e}}_k) \end{aligned} \tag{9.38}$$

Applying Eq. (5.33), the above equation becomes

$$\underset{\approx}{\mathbf{A}} \times \underset{\sim}{\mathbf{v}} = A_{ij} v_k \varepsilon_{pjk} \underset{\sim}{\mathbf{e}}_i \underset{\sim}{\mathbf{e}}_p \tag{9.39}$$

This final form contains a dyad $\underset{\sim}{\mathbf{e}}_i \underset{\sim}{\mathbf{e}}_p$. The reader (who might never have heard of crossing a tensor into a vector) would be able to conclude that $\underset{\approx}{\mathbf{A}} \times \underset{\sim}{\mathbf{v}}$ must be a second-order tensor, with ip components given by

$$(\underset{\approx}{\mathbf{A}} \times \underset{\sim}{\mathbf{v}})_{ip} = A_{ij} v_k \varepsilon_{pjk} \tag{9.40}$$

Similarly, a reader would be able to deduce what it means when *we* write two tensors written side-by-side with no symbol between. Following *our* notational conventions, the result is a fourth-order tensor,

$$\underset{\approx}{\mathbf{A}} \underset{\approx}{\mathbf{B}} = A_{ij} B_{kl} \underset{\sim}{\mathbf{e}}_i \underset{\sim}{\mathbf{e}}_j \underset{\sim}{\mathbf{e}}_k \underset{\sim}{\mathbf{e}}_l \tag{9.41}$$

We have already introduced the notion that any linear combination of $\underset{\sim}{\mathbf{e}}_i$'s is a vector, and any linear combination of $\underset{\sim}{\mathbf{e}}_i \underset{\sim}{\mathbf{e}}_j$ dyads is a second-order tensor. Generalizing this idea, we will later introduce basis "triads" $\underset{\sim}{\mathbf{e}}_i \underset{\sim}{\mathbf{e}}_j \underset{\sim}{\mathbf{e}}_k$ which will be identified with third order tensors. Likewise, any linear combination of $\underset{\sim}{\mathbf{e}}_i \underset{\sim}{\mathbf{e}}_j \underset{\sim}{\mathbf{e}}_k \underset{\sim}{\mathbf{e}}_l$ must be a fourth-order tensor. Hence, the notation $\underset{\approx}{\mathbf{A}} \underset{\approx}{\mathbf{B}}$ in Eq. (9.41) must represent a fourth-order tensor whose $ijkl$ components must be $A_{ij} B_{kl}$. Recall that $\underset{\sim}{\mathbf{u}} \otimes \underset{\sim}{\mathbf{v}}$ means the same thing as $\underset{\sim}{\mathbf{u}} \underset{\sim}{\mathbf{v}}$. Likewise, when publishing for a journal article, it might be wise to use the notation $\underset{\approx}{\mathbf{A}} \otimes \underset{\approx}{\mathbf{B}}$ instead of $\underset{\approx}{\mathbf{A}} \underset{\approx}{\mathbf{B}}$ because your readers might confuse $\underset{\approx}{\mathbf{A}} \underset{\approx}{\mathbf{B}}$ with tensor composition, which we here denote as $\underset{\approx}{\mathbf{A}} \bullet \underset{\approx}{\mathbf{B}}$.

The inverse of a tensor

Recall from Eq. (9.29) the two notations for the composition of two linear operators f and g

$$f(g(\underset{\sim}{\mathbf{x}})) = (\underset{\approx}{\mathbf{F}} \bullet \underset{\approx}{\mathbf{G}}) \bullet \underset{\sim}{\mathbf{x}} \quad \text{for all vectors } \underset{\sim}{\mathbf{x}} \tag{9.42}$$

Now suppose that f is defined to "undo" the action of g so that

$$f(g(\underset{\sim}{\mathbf{x}})) = \underset{\sim}{\mathbf{x}} \quad \text{for all vectors } \underset{\sim}{\mathbf{x}} \tag{9.43}$$

Then f is called the *inverse* of g, and is denoted g^{-1}. The tensor associated with the inverse of g is denoted $\underset{\approx}{\mathbf{G}}^{-1}$. Hence, putting Eq. (9.43) into (9.42) gives the tensor definition of the inverse:

$$\underset{\sim}{\mathbf{x}} = (\underset{\approx}{\mathbf{G}}^{-1} \bullet \underset{\approx}{\mathbf{G}}) \bullet \underset{\sim}{\mathbf{x}} \quad \text{for all vectors } \underset{\sim}{\mathbf{x}} \tag{9.44}$$

In other words, the tensor inverse is defined such that

$$\underset{\approx}{\mathbf{G}}^{-1} \bullet \underset{\approx}{\mathbf{G}} = \underset{\approx}{\mathbf{I}}, \tag{9.45}$$

which means that the *matrix* for $\underset{\approx}{\mathbf{G}}^{-1}$ is obtained by inverting the matrix for $\underset{\approx}{\mathbf{G}}$.

The COFACTOR tensor

In direct notation [see Ref. 21], the cofactor tensor $\underset{\approx}{\mathbf{G}}^C$ is defined by

$$\underset{\approx}{\mathbf{G}}^C \bullet (\underset{\sim}{\mathbf{u}} \times \underset{\sim}{\mathbf{v}}) = (\underset{\approx}{\mathbf{G}} \bullet \underset{\sim}{\mathbf{u}}) \times (\underset{\approx}{\mathbf{G}} \bullet \underset{\sim}{\mathbf{v}}) \quad \text{for all vectors } \underset{\sim}{\mathbf{u}} \text{ and } \underset{\sim}{\mathbf{v}} \tag{9.46}$$

This definition might seem quite strange until you think about it physically. Recalling Fig. 5.2, the cross product $\underset{\sim}{\boldsymbol{u}} \times \underset{\sim}{\boldsymbol{v}}$ is perpendicular to the plane spanned by $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$ and its magnitude equals the area of the parallelogram formed by $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$. Suppose that all vectors are now transformed by a linear operator $\underset{\approx}{\boldsymbol{G}}$. Then the vector $\underset{\sim}{\boldsymbol{u}}$ becomes $\underset{\approx}{\boldsymbol{G}} \bullet \underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$ becomes $\underset{\approx}{\boldsymbol{G}} \bullet \underset{\sim}{\boldsymbol{v}}$. Thus, the vector on the right-hand-side of Eq. (9.46) is the *new area vector* formed by the *transformed* vectors. Though not at all obvious, it turns out that this *new* area vector is linearly related to the *old* area vector $\underset{\sim}{\boldsymbol{u}} \times \underset{\sim}{\boldsymbol{v}}$. Since the relationship between the old and new areas is linear, Eq. (9.7) tells us there must exist a tensor, which we denote $\underset{\approx}{\boldsymbol{G}}^C$, that can act on the old area to give the new area.

In direct notation, Eq. (9.46) is written

$$G^C_{ij}(\varepsilon_{jkm}u_k v_m) = \varepsilon_{irs}(G_{rk}u_k)(G_{sm}v_m) \tag{9.47}$$

or, since this must hold for all u_k and v_m,

$$G^C_{ij}(\varepsilon_{jkm}) = \varepsilon_{irs}G_{rk}G_{sm}\,. \tag{9.48}$$

Multiplying both sides by ε_{pkm} and using Eq. (3.87) shows that

$$2G^C_{pj} = \varepsilon_{irs}G_{rk}G_{rm}\varepsilon_{pkm} \tag{9.49}$$

or, rearranging and changing the symbols used for the free indices,

$$\boxed{G^C_{ij} = \tfrac{1}{2}\,\varepsilon_{ipq}G_{pr}G_{qs}\varepsilon_{rsj}} \tag{9.50}$$

In direct notation,

$$\underset{\approx}{\boldsymbol{G}}^C = \frac{1}{2}\underset{\approx}{\boldsymbol{G}}:\underset{\approx\approx\approx}{\boldsymbol{H}}:\underset{\approx}{\boldsymbol{G}},$$

where $\underset{\approx\approx\approx}{\boldsymbol{H}}$ is a *sixth-order* tensor* whose components are given by

$$H_{ijmnpr} = \varepsilon_{imp}\varepsilon_{jnr} \tag{9.51}$$

Writing out Eq. (9.50) explicitly reveals that the component *matrix* associated with the cofactor *tensor* equals the cofactor of the component matrix for $[\underset{\approx}{\boldsymbol{G}}]$ (see Eq. 3.94):

$$[\underset{\approx}{\boldsymbol{G}}^C] = [\underset{\approx}{\boldsymbol{G}}]^C \tag{9.52}$$

* Clearly, the teaching benefit of putting as many under-tildes beneath a tensor as the order of the tensor is reaching the point of comically diminishing returns — typesetting them is hard for the writer and counting them is hard for the reader.

Incidentally, the *transpose* of the cofactor tensor is often called the “adjugate” tensor. Unfortunately, some writers also call this the “adjoint,” but this is a dangerous misnomer (perhaps originating from mis-hearing the word “adjugate”). When applied to tensors, the term “adjoint” should be defined to mean the complex conjugate of the transpose — at least that’s how it should be defined if you want your lexicon to agree with that of mathematicians. Of course, if the components are real the adjoint is just the transpose, and it is not the same as the adjugate. We use the term “cofactor” to side-step the whole ugly issue.

Examination of Eq. (9.50) reveals that the cofactor of the transpose equals the transpose of the cofactor. In other words, these two operations *commute*, and you can write $\underset{\approx}{\boldsymbol{G}}^{CT} = (\underset{\approx}{\boldsymbol{G}}^{C})^{T} = (\underset{\approx}{\boldsymbol{G}}^{T})^{C} = \underset{\approx}{\boldsymbol{G}}^{TC}$.

The cofactor tensor $\underset{\approx}{\boldsymbol{G}}^{C}$ is well defined even if $\underset{\approx}{\boldsymbol{G}}$ is non-invertible. However, if $\underset{\approx}{\boldsymbol{G}}$ happens to be invertible, then

$$\underset{\approx}{\boldsymbol{G}}^{C} = \det(\underset{\approx}{\boldsymbol{G}})\ \underset{\approx}{\boldsymbol{G}}^{-T} \tag{9.53}$$

Incidentally, if we define $\underset{\sim}{\boldsymbol{A}}_o \equiv \underset{\sim}{\boldsymbol{u}} \times \underset{\sim}{\boldsymbol{v}}$ to be an “initial” area vector and if we define $\underset{\sim}{\boldsymbol{A}} \equiv (\underset{\approx}{\boldsymbol{G}} \bullet \underset{\sim}{\boldsymbol{u}}) \times (\underset{\approx}{\boldsymbol{G}} \bullet \underset{\sim}{\boldsymbol{v}})$ to be the “transformed” area vector, then substituting Eq. (9.53) into Eq. (9.46) gives

$$\underset{\sim}{\boldsymbol{A}} = \det(\underset{\approx}{\boldsymbol{G}})\ \underset{\approx}{\boldsymbol{G}}^{-T} \bullet \underset{\sim}{\boldsymbol{A}}_o \tag{9.54}$$

In continuum mechanics, this equation is known as **Nanson’s formula**.

For the record, the result of differentiating $\underset{\approx}{\boldsymbol{G}}^{C}$ with respect to $\underset{\approx}{\boldsymbol{G}}$ is

$$\frac{\partial G^C_{ij}}{\partial G_{rs}} = \varepsilon_{irm} G_{mn} \varepsilon_{njs} \tag{9.55}$$

Axial tensors (tensor associated with a cross-product)

Consider the cross-product, $\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{x}}$. If the vector $\underset{\sim}{\boldsymbol{a}}$ is regarded as fixed, then the cross product may be regarded as a vector-to-vector transformation $f(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{x}}$. This transformation is linear because, for any scalars α and β and any vectors $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$, $f(\alpha\underset{\sim}{\boldsymbol{u}} + \beta\underset{\sim}{\boldsymbol{v}}) = \alpha f(\underset{\sim}{\boldsymbol{u}}) + \beta f(\underset{\sim}{\boldsymbol{v}})$. Consequently, recalling Eq. (9.7), we can assert that there must exist a tensor $\underset{\approx}{\boldsymbol{A}}$ such that $f(\underset{\sim}{\boldsymbol{x}}) = \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{x}}$. In other words, we seek a tensor $\underset{\approx}{\boldsymbol{A}}$ such that

$$\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{x}} = \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{x}} \tag{9.56}$$

Writing this out in component form gives

$$\varepsilon_{ijk} a_j x_k = A_{ik} x_k \tag{9.57}$$

We used the symbol “k” as the dummy summation symbol on the right hand side so that the subscript on x there would be the same as on the left-hand-side. Asserting that this equation must hold for all x_k gives the formula for the tensor:

$$A_{ik} = \varepsilon_{ijk} a_j \tag{9.58}$$

Equivalently,

$$A_{ik} = -\varepsilon_{ikj} a_j \tag{9.59}$$

By moving the summed index to abutting locations, we can write this result in direct (structured) notation as

$$\underset{\approx}{\boldsymbol{A}} \equiv -\underset{\underset{\sim}{\approx}}{\varepsilon} \bullet \underset{\sim}{\boldsymbol{a}} \tag{9.60}$$

Note that the tensor $\underset{\approx}{\boldsymbol{A}}$ depends on $\underset{\sim}{\boldsymbol{a}}$, but is independent of $\underset{\sim}{\boldsymbol{x}}$. We can regard this result itself as a definition of a vector-to-tensor operation that transforms the vector $\underset{\sim}{\boldsymbol{a}}$ into a tensor $\underset{\approx}{\boldsymbol{A}}$. Below, we emphasize that $\underset{\approx}{\boldsymbol{A}}$ is given by an operation on $\underset{\sim}{\boldsymbol{a}}$ by using the symbol $\underset{\approx}{\Omega}^{<a>}$ instead of the symbol $\underset{\approx}{\boldsymbol{A}}$.

Given any vector $\underset{\sim}{\boldsymbol{a}}$, you can always construct a skew-symmetric "**axial tensor**" defined by

$$\underset{\approx}{\Omega}^{<a>} \equiv -\underset{\underset{\sim}{\approx}}{\varepsilon} \bullet \underset{\sim}{\boldsymbol{a}} \tag{9.61}$$

The components of the axial tensor are given by

$$\Omega_{ij}^{<a>} = -\varepsilon_{ijk} a_k \tag{9.62}$$

Equivalently,

$$\Omega_{ij}^{<a>} = \varepsilon_{ikj} a_k \tag{9.63}$$

Expanding out the implied summation over the index k (noting that all terms for which $i=j$ are zero), the matrix for the axial tensor is related to the components of the vector $\underset{\sim}{\boldsymbol{a}}$ according to

$$[\underset{\approx}{\Omega}^{<a>}] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{9.64}$$

For any vector $\underset{\sim}{\boldsymbol{x}}$, note that

$$\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_2 x_3 - a_3 x_2 \\ a_3 x_1 - a_1 x_3 \\ a_1 x_2 - a_2 x_1 \end{bmatrix} \tag{9.65}$$

The components on the right-hand-side of this equation are components of the cross product $\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{x}}$. Stated in direct notation,

$$\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{x}} = \underset{\approx}{\Omega}^{<a>} \bullet \underset{\sim}{\boldsymbol{x}} \tag{9.66}$$

Similarly,

$$\underset{\sim}{\boldsymbol{x}} \times \underset{\sim}{\boldsymbol{a}} = \underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\Omega}^{<a>} \tag{9.67}$$

These results show that axial tensors can be used to convert a cross product into a dot product between a tensor and a vector. The axial tensor associated with a vector can be regarded as simply an alternative way to represent a vector, and this formulation is convenient in any expression involving cross products.

Glide plane expressions. The mathematical description of any problem that deals with material shearing on a plane frequently ends up involving dyadic expressions of the form $\underset{\sim}{\boldsymbol{b}}\underset{\sim}{\boldsymbol{c}}-\underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{b}}$. We have not yet defined what a dyad or a dyadic means. For now, regard this particular dyadic as a special skew symmetric tensor (the alternating tensor), and its associated axial vector is the cross product between $\underset{\sim}{\boldsymbol{b}}$ and $\underset{\sim}{\boldsymbol{c}}$. That is,

$$\underset{\approx}{\xi}\bullet(\underset{\sim}{\boldsymbol{b}}\times\underset{\sim}{\boldsymbol{c}}) = \underset{\sim}{\boldsymbol{b}}\underset{\sim}{\boldsymbol{c}}-\underset{\sim}{\boldsymbol{c}}\underset{\sim}{\boldsymbol{b}}, \tag{9.68}$$

where, for a right-handed orthonormal basis, $\xi_{ijk} = \varepsilon_{ijk}$ (the permutation symbol). We placed this result here because of its intimate connection with cross products. Expressions like this show up frequently in single crystal plasticity theory.

Axial vectors

Given any tensor $\underset{\approx}{\boldsymbol{B}}$, the associated axial *vector* can be constructed by

$$\underset{\sim}{\omega}^{<B>} = -\frac{1}{2}\underset{\approx}{\varepsilon}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}} \tag{9.69}$$

In component form, this notation means

$$\omega_k^{<B>} = -\frac{1}{2}\varepsilon_{kij}B_{ij} \tag{9.70}$$

Expanding out the summations over i and j shows that

$$\omega_1^{<B>} = \frac{1}{2}(B_{32}-B_{23}) \tag{9.71}$$

$$\omega_2^{<B>} = \frac{1}{2}(B_{13}-B_{31}) \tag{9.72}$$

$$\omega_3^{<B>} = \frac{1}{2}(B_{21}-B_{12}) \tag{9.73}$$

Note that

$$\underset{\sim}{\omega}^{<B>} = \underset{\sim}{\omega}^{<\mathrm{skw}B>} \tag{9.74}$$

In other words, the axial vector is determined entirely from the skew-symmetric part of a tensor. The symmetric part has no influence.

Cofactor tensor associated with a *vector*

The cofactor operation is really meant to be applied to an even-order tensor (second-order, fourth-order, etc.). Consider, however, the most general form for a *skew-symmetric tensor* $\underset{\approx}{\mathbf{W}}$. To be skew-symmetric, this tensor must satisfy

$$W_{ji} = -W_{ij}, \tag{9.75}$$

from which it follows that $W_{11} = W_{22} = W_{33}$, $W_{12} = -W_{21}$, $W_{23} = -W_{32}$, and $W_{31} = -W_{13}$. As will be discussed later, it is common practice to associate a *vector* $\underset{\sim}{\mathbf{w}}$ with a skew-symmetric tensor, defined such that $w_1 = -W_{23}$, $w_2 = -W_{31}$, and $w_3 = -W_{12}$. In terms of this vector, the most general form for the 3×3 matrix of a skew-symmetric tensor is

$$[\underset{\approx}{\mathbf{W}}] = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \tag{9.76}$$

Taking the cofactor of this tensor gives

$$[\underset{\approx}{\mathbf{W}}^C] = \begin{bmatrix} w_1^2 & w_1 w_2 & w_1 w_3 \\ w_2 w_1 & w_2^2 & w_2 w_3 \\ w_3 w_1 & w_3 w_2 & w_3^2 \end{bmatrix} \tag{9.77}$$

Referring to Eq. (7.38), we recognize that this matrix is simply the matrix associated with the *dyad* $\underset{\sim}{\mathbf{w}}\underset{\sim}{\mathbf{w}}$. Therefore

If $\underset{\sim}{\mathbf{w}}$ is the axial vector associated with a skew-symmetric tensor $\underset{\approx}{\mathbf{W}}$, then

$$\underset{\approx}{\mathbf{W}}^C = \underset{\sim}{\mathbf{w}}\underset{\sim}{\mathbf{w}} \tag{9.78}$$

In many computational analysis codes, skew-symmetric tensors are not saved as 3×3 matrices (doing so would be a waste of valuable memory). Instead, when a subroutine requests a skew-symmetric tensor from the host code, it will instead be given the three components of the axial vector $\underset{\sim}{\mathbf{w}}$. If that routine needs to compute the cofactor of the full skew-symmetric tensor $\underset{\approx}{\mathbf{W}}$, then it must recognize that all it needs to do is construct the dyad $\underset{\sim}{\mathbf{w}}\underset{\sim}{\mathbf{w}}$.

Cramer’s rule for the inverse

The inverse may be expressed in terms of the cofactor tensor as

$$\underset{\approx}{\mathbf{G}}^{-1} = \frac{\underset{\approx}{\mathbf{G}}^{CT}}{\det(\underset{\approx}{\mathbf{G}})} \tag{9.79}$$

Cramer’s rule is very inefficient for computing the inverse of a large matrix, but it is perfectly adequate for the 3×3 matrices associated with second-order tensors in 3D space.

Inverse of a rank-1 modification (Sherman-Morrison formula)

In general, there is no direct notation formula for the inverse of the sum of two matrices. But there *is* a simple formula for any rank-1 modification of a tensor. Specifically, if

$$\underset{\approx}{\mathbf{B}} = \underset{\approx}{\mathbf{A}} + \underset{\sim}{\mathbf{v}}\underset{\sim}{\mathbf{w}}, \qquad \text{(or, in indicial form, } B_{ij} = A_{ij} + v_i w_j\text{)} \tag{9.80}$$

then (as readily verified by back substitution)

$$\underset{\approx}{\mathbf{B}}^{-1} = \underset{\approx}{\mathbf{A}}^{-1} - \frac{\underset{\approx}{\mathbf{A}}^{-1} \bullet \underset{\sim}{\mathbf{v}}\underset{\sim}{\mathbf{w}} \bullet \underset{\approx}{\mathbf{A}}^{-1}}{1 + \underset{\sim}{\mathbf{w}} \bullet \underset{\approx}{\mathbf{A}}^{-1} \bullet \underset{\sim}{\mathbf{v}}} \qquad \left(B^{-1}_{ij} = A^{-1}_{ij} - \frac{A^{-1}_{im} v_m w_n A^{-1}_{nj}}{1 - w_p A^{-1}_{pq} v_q} \right) \tag{9.81}$$

One disadvantage of the Sherman-Morrison formula as cited above is that it presumes that the tensor $\underset{\approx}{\mathbf{A}}$ is invertible. However, $\underset{\approx}{\mathbf{A}}$ does not necessarily need to be invertible itself in order for the expression $\underset{\approx}{\mathbf{A}} + \underset{\sim}{\mathbf{v}}\underset{\sim}{\mathbf{w}}$ to be invertible. For example, it might be that $[\underset{\approx}{\mathbf{A}}] = \mathrm{DIAG}[0, 1, 1]$, which is not invertible. But if $\{\underset{\sim}{\mathbf{v}}\} = \{\underset{\sim}{\mathbf{w}}\} = \{1, 0, 0\}$, then $[\underset{\approx}{\mathbf{A}} + \underset{\sim}{\mathbf{v}}\underset{\sim}{\mathbf{w}}]$ will equal $\mathrm{DIAG}[1, 1, 1]$, which *is* invertible. To generalize the Sherman-Morrison formula to be able to handle cases like this, what we really need are separate formulas for the determinant and cofactor of $\underset{\approx}{\mathbf{A}} + \underset{\sim}{\mathbf{v}}\underset{\sim}{\mathbf{w}}$. These formulas are

$$\det[\underset{\approx}{\mathbf{A}} + \underset{\sim}{\mathbf{v}}\underset{\sim}{\mathbf{w}}] = \det(\underset{\approx}{\mathbf{A}}) + \underset{\approx}{\mathbf{A}}^{C} \mathbf{:} \underset{\sim}{\mathbf{v}}\underset{\sim}{\mathbf{w}} \tag{9.82}$$

and

$$\underset{\approx}{\mathbf{B}}^{C} = \underset{\approx}{\mathbf{A}}^{C} - \underset{\sim}{\mathbf{v}} \times \underset{\approx}{\mathbf{A}} \times \underset{\sim}{\mathbf{w}} \tag{9.83}$$

Here $\underset{\sim}{\mathbf{v}} \times \underset{\approx}{\mathbf{A}} \times \underset{\sim}{\mathbf{w}}$ is a second-order tensor with ij components $\varepsilon_{imn}\varepsilon_{jrs} v_m A_{nr} w_s$. Rather than using this formula directly, it is easiest to evaluate this tensor by

$$\underset{\sim}{\mathbf{v}} \times \underset{\approx}{\mathbf{A}} \times \underset{\sim}{\mathbf{w}} = \underset{\approx}{\Omega}^{<v>} \bullet \underset{\approx}{\mathbf{A}} \bullet \underset{\approx}{\Omega}^{<w>}, \tag{9.84}$$

where $\underset{\approx}{\Omega}^{<v>}$ and $\underset{\approx}{\Omega}^{<w>}$ are the axial *tensors* associated with the vectors $\underset{\sim}{\mathbf{v}}$ and $\underset{\sim}{\mathbf{w}}$ respectively (see page ___).

Derivative of a determinant

Let J denote the determinant of a tensor $\underset{\approx}{\mathbf{F}}$:

$$J \equiv \det \underset{\approx}{\mathbf{F}} \tag{9.85}$$

Recall that the determinant of a *tensor* can be obtained by simply taking the determinant of the matrix of components associated with that tensor:

$$J \equiv \det \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \tag{9.86}$$

From this we see that $J = J(F_{11}, \ldots, F_{33})$. In other words, J is a function of the nine components of $\underset{\approx}{\mathbf{F}}$. Now we seek to know the partial derivative of J with respect to any one of the components F_{ij} holding the other components fixed. Recall from simple matrix analysis that the determinant in Eq. (9.86) can be computed using cofactors:

$$J = \sum_{j=1}^{3} F_{ij} F_{ij}^C \quad \text{(no implied sum on } i\text{)} \tag{9.87}$$

Now recall that the cofactor F_{ij}^C is the signed determinate of the submatrix obtained by striking out the i^{th} row and j^{th} column of $[F]$. Consequently, this striking out of rows and columns at the ij position implies that F_{ij}^C does not depend on the value of F_{ij}, and the derivative of Eq. (9.87) gives simply

$$\frac{\partial J}{\partial F_{ij}} = F_{ij}^C \tag{9.88}$$

In direct notation, this result is written as, $\boxed{\dfrac{dJ}{d\underset{\approx}{\mathbf{F}}} = \underset{\approx}{\mathbf{F}}^C}$ or, if $\underset{\approx}{\mathbf{F}}$ is invertible,

$$\boxed{\frac{dJ}{d\underset{\approx}{\mathbf{F}}} = J\underset{\approx}{\mathbf{F}}^{-T}} \tag{9.89}$$

Exploiting operator invariance with "preferred" bases

GOAL:

An important technique in tensor analysis uses invariance of tensor operations to justify performing proofs in the context of a cleverly chosen or otherwise convenient basis. Whenever working with a symmetric matrix, for example, a particularly convenient basis is the principal basis for which the component matrix becomes diagonal. If $\underset{\approx}{\mathbf{A}}$ is symmetric, then

$$[\underset{\approx}{\mathbf{A}}] = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix} \text{ with respect to the principal basis for } \underset{\approx}{\mathbf{A}} \tag{9.90}$$

Other tensors will not generate diagonal matrices with respect to the principal basis for $\underset{\approx}{\mathbf{A}}$. If, however, the matrix for a different tensor happens to be diagonal with respect to the principal basis for $\underset{\approx}{\mathbf{A}}$, then you can immediately assert that the eigenvectors for that matrix coincide with those for $\underset{\approx}{\mathbf{A}}$. Knowing that composition (i.e. the dot product) between two tensors corresponds to matrix multiplication and knowing that composition is an invariant operation, you can perform matrix multiplications within any convenient basis, and then recast the result back to invariant (structured/direct/Gibbs) notation when you are done, and, *as long as all of the operations that you performed in the matrix context corresponded to tensor invariant operations*, you will be justified that the final structure notation result will be correct when cast in terms of some different basis.

A sometimes useful generalization of Eq. (9.90) that applies to a *non-symmetric* tensor $\underset{\approx}{\mathbf{F}}$ follows from our knowledge that any tensor can be decomposed uniquely into the sum of a symmetric tensor $\underset{\approx}{\mathbf{A}} = \frac{1}{2}(\underset{\approx}{\mathbf{F}} + \underset{\approx}{\mathbf{F}}^T)$ plus a skew-symmetric tensor $\underset{\approx}{\mathbf{B}} = \frac{1}{2}(\underset{\approx}{\mathbf{F}} - \underset{\approx}{\mathbf{F}}^T)$.

$$\underset{\approx}{\mathbf{F}} = \underset{\approx}{\mathbf{A}} + \underset{\approx}{\mathbf{B}} \tag{9.91}$$

Being symmetric, we know there exists a basis (the principal basis for $\underset{\approx}{\mathbf{A}}$) such that the matrix for $\underset{\approx}{\mathbf{A}}$ is diagonal as in Eq. (9.90). We also know that any skew-symmetric basis will have a matrix expressible in the form of Eq. (9.76) *for any basis*. Thus, knowing that tensor addition, which is an invariant operation, can be cast as matrix addition, we can assert that there will always exist an orthonormal basis (namely the principal basis for $\underset{\approx}{\mathbf{A}}$) such that its components for *any* generally non-symmetric tensor $\underset{\approx}{\mathbf{F}}$ will be of the form

$$[\underset{\approx}{\mathbf{F}}] = \begin{bmatrix} a_1 & -b_3 & b_2 \\ b_3 & a_2 & -b_1 \\ -b_2 & b_1 & a_3 \end{bmatrix} \text{ with respect to the principal basis for sym}\underset{\approx}{\mathbf{F}} \tag{9.92}$$

Here, the a_k are the eigenvalues of sym$\underset{\approx}{\mathbf{F}}$ and the b_k are the components of the axial vector for skw$\underset{\approx}{\mathbf{F}}$ with respect to this principal basis.

By using this special matrix representation, some proofs become much easier. Suppose, for example, that we wish to find a formula for the invariants of $\underset{\approx}{\mathbf{F}}$ expressed in terms of scalar invariants of $\underset{\approx}{\mathbf{A}} = \text{sym}\underset{\approx}{\mathbf{F}}$ and $\underset{\approx}{\mathbf{B}} = \text{skw}\underset{\approx}{\mathbf{F}}$. We know that scalar invariants can be computed by taking the *matrix* invariants of the RCS matrix for $\underset{\approx}{\mathbf{F}}$. Being invariant, we know that the result we obtain using one basis will be identical to the result obtained using any other basis, so why not use the *convenient* basis in Eq. (9.92). Taking the trace of Eq. (9.92) gives

$$\begin{aligned} I_2(\underset{\approx}{\mathbf{F}}) &= a_2a_3 + b_1^2 + a_3a_1 + b_2^2 + a_1a_2 + b_3^2 \\ &= a_2a_3 + a_3a_1 + a_1a_2 + b_1^2 + b_2^2 + b_3^2 \end{aligned} \tag{9.93}$$

Recognizing the first three terms to be the second invariant of $\underset{\approx}{\mathbf{A}}$ and recognizing the last three terms to be the square magnitude of the axial vector associated with $\underset{\approx}{\mathbf{B}}$, and knowing that both the second invariant and vector magnitudes are invariants, we can now cast this result in a structured direct notation that applies to *any basis* (even though we derived in a special basis):

$$I_2(\underset{\approx}{\mathbf{F}}) = I_2(\underset{\approx}{\mathbf{A}}) + \underset{\sim}{\mathbf{b}} \bullet \underset{\sim}{\mathbf{b}} \tag{9.94}$$

Preferred bases are not limited to principal bases. You might, for example, want to set up a basis that is aligned with a structural component of interest. Your vectors and tensors might not have particularly simplified components in this basis, but other reasons might make the choice nonetheless convenient.

> *"Human salvation lies in the hands of the creatively maladjusted."*
> *— Martin Luther King, Jr.*

10. Projectors in tensor notation

Nonlinear projections do not have a tensor representation

A projection transformation $P(\underset{\sim}{x})$ (regardless of whether it is linear or nonlinear) will have a range space and a family of level sets. Given a vector $\underset{\sim}{y}$ in the range space, the associated **level set** is the set of all vectors $\underset{\sim}{x}$ for which $P(\underset{\sim}{x}) = \underset{\sim}{y}$. In other words, the level set is the collection of vectors that will all project to the exact same result. Level sets associated with two *different* $\underset{\sim}{y}$ vectors will not intersect.

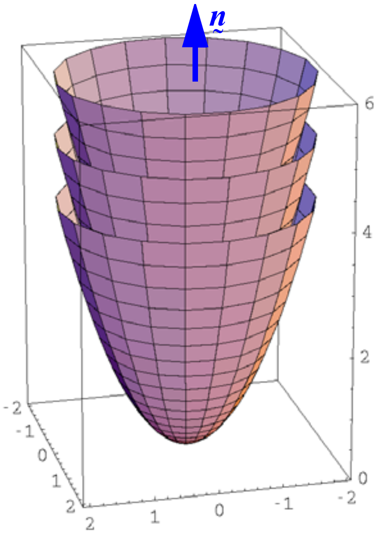
A rank-m projection is one whose range space is m-dimensional, and, for projections in general N-dimensional space, the level sets will then be $N-m$ dimensional. Nonlinear projections will have a curvilinear range space and/or at least one curvilinear level set.

Consider, for example, the following nonlinear projection:

$$P(\underset{\sim}{x}) = [(\underset{\sim}{x} \bullet \underset{\sim}{n}) + (\underset{\sim}{x} \bullet \underset{\sim}{n})^2 - \underset{\sim}{x} \bullet \underset{\sim}{x}]\underset{\sim}{n} \qquad (10.1)$$

where $\underset{\sim}{n}$ is a specified (known) unit vector. The expression in the brackets is a scalar, so this equation says that $P(\underset{\sim}{x})$ is a scalar times the vector $\underset{\sim}{n}$.* Thus, the range space is a straight line in space; the line passes through the origin because $P(\underset{\sim}{0}) = \underset{\sim}{0}$. Knowing that the zero vector happens to belong to the range space, what is the level set associated with the zero vector? The answer is the set of all $\underset{\sim}{x}$ for which $P(\underset{\sim}{x}) = \underset{\sim}{0}$, or the set of all $\underset{\sim}{x}$ for which

$$\underset{\sim}{x} \bullet \underset{\sim}{n} = \underset{\sim}{x} \bullet \underset{\sim}{x} - (\underset{\sim}{x} \bullet \underset{\sim}{n})^2 \qquad (10.2)$$



To get a better handle on the geometrical description of this level set, let's set up a coordinate system for which $\underset{\sim}{e}_3 = \underset{\sim}{n}$. Then the component form for Eq. (10.2) becomes

* Knowing that $P(\underset{\sim}{x})$ will be of the form $\alpha\underset{\sim}{n}$ for some scalar α, you can readily verify that P is indeed a projector because $P(\alpha\underset{\sim}{n}) = \alpha\underset{\sim}{n}$, or $P(P(\underset{\sim}{x})) = P(\underset{\sim}{x})$.

$$x_3 = x_1^2 + x_2^2 \qquad (10.3)$$

This is the equation for a paraboloid centered about $\underset{\sim}{\boldsymbol{n}}$. The level set associated with a different member of the range space, $\alpha\underset{\sim}{\boldsymbol{n}}$, will be a paraboloid of the same size and shape, but passing through $\alpha\underset{\sim}{\boldsymbol{n}}$. The inset figure shows the level sets for the vectors $\underset{\sim}{\boldsymbol{0}}$, $\underset{\sim}{\boldsymbol{n}}$, and $2\underset{\sim}{\boldsymbol{n}}$ in the range space. All points on a given paraboloid will project to the same point in the range space.

For this example, the range space is 1-dimensional (the line parallel to $\underset{\sim}{\boldsymbol{n}}$) and the level sets are each 2-dimensional (paraboloids). The sum of these dimensions equals the dimension of space (three). For this example, the range space was linear, but the level sets were curvilinear.

The signum function is a different projection operator for which the range space is two-dimensional (the curvilinear surface of a unit sphere) and the level sets are 1-dimensional (lines emanating radially from the origin). For projections in three dimensions, the dimensions of the range and level set spaces will always sum to three. For nonlinear projections, either the range and/or the level set space will be curvilinear.

Recall that tensors characterize only *linear* operators. Consequently, a nonlinear projection $P(\underset{\sim}{\boldsymbol{x}})$ cannot be fully characterized by a tensor that is independent of $\underset{\sim}{\boldsymbol{x}}$.

Linear orthogonal projectors expressed in terms of dyads

For *any* linear vector-to-vector function $f(\underset{\sim}{\boldsymbol{x}})$, recall that linearity guarantees that $f(\underset{\sim}{\boldsymbol{0}}) = \underset{\sim}{\boldsymbol{0}}$. Hence, the zero vector *must* be a member of the range space for *any* linear function. Moreover, the range space for *any* linear operator must be a linear manifold. A linear manifold is a set of vectors for which any linear combination of vectors in that manifold will result in a vector that is *also* in the manifold. The surface of the unit sphere (i.e., the set of all unit vectors) is an example of a 2D space that is *not* a linear manifold because a linear combination of unit vectors is not itself generally a unit vector. In ordinary (Newtonian) 3D space, a linear manifold is typically a line or a plane *that passes through the origin*. The entirety of this 3D space is also be a (degenerate) linear manifold because any linear combination of 3D vectors is itself a 3D vector. The zero vector is a degenerate zero-dimensional linear manifold because any linear combination of zero vectors is itself the zero vector. For *any* linear function $f(\underset{\sim}{\boldsymbol{x}})$, recall that the Representation Theorem of Eq. (9.7) guarantees the existence of a tensor $\underset{\approx}{\boldsymbol{F}}$ that is independent of $\underset{\sim}{\boldsymbol{x}}$ for which $f(\underset{\sim}{\boldsymbol{x}}) = \underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{x}}$. Our goal now is to apply this fact to the special case of linear *projection* operators.

Linear projections will always have *both* linear range spaces *and* affine level sets. In three dimensions, an affine space is simply line or a plane that possibly does *not* pass through the origin. The range space for a rank-1 linear projection will be a 1D straight line passing through the origin and the level sets will be families of parallel 2D planes that intersect the line. The range space for a rank-2 linear projection will be a plane passing through the origin and the level sets will be a family of parallel straight lines that intersect the plane (think of these level sets as the “light rays” that cast the shadow of a vector onto the plane). For oblique linear projections, the range space will *not* be perpendicular to the level sets. The level set dimension will be complementary to that of the range space (if the range is a line, the level set will be a plane, and vice versa). Except for degenerate projections (the zero or identity operator), the level sets will be families of parallel lines or parallel planes; of these, exactly one will pass through the origin and will therefore be a linear manifold. This special level set will be called the **null space** for the projection operator. Because the null space is the level set corresponding to the zero vector, it must be the set of all vectors $\underset{\sim}{\boldsymbol{x}}$ for which $P(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{0}}$.

A linear projection function must, according to Eq. (9.7) submit to a tensorial representation. The tensor will be non-symmetric for oblique projections and symmetric for orthogonal (nearest point) projections. In Fig. 6.3, the range space is the plane whose normal is $\underset{\sim}{\boldsymbol{b}}$ and the projection direction is in the direction of $\underset{\sim}{\boldsymbol{a}}$. For an *orthogonal* projection, $\underset{\sim}{\boldsymbol{a}}$ is proportional to $\underset{\sim}{\boldsymbol{b}}$ and, referring to Eq. (6.12), the projection transformation can be written

$$Q(\underset{\sim}{\boldsymbol{x}}) = \underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{n}}(\underset{\sim}{\boldsymbol{n}} \bullet \underset{\sim}{\boldsymbol{x}}) \tag{10.4}$$

Applying the definition of a dyad, this can be written

$$Q(\underset{\sim}{\boldsymbol{x}}) = \underset{\approx}{\boldsymbol{Q}} \bullet \underset{\sim}{\boldsymbol{x}} \tag{10.5}$$

where

$$\underset{\approx}{\boldsymbol{Q}} = \underset{\approx}{\boldsymbol{I}} - \underset{\sim}{\boldsymbol{n}}\underset{\sim}{\boldsymbol{n}} \tag{10.6}$$

This is the projection *tensor* (or **projector**) associated with the linear projection *operator* in Eq. (10.4). The operator $Q(\underset{\sim}{\boldsymbol{x}})$ is a rank-2 projection because its range space is two dimensional, and we show below that the *matrix* rank of $[\underset{\approx}{\boldsymbol{Q}}]$ is also two. The complementary rank-1 orthogonal null projection of Eq. (6.65) can now be written

$$P(\underset{\sim}{\boldsymbol{x}}) = \underset{\approx}{\boldsymbol{P}} \bullet \underset{\sim}{\boldsymbol{x}}, \tag{10.7}$$

where

$$\underset{\approx}{\boldsymbol{P}} = \underset{\approx}{\boldsymbol{I}} - \underset{\approx}{\boldsymbol{Q}} = \underset{\sim}{\boldsymbol{n}}\underset{\sim}{\boldsymbol{n}} \tag{10.8}$$

Being a rank-1 projection, the matrix rank of $[\underset{\approx}{\boldsymbol{P}}]$ is also 1. If we set up a basis for which $\underset{\sim}{\boldsymbol{e}}_3 = \underset{\sim}{\boldsymbol{n}}$, then [recalling Eq. (9.24)],

$$\underset{\approx}{\boldsymbol{Q}} = \underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1 + \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2 \qquad \Leftrightarrow \qquad [\underset{\approx}{\boldsymbol{Q}}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{w.r.t. } \underset{\sim}{\boldsymbol{e}}_3{=}\underset{\sim}{\boldsymbol{n}} \tag{10.9}$$

$$\underset{\approx}{\boldsymbol{P}} = \underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_3 \qquad \Leftrightarrow \qquad [\underset{\approx}{\boldsymbol{P}}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{w.r.t. } \underset{\sim}{\boldsymbol{e}}_3{=}\underset{\sim}{\boldsymbol{n}} \tag{10.10}$$

When expressed in the preferred (aligned) basis, each projector is like “a piece of the identity tensor.” This makes a lot of sense. If a vector $\underset{\sim}{\boldsymbol{x}}$ is already in the range space for a projector $\underset{\approx}{\boldsymbol{P}}$, then $\underset{\approx}{\boldsymbol{P}} \bullet \underset{\sim}{\boldsymbol{x}} = \underset{\sim}{\boldsymbol{x}}$, so the projector is the identity operator within the range space. If, on the other hand, a vector $\underset{\sim}{\boldsymbol{x}}$ is in the null space for a projector $\underset{\approx}{\boldsymbol{P}}$, then $\underset{\approx}{\boldsymbol{P}} \bullet \underset{\sim}{\boldsymbol{x}} = \underset{\sim}{\boldsymbol{0}}$.

Just one esoteric application of projectors

GOAL: Give an illustration of a physical problem whose governing equations are improved through the use of projectors.

Suppose, that it is known that most of the vectors and tensors of interest in a physical problem have some intimate symmetry connection to a plane of normal $\underset{\sim}{\boldsymbol{n}}$. Then it makes sense to set up a basis that is aligned with this plane in order to search for simplifications in the equations governing the phenomenon. This is done, for example, when shock physicists state that the jump in the “shear” components of the deformation gradient tensor must be zero, and they phrase this requirement mathematically by stating that, with respect to a basis having the 3-direction parallel to the shock normal $\underset{\sim}{\boldsymbol{n}}$, the jump in the deformation gradient tensor components must satisfy

$$[[F_{i\alpha}]] = 0\text{, where } i \text{ ranges from 1 to 3, and } \alpha \text{ ranges from 1 to 2} \tag{10.11}$$

Where the double brackets denote the jump (value behind the shock minus value ahead; i.e., $[[(\quad)]] \equiv (\quad)^{\text{behind}} - (\quad)^{\text{ahead}}$). Since the index α is allowed to range only from 1 to 2, the above equation is stating that

$$[[\underset{\approx}{\boldsymbol{F}}]] \bullet \underset{\approx}{\boldsymbol{Q}} = \underset{\approx}{\boldsymbol{0}}\text{, where } \underset{\approx}{\boldsymbol{Q}} \equiv \underset{\approx}{\boldsymbol{I}} - \underset{\sim}{\boldsymbol{n}}\underset{\sim}{\boldsymbol{n}} \tag{10.12}$$

or, simply,

$$[[\underset{\approx}{\boldsymbol{F}}]] = [[\underset{\approx}{\boldsymbol{F}}]] \bullet \underset{\sim}{\boldsymbol{n}}\underset{\sim}{\boldsymbol{n}} \tag{10.13}$$

This direct notation form of the same result is far more useful because (1) it does not require setting up a basis aligned with the shock front, (2) it permits taking time rates that allow the orientation of the shock front itself to changing with time and/or space, and (3) it lends itself nicely to extended, more complicated, analysis of the jump across intersecting shock fronts.

IMPORTANT: Finding a projection to a desired target space

Sometimes the projection function P is not known in advance. Instead, only the desired range space is known. Referring to Eq. (10.9a), note that $\underset{\approx}{\boldsymbol{P}}$ is merely the sum of dyads of an orthonormal basis for the range space. In general, if a range space is known, all one has to do is obtain an orthonormal basis for the range space and sum the *diagonal* basis dyads to obtain the projector. This concept is fairly trivial in ordinary 3D space, but it becomes more subtle and extremely useful in advanced materials modeling, which deal with nine- or six-dimensional vector spaces (spaces of class V_9^n or V_6^n). Keep in mind that

$\underset{\approx}{\boldsymbol{Q}}$ is also a projector. Its associated range space in this example is simply the 3-axis. The above statement also applies to constructing $\underset{\approx}{\boldsymbol{Q}}$. In summary, if you have a collection of vectors $\{\underset{\sim}{\boldsymbol{v}}_1, \underset{\sim}{\boldsymbol{v}}_2, \ldots\}$, and you wish to construct the orthogonal (nearest point, high noon) projector to the space or subspace spanned by them, then you must first use Gram-Schmidt orthogonalization to generate an orthonormal basis $\{\underset{\sim}{\boldsymbol{n}}_k\}$ where k ranges from 1 to the dimension of the space. Then the associated projector is

$$\underset{\approx}{\boldsymbol{P}} = \sum_k \underset{\sim}{\boldsymbol{n}}_k \underset{\sim}{\boldsymbol{n}}_k \tag{10.14}$$

If, for example, you seek the projector to the one-dimensional space spanned by a single vector $\underset{\sim}{\boldsymbol{v}}$, then the projector to the line parallel to $\underset{\sim}{\boldsymbol{v}}$ would be

$$\underset{\approx}{\boldsymbol{P}} = \frac{\underset{\sim}{\boldsymbol{v}} \otimes \underset{\sim}{\boldsymbol{v}}}{\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{v}}}, \tag{10.15}$$

where we have inserted the dyadic multiplication symbol for clarity. If you have two vectors, $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$, that are already orthogonal, but not necessarily normalized, then the projector to the plane containing $\underset{\sim}{\boldsymbol{u}}$ and $\underset{\sim}{\boldsymbol{v}}$ will be

$$\underset{\approx}{\boldsymbol{P}} = \frac{\underset{\sim}{\boldsymbol{u}} \otimes \underset{\sim}{\boldsymbol{u}}}{\underset{\sim}{\boldsymbol{u}} \bullet \underset{\sim}{\boldsymbol{u}}} + \frac{\underset{\sim}{\boldsymbol{v}} \otimes \underset{\sim}{\boldsymbol{v}}}{\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{v}}}, \tag{10.16}$$

As discussed in great length on page 237, a subset of vector space is called a *subspace* if every linear combination of members of the set is itself in that set. For vectors in 3D, subspaces are boring (either lines, planes, or the space itself). However, subspaces in higher dimensions are rich with information. Later on, for example, we will show that engineering second-order tensors are themselves nine-dimensional vectors. The set of all *isotropic* tensors is a subspace because any linear combination of isotropic tensors is itself isotropic. Therefore, there must exist a *fourth-order* projection tensor that will transform any tensor $\underset{\approx}{\boldsymbol{T}}$ into its isotropic part. Later on, we will show that any isotropic tensor (in 3D) must be expressible in the form of a scalar multiplied by the identity tensor $\underset{\approx}{\boldsymbol{I}}$. Therefore, the identity tensor itself forms a basis for the set of all isotropic tensors. In analogy to Eq. (10.15), the associated fourth-order projection operator is given by

$$\underset{\underset{\sim}{\approx}}{\boldsymbol{P}} = \frac{\underset{\approx}{\boldsymbol{I}} \otimes \underset{\approx}{\boldsymbol{I}}}{\underset{\approx}{\boldsymbol{I}} : \underset{\approx}{\boldsymbol{I}}} \tag{10.17}$$

where "**:**" denotes the inner product for second-order tensors, defined later. The $ijkl$ components of this projector turn out to be $\frac{1}{3}\delta_{ij}\delta_{kl}$. The complementary projector is an operator that returns the part of a tensor that is "perpendicular" to its isotropic part — in other words, it gives the deviatoric part.

For every subspace, a corresponding projector can and should be constructed so that the projection theorem can be invoked. The more complicated and higher dimensional the space, the more useful this concept becomes.

Properties of complementary projection tensors

Consider a projection transformation $P(\underset{\sim}{x})$ and its complement $Q(\underset{\sim}{x})$ as defined in Eq. (6.66). If the function P is linear, then so is Q, and they therefore have associated tensors $\underset{\approx}{\Pi}$ and $\underset{\approx}{\Theta}$. For linear projectors, Eqs. (6.67) may be written

$$\underset{\approx}{\Pi} \bullet \underset{\approx}{\Pi} = \underset{\approx}{\Pi} \qquad\qquad \underset{\approx}{\Theta} \bullet \underset{\approx}{\Theta} = \underset{\approx}{\Theta}$$

$$\underset{\approx}{\Pi} \bullet \underset{\approx}{\Theta} = \underset{\approx}{\Theta} \bullet \underset{\approx}{\Pi} = \underset{\approx}{\mathbf{0}}$$

$$\underset{\approx}{\Pi} + \underset{\approx}{\Theta} = \underset{\approx}{\mathbf{I}}$$

$$\underset{\sim}{x}^P \bullet \underset{\sim}{x}^Q = \underset{\sim}{x} \bullet \underset{\approx}{\Pi}^T \bullet \underset{\approx}{\mathbf{Q}} \bullet \underset{\sim}{x} \qquad (10.18)$$

For example, the projection tensor associated with Eq. (6.66) is

$$\underset{\approx}{\Theta} = \underset{\approx}{\mathbf{I}} - \frac{\underset{\sim}{a} \otimes \underset{\sim}{b}}{\underset{\sim}{a} \bullet \underset{\sim}{b}} \qquad (10.19)$$

or, using the “no-symbol” dyadic notation,

$$\underset{\approx}{\Theta} = \underset{\approx}{\mathbf{I}} - \frac{\underset{\sim}{a}\underset{\sim}{b}}{\underset{\sim}{a} \bullet \underset{\sim}{b}} \qquad (10.20)$$

The complementary projector is

$$\underset{\approx}{\Pi} = \frac{\underset{\sim}{a}\underset{\sim}{b}}{\underset{\sim}{a} \bullet \underset{\sim}{b}} \qquad (10.21)$$

The general properties listed in Eq. (10.18) can be readily verified to hold for this particular example.

Self-adjoint (orthogonal) projectors

A projection operator $P(\underset{\sim}{x})$ is “self-adjoint” if and only if $\underset{\sim}{y} \bullet P(\underset{\sim}{x}) = \underset{\sim}{x} \bullet P(\underset{\sim}{y})$ for all vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$. If the projection is linear, this means that its associated tensor $\underset{\approx}{\mathbf{P}}$ must be symmetric:

A projection P is self-adjoint (orthogonal) if and only if $\underset{\approx}{\mathbf{P}}^T = \underset{\approx}{\mathbf{P}}$.

Therefore, $\underset{\approx}{\mathbf{Q}}^T = \underset{\approx}{\mathbf{Q}}$ also holds. (10.22)

For example, we see that the projector in Eq. (10.21) is self adjoint if and only if $\underset{\sim}{a}\underset{\sim}{b} = \underset{\sim}{b}\underset{\sim}{a}$. Recalling Eq. (7.31), this is possible only if $\underset{\sim}{b} = \alpha\underset{\sim}{a}$, in which case the projection reduces to the *orthogonal* projection of Eq. (6.12) in which the unit vector $\underset{\sim}{n}$ is just $\underset{\sim}{a}/\|\underset{\sim}{a}\|$.

A linear self-adjoint projector transforms its argument to the *nearest point* on the range space. This means that the null space is orthogonal to the range space. For ordinary projections in 3D-space, this means that the vector is projected orthogonally, *not obliquely.* When Eq. (10.22) holds, the last property listed in Eq. (10.18) becomes

$$\underset{\sim}{x}^P \bullet \underset{\sim}{x}^Q = \underset{\sim}{0} \tag{10.23}$$

Suppose that two distinct vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$ are decomposed via the projection theorem as

$$\begin{aligned} \underset{\sim}{x} &= \underset{\sim}{x}^P + \underset{\sim}{x}^Q \\ \underset{\sim}{y} &= \underset{\sim}{y}^P + \underset{\sim}{y}^Q \end{aligned} \tag{10.24}$$

Then their inner product is given by

$$\underset{\sim}{x} \bullet \underset{\sim}{y} = \underset{\sim}{x}^P \bullet \underset{\sim}{y}^P + \underset{\sim}{x}^Q \bullet \underset{\sim}{y}^P + \underset{\sim}{x}^P \bullet \underset{\sim}{y}^Q + \underset{\sim}{x}^Q \bullet \underset{\sim}{y}^Q \tag{10.25}$$

If the projection is self-adjoint, then any vector in Q-space must be perpendicular to any vector in P-space. Thus, the middle two inner products in the above equation are both zero, and we obtain

$$\underset{\sim}{x} \bullet \underset{\sim}{y} = \underset{\sim}{x}^P \bullet \underset{\sim}{y}^P + \underset{\sim}{x}^Q \bullet \underset{\sim}{y}^Q \tag{10.26}$$

Equivalently,

$$\underset{\sim}{x} \bullet \underset{\sim}{y} = \underset{\sim}{x} \bullet \underset{\approx}{P} \bullet \underset{\sim}{y} + \underset{\sim}{x} \bullet \underset{\approx}{Q} \bullet \underset{\sim}{y} \tag{10.27}$$

which could have been deduced more quickly by simply dotting the relationship $\underset{\approx}{P} + \underset{\approx}{Q} = \underset{\approx}{I}$ cited in Eq. (10.18) from the left by $\underset{\sim}{x}$ and from the right by $\underset{\sim}{y}$.

Non-self-adjoint (oblique) projectors

GOAL: Call out some differences between orthogonal and oblique projectors

WARNING TO THE READER: I am in the middle of a notation change so that different symbols will be used for the projectors when they are non-self-adjoint. In this section, the tensors $\underset{\approx}{\Pi}$ and $\underset{\approx}{\Theta}$ mean the same thing as $\underset{\approx}{P}$ and $\underset{\approx}{Q}$. This notation change has not yet propagated everywhere in this document, so you will have to decide from context whether or not a projector is self-adjoint.

Even for oblique projections, the relationship $\underset{\approx}{\Pi} + \underset{\approx}{\Theta} = \underset{\approx}{I}$ still holds and therefore

$$\underset{\sim}{x} \bullet \underset{\sim}{y} = \underset{\sim}{x} \bullet \underset{\approx}{\Pi} \bullet \underset{\sim}{y} + \underset{\sim}{x} \bullet \underset{\approx}{\Theta} \bullet \underset{\sim}{y} \tag{10.28}$$

As was the case for self-adjoint projectors, we can still define parts of a vector

$$\underset{\sim}{x}^{(P)} = \underset{\approx}{\Pi} \bullet \underset{\sim}{x} \text{ and } \underset{\sim}{x}^{(Q)} = \underset{\approx}{\Theta} \bullet \underset{\sim}{x} \tag{10.29}$$

For oblique projections, the projectors are *not* symmetric. We can define *dual* projections by

$$\underset{\sim}{x}_{(P)} = \underset{\approx}{\Pi}^T \bullet \underset{\sim}{x} \text{ and } \underset{\sim}{x}_{(Q)} = \underset{\approx}{\Theta}^T \bullet \underset{\sim}{x} \tag{10.30}$$

Similar breakdown of the vector $\underset{\sim}{y}$ can be performed, and it turns out that

$$\underset{\sim}{x} \bullet \underset{\sim}{y} = \underset{\sim}{x}_{(P)} \bullet \underset{\sim}{y}^{(P)} + \underset{\sim}{x}_{(Q)} \bullet \underset{\sim}{y}^{(Q)} \tag{10.31}$$

Note the distinction between this result and Eq. (10.26).

Generalized complementary projectors

Given a projector $\underset{\approx}{\boldsymbol{P}}$, its complement is also a projector defined by $\underset{\approx}{\boldsymbol{Q}} = \underset{\approx}{\boldsymbol{I}} - \underset{\approx}{\boldsymbol{P}}$. Suppose that the range space associated with one of these projectors has a dimension greater than one. Then that projector may always be further decomposed into more projectors, all of which may be made complementary to each other. For second-order tensors referenced to ordinary 3D space, there can be up to three projectors $\underset{\approx}{\boldsymbol{P}}_1$, $\underset{\approx}{\boldsymbol{P}}_2$, and $\underset{\approx}{\boldsymbol{P}}_3$. For second-order tensors referenced to N-dimensional space, there can be a total of M orthogonal projections, where $M \le N$. These generalized complementary projectors must satisfy

$$\underset{\approx}{\boldsymbol{P}}_i \bullet \underset{\approx}{\boldsymbol{P}}_i = \underset{\approx}{\boldsymbol{P}}_i \text{, where } i \text{ is a free index (no summation)}$$

$$\underset{\approx}{\boldsymbol{P}}_i \bullet \underset{\approx}{\boldsymbol{P}}_j = \underset{\approx}{\boldsymbol{0}} \text{ if } i \neq j$$

$$\sum_{k=1}^{M} \underset{\approx}{\boldsymbol{P}}_k = \underset{\approx}{\boldsymbol{I}} \tag{10.32}$$

The dimension of the range space for each projector is given by

$$\dim \underset{\approx}{\boldsymbol{P}}_i = \text{rank}[\underset{\approx}{\boldsymbol{P}}_i] = \text{tr}\underset{\approx}{\boldsymbol{P}}_i \tag{10.33}$$

If all of the projectors are self-adjoint (i.e., if they are all symmetric) then $\dim \underset{\approx}{\boldsymbol{P}}_i$ may alternatively be computed by

$$\dim \underset{\approx}{\boldsymbol{P}}_i = \underset{\approx}{\boldsymbol{P}}_i \boldsymbol{:} \underset{\approx}{\boldsymbol{P}}_i \tag{10.34}$$

where the double dot operation is defined such that, for any tensors $\underset{\approx}{\boldsymbol{A}}$ and $\underset{\approx}{\boldsymbol{B}}$, $\underset{\approx}{\boldsymbol{A}} \boldsymbol{:} \underset{\approx}{\boldsymbol{B}} = A_{ij}B_{ij}$ (see Eq. 3.60). For self-adjoint (symmetric) projectors, the generalization of Eq. (10.27) is

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{y}} = \underset{\sim}{\boldsymbol{x}} \bullet \left(\sum_{k=1}^{M} \underset{\approx}{\boldsymbol{P}}_k \right) \bullet \underset{\sim}{\boldsymbol{y}} \tag{10.35}$$

For example, the projector $\underset{\approx}{\boldsymbol{P}}$ Eq. (10.9) represents a projection to the 12 plane. This projection can be decomposed further into more primitive projectors; namely $\underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1$ and $\underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2$. Hence, if desired, we could define a set of three generalized mutually orthogonal complementary projectors as

$$\underset{\approx}{\boldsymbol{P}}_1 = \underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1 \qquad \Leftrightarrow \qquad [\underset{\approx}{\boldsymbol{P}}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{10.36}$$

$$\underset{\approx}{\boldsymbol{P}}_2 = \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2 \qquad \Leftrightarrow \qquad [\underset{\approx}{\boldsymbol{P}}_2] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{10.37}$$

$$\underset{\approx}{\boldsymbol{P}}_3 = \underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_3 \qquad \Leftrightarrow \qquad [\underset{\approx}{\boldsymbol{P}}_3] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10.38}$$

For this example, Eq. (10.35) becomes simply

$$\begin{aligned} \underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{y}} &= \underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1 \bullet \underset{\sim}{\boldsymbol{y}} + \underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2 \bullet \underset{\sim}{\boldsymbol{y}} + \underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_3 \bullet \underset{\sim}{\boldsymbol{y}} \\ &= x_1 y_1 + x_2 y_2 + x_3 y_3 , \end{aligned} \tag{10.39}$$

which is a comfortingly familiar formula! In the next section, we will define the eigenproblem and show how each distinct eigenvalue has a corresponding *unique* eigenprojector. If a complete set of eigenvectors exist, then they may be combined as sums of dyads to generate a complementary projector set satisfying the properties of Eq. (10.32). If the original tensor is symmetric, then its eigentensors are also symmetric.

> *"If you live to the age of a hundred, you've got it made, because very few people die past the age of a hundred" — George Burns*

11. More Tensor primitives

Tensor properties

GOAL: List the properties that a person should look for in a tensor because numerous useful theorems exist for tensors with certain properties.

In subsequent sections we will categorize tensors according to whether or not they have certain properties. For example, a tensor $\underset{\approx}{\mathbf{F}}$ is said to be **symmetric** if $F_{ij} = F_{ji}$. It is **skew-symmetric** if $F_{ij} = -F_{ji}$. It is **orthogonal** if $F_{ki}F_{kj} = \delta_{ij}$. It is a **rotation** if it is orthogonal and has a positive determinant. It is **invertible** if it has a nonzero determinate. It is a **deformation** if it has a positive determinant. It is a **stretch** if it is symmetric and has all positive eigenvalues. It is a **commuter** with a second tensor $\underset{\approx}{\mathbf{G}}$ if $\underset{\approx}{\mathbf{F}} \bullet \underset{\approx}{\mathbf{G}} = \underset{\approx}{\mathbf{G}} \bullet \underset{\approx}{\mathbf{F}}$. Investigating whether or not a tensor has any of these special properties is an essential activity because many tensors that occur in physical applications fall into these categories and therefore lend themselves to numerous special theorems that may help you in your analyses. These classifications of tensors are so important, in fact, that Gurtin [24] even gave them the following names:

Lin = the set of all tensors
Lin+ = the set of all deformation tensors
Sym = the set of all symmetric tensors
Skw = the set of all skew-symmetric tensors
Psym = the set of all stretch tensors
Orth = the set of all orthogonal tensors
Orth+ = the set of all rotation tensors

Of these, Lin, Sym, and Skw are linear manifolds of dimension 9, 6, and 3, respectively. A linear manifold is a set for which any linear combination of members of that set will itself be in the set. For example, a combination of symmetric tensors will itself be symmetric. Orth is *not* a linear manifold because a linear combination of orthogonal tensors will not be itself orthogonal. Incidentally, the dimensions of Lin, Sym, and Skw presume that the underlying physical space is three-dimensional so that general tensors contain 9 independent components, symmetric tensors have 6 independent components, and skew tensors have 3 independent components. In fact, Sym and Skw are perpendicular subspaces of Lin (by this we mean that any member of Sym is also a member of Lin, any member of Skw is also a member of Lin, and any member of Sym will have a zero inner product with any member of Skw). Linear manifolds can be interpreted geometrically as hyperplanes that pass through the origin. Any linear combination of vectors in a plane through the origin will itself belong to the plane. For tensors, the term *hyperplane* is used to emphasize that the plane is more than three dimensional.

Lin+, Orth, and Orth+ are not linear manifolds, but they are *groups* under multiplication. A set of tensors is a **group under multiplication** if taking the dot product between two members of the set will result in a tensor that is itself a member of the set. For example, the product of two orthogonal tensors will itself be orthogonal. Groups can also be identified with hyper-geometries. However, because they are not linear manifolds, these geometries will be curvilinear spaces (in much the same way that the set of all unit vectors can be described geometrically as being the set of all points on the surface of a unit sphere). In some cases, a curvilinear tensor space can be assigned a dimension, which is determined by the nominal dimension of Lin (nine) minus the number of independent equality constraints that must be satisfied by the components. It can be shown, for example, that Orth+ is three dimensional because (being the set of all rotations) any member of Orth+ can be uniquely defined by specifying an angle of rotation (1 number) and a unit axis of rotation (two independent numbers, with the third component being constrained to generate a unit vector). Geometrically, Orth+ can be identified with a portion of a three-dimensional hypersphere.

Now we are going to proceed with giving more detailed discussions of tensor properties and their associated specialized theorems.

Orthogonal (unitary*) tensors

Consider a linear transformation $f(\underset{\sim}{x})$ in which the transformed vector always has the same length as the original vector. In other words,

$$f(\underset{\sim}{x}) \bullet f(\underset{\sim}{x}) = \underset{\sim}{x} \bullet \underset{\sim}{x} \qquad (11.1)$$

Since the transformation is linear, the Representation Theorem of Eq. (9.7) says there must exist a second-order tensor $\underset{\approx}{Q}$ such that

$$f(\underset{\sim}{x}) = \underset{\approx}{Q} \bullet \underset{\sim}{x}, \qquad (11.2)$$

so Eq. (15.37) becomes

$$(\underset{\approx}{Q} \bullet \underset{\sim}{x}) \bullet (\underset{\approx}{Q} \bullet \underset{\sim}{x}) = \underset{\sim}{x} \bullet \underset{\sim}{x} \qquad (11.3)$$

or

$$\underset{\sim}{x} \bullet (\underset{\approx}{Q}^T \bullet \underset{\approx}{Q} - \underset{\approx}{I}) \bullet \underset{\sim}{x} = 0 \qquad (11.4)$$

[*] Typically, when dealing with *real* tensors, you will want to use the term “orthogonal.” The word “unitary” is normally reserved for tensors or matrices that might have complex components. A matrix $[Q]$ is unitary if $[Q]^{-1} = [Q]^H$, where “H” is the hermitian (transpose of the conjugate).

This must hold for all $\underset{\sim}{x}$ and (because the tensor in parentheses is symmetric), the above equation therefore implies that

$$\underset{\approx}{Q}^T \bullet \underset{\approx}{Q} - \underset{\approx}{I} = \underset{\approx}{0} \tag{11.5}$$

Thus, the transpose of the tensor $\underset{\approx}{Q}$ must equal its inverse. A tensor is said to be **orthogonal** if its inverse is the same thing as its transpose:

$$\boxed{\underset{\approx}{Q}^{-1} = \underset{\approx}{Q}^T} \qquad \Leftrightarrow \qquad \underset{\approx}{Q} \text{ is orthogonal} \tag{11.6}$$

Equivalently,

$$\underset{\approx}{Q}^T \bullet \underset{\approx}{Q} = \underset{\approx}{Q} \bullet \underset{\approx}{Q}^T = \underset{\approx}{I} \tag{11.7}$$

By taking the determinant of this equation, we see that

$$(\det \underset{\approx}{Q})^2 = 1 \tag{11.8}$$

The term "proper" is used if $\det \underset{\approx}{Q} = +1$ and "improper" if $\det \underset{\approx}{Q} = -1$. These terms are unfortunate since they imply there is something unsavory about an orthogonal tensor whose determinant is negative.

A *proper* orthogonal tensor corresponds to a rigid rotation about some axis, defined by a unit vector $\underset{\sim}{a}$, and, if we set up a basis in which the 3-direction is aligned with the axis of rotation, then [7]

$$[\underset{\approx}{Q}] = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ with respect to basis having } \underset{\sim}{e}_3 = \underset{\sim}{a}. \tag{11.9}$$

Here, α is the angle of rotation. This expression may be written

$$[\underset{\approx}{Q}] = \cos\alpha \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \sin\alpha \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{with respect to basis where } \underset{\sim}{e}_3 = \underset{\sim}{a}. \tag{11.10}$$

Now we note that

$$[\underset{\approx}{I} - \underset{\sim}{a}\underset{\sim}{a}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad [\underset{\sim}{a}\underset{\sim}{a}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad [\underset{\approx}{A}] = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{with respect to basis having } \underset{\sim}{e}_3 = \underset{\sim}{a}. \tag{11.11}$$

where the tensor $\underset{\approx}{A}$ is defined by $A_{ij} = -\varepsilon_{ijk} a_k$. Thus, we may write a direct notation expression for any rotation as

$$\underset{\approx}{Q} = \cos\alpha(\underset{\approx}{I} - \underset{\sim}{a}\underset{\sim}{a}) + (\underset{\sim}{a}\underset{\sim}{a}) + \sin\alpha\underset{\approx}{A}, \tag{11.12}$$

which is sometimes called the Euler-Rodrigues formula for a rotation.

Many people *wrongly* claim that an “improper” orthogonal tensor corresponds physically to a *reflection*, but it generally represents a reflection *in combination with a rotation*. In 3D, there are two types of simple reflections: (1) a reflection about the origin, which merely reverses the direction of all vectors without affecting their magnitudes, or (2) a “mirror-like” operation that will transform any vector $\underset{\sim}{x}$ to become its mirror image across some plane defined by a unit normal $\underset{\sim}{n}$. The first type of reflection fully “inverts” space (east becomes west, north becomes south, and up becomes down). The second type of reflection inverts only one direction, leaving the others unchanged (left becomes right, but up does not turn into down). It’s easy enough to guess the tensor that induces the first type of reflection; namely, if every vector “turns around”, then the associated tensor must simply be the negative of the identity tensor, $\underset{\approx}{Q} = -\underset{\approx}{I}$. For the second type of reflection, the part of the vector $\underset{\sim}{x}$ in the direction of the “mirror” unit normal $\underset{\sim}{n}$ is given by $\underset{\sim}{n}\underset{\sim}{n} \bullet \underset{\sim}{x}$; the reflection changes the sign of this part of the vector $\underset{\sim}{x}$ to become $-\underset{\sim}{n}\underset{\sim}{n} \bullet \underset{\sim}{x}$. However, the projection of the vector *onto* the plane of the mirror, $\underset{\sim}{x} - \underset{\sim}{n}\underset{\sim}{n} \bullet \underset{\sim}{x}$ remains unchanged. Thus, the reflection is given by the sum of this unchanged part plus the reversed part to give

$$f(\underset{\sim}{x}) = \underset{\sim}{x} - 2(\underset{\sim}{n}\underset{\sim}{n} \bullet \underset{\sim}{x}) \tag{11.13}$$

The corresponding reflection tensor is

$$\underset{\approx}{Q} = \underset{\approx}{I} - 2\underset{\sim}{n}\underset{\sim}{n} \tag{11.14}$$

or

$$[\underset{\approx}{Q}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \text{ with respect to basis where } \underset{\sim}{e}_3 = \underset{\sim}{n}. \tag{11.15}$$

In 3D, that this “second-type” of reflection can always be regarded as a 180° rotation about $\underset{\sim}{n}$ in combination with a “type-one” reflection about the origin:

$$[\underset{\approx}{Q}] = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{with respect to basis where } \underset{\sim}{e}_3 = \underset{\sim}{n}. \tag{11.16}$$

Note that the last matrix in this equation is identical to the matrix in Eq. (11.9) with the rotation angle α set to $180°$. This is just one special case of a more general statement. Namely, in 3D, if $\underset{\approx}{Q}$ is an improper orthogonal tensor, then $\underset{\approx}{Q} = -\underset{\approx}{R}$, where $\underset{\approx}{R}$ is *proper* orthogonal.

Tensor associated with the cross product

Using the permutation symbol, the component formulas for the cross product in a right-handed orthonormal basis given in Eq. (5.29) can be written compactly as

$$(\underset{\sim}{\mathbf{a}} \times \underset{\sim}{\mathbf{b}})_i = \varepsilon_{ijk} a_j b_k \tag{11.17}$$

Note that the operation, $\underset{\sim}{\mathbf{a}} \times \underset{\sim}{\mathbf{b}}$ is linear with respect to $\underset{\sim}{\mathbf{b}}$. Hence, the Representation Theorem of Eq. (9.7) says there must exist a tensor $\underset{\approx}{\mathbf{A}}$ that is independent of $\underset{\sim}{\mathbf{b}}$ such that

$$\underset{\sim}{\mathbf{a}} \times \underset{\sim}{\mathbf{b}} = \underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{b}} \tag{11.18}$$

The i^{th} component of $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{b}}$ is

$$(\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{b}})_i = A_{ij} b_j \tag{11.19}$$

For Eq. (11.18) to hold, the right hand sides of Eqs. (11.17) and (11.19) must be equal:

$$A_{ij} b_j = \varepsilon_{ijk} a_j b_k \tag{11.20}$$

This relationship must hold for all $\underset{\sim}{\mathbf{b}}$. Before we can eliminate $\underset{\sim}{\mathbf{b}}$, however, we must change the dummy summation subscripts so that the indices on the b's are the same on both sides of the above equation. We have b_j on the left hand side of the equation. Consequently, on the right hand side of the equation, we must change the dummy subscript k to j. Doing that, however, would result in four instances of j, which would violate the summation conventions. Hence, in order to change k to j, we must change j to something else. An easy way to do this is to change k to j and j to k. Hence, Eq. (11.20) is equivalent to

$$A_{ij} b_j = \varepsilon_{ikj} a_k b_j \tag{11.21}$$

Now that both b's have the same index, we may assert that this holds for all b_j and therefore

$$A_{ij} = \varepsilon_{ikj} a_k = -\varepsilon_{ijk} a_k, \tag{11.22}$$

Let's quickly re-derive this result in an alternative manner by using Eq. (9.18). Specifically, we know that A_{in} is the derivative of the i^{th} component of $\underset{\sim}{\mathbf{a}} \times \underset{\sim}{\mathbf{b}}$ with respect to b_n. We arbitrarily selected the index symbol n because it is not already being used as an index anywhere in Eq. (11.17). Applying Eq. (9.18) gives the in component of $\underset{\approx}{\mathbf{A}}$:

$$A_{in} = \frac{\partial(\underset{\sim}{\mathbf{a}} \times \underset{\sim}{\mathbf{b}})_i}{\partial b_n} = \frac{\partial(\varepsilon_{ijk} a_j b_k)}{\partial b_n} = \varepsilon_{ijk} a_j \frac{\partial b_k}{\partial b_n} \tag{11.23}$$

In the last step, we have used the fact that neither ε_{ijk} nor a_j depends on $\underset{\sim}{\mathbf{b}}$. To simplify this result, we apply Eq. (9.18) to write $\partial b_k / \partial b_n = \delta_{kn}$ so that

$$A_{in} = \varepsilon_{ijk} a_j \delta_{kn} \tag{11.24}$$

or, applying Eq. (4.43),

$$A_{in} = \varepsilon_{ijn} a_j \tag{11.25}$$

The index n is a free index, so we could change n to k, thereby demonstrating that Eq. (11.25) equivalent to the earlier result of Eq. (11.22).

Now, we wish to write Eq. (11.22) in direct notation. Noting that $\varepsilon_{ikj} = -\varepsilon_{ijk}$, Eq. (11.22) may be written

$$A_{ij} = -\varepsilon_{ijk}a_k \tag{11.26}$$

In direct notation, this result is written

$$\underset{\approx}{A} = -\underset{\underset{\sim}{\approx}}{\xi} \bullet \underset{\sim}{a} \tag{11.27}$$

In matrix form, the components of $\underset{\approx}{A}$ are

$$[\underset{\approx}{A}] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{11.28}$$

which (incidentally) is equivalent to the tensor $\underset{\approx}{A}$ in Eq. (11.11). Whenever a relationship of the form in Eq. (11.27) holds, we say that $\underset{\sim}{a}$ is the axial vector associated with the skew symmetric tensor $\underset{\approx}{A}$.

By the way, observe that

$$\underset{\sim}{b} \times \underset{\sim}{a} = -(\underset{\sim}{a} \times \underset{\sim}{b}) = -\underset{\approx}{A} \bullet \underset{\sim}{b} = -\underset{\sim}{b} \bullet \underset{\approx}{A}^T \tag{11.29}$$

However, we see from Eq. (11.28) that the tensor $\underset{\approx}{A}$ has the skew-symmetry property that

$$\underset{\approx}{A}^T = -\underset{\approx}{A} \tag{11.30}$$

Therefore, Eq. (11.29) becomes

$$\boxed{\underset{\sim}{b} \times \underset{\sim}{a} = \underset{\sim}{b} \bullet \underset{\approx}{A}} \tag{11.31}$$

Cross-products in left-handed and general bases. The formula in Eq. (11.17) applies only when the basis is orthonormal *and right-handed*. We seek here to present the generalized formula for the cross product that applies to some *different* (not necessarily orthogonal or right-handed) basis $\{\underset{\sim}{g}_1, \underset{\sim}{g}_2, \underset{\sim}{g}_3\}$. When using a non-orthogonal or non-right-handed basis, standard notational conventions alter the summation rules to allow for two kinds of indices: a superscripted "contravariant" index and a subscripted "covariant" index and implied summations must always occur on different levels so that, for example, the basis expansion of the vector $\underset{\sim}{a}$ is written

$$\underset{\sim}{a} = a^i \underset{\sim}{g}_i \tag{11.32}$$

Since the three vectors, $\{\underset{\sim}{g}_1, \underset{\sim}{g}_2, \underset{\sim}{g}_3\}$ form a basis, we know that the a^i coefficients *exist*, and you can refer to Ref. [6] for instructions on how to compute them. That reference also explains that there is no difference between super- and sub-scripted quantities when the basis is orthonormal. Hence, in the following equations, the orthonormal right-handed basis $\{\underset{\sim}{e}_1, \underset{\sim}{e}_2, \underset{\sim}{e}_3\}$ is the same as $\{\underset{\sim}{e}^1, \underset{\sim}{e}^2, \underset{\sim}{e}^3\}$.

A more general version of the cross-product formula is

$$(\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{b}})_i = \xi_{ijk} a^j b^k \tag{11.33}$$

where ξ_{ijk} are components of a third order tensor whose basis expansion with respect to an orthonormal right-handed basis is

$$\underset{\approx}{\xi} = \varepsilon_{ijk} \underset{\sim}{\boldsymbol{e}}^i \underset{\sim}{\boldsymbol{e}}^j \underset{\sim}{\boldsymbol{e}}^k \tag{11.34}$$

We have the basis expansion of $\underset{\approx}{\xi}$ with respect to our familiar and comfortable orthonormal right-handed basis, and we can obtain the components of $\underset{\approx}{\xi}$ with respect to the irregular basis through the following formula

$$\xi_{ijk} = a_i{}^p a_i{}^q a_i{}^r \varepsilon_{pqr},$$
$$\text{where } a_m{}^n = \underset{\sim}{\boldsymbol{g}}_m \bullet \underset{\sim}{\boldsymbol{e}}^n \tag{11.35}$$

Though not immediately obvious, you can show that ξ_{ijk} can be computed by the triple scalar product of the base vectors:

$$\xi_{ijk} = [\underset{\sim}{\boldsymbol{g}}_i, \underset{\sim}{\boldsymbol{g}}_j, \underset{\sim}{\boldsymbol{g}}_k] \tag{11.36}$$

Specializing this formula to an orthonormal *but left-handed* basis gives

$$\xi_{ijk} = -\varepsilon_{ijk} \tag{11.37}$$

For an orthonormal basis, there is no difference between a^i and a_i. Likewise, $b^i = b_i$. Hence, for an orthonormal *left-handed* basis, Eq. (11.33) becomes

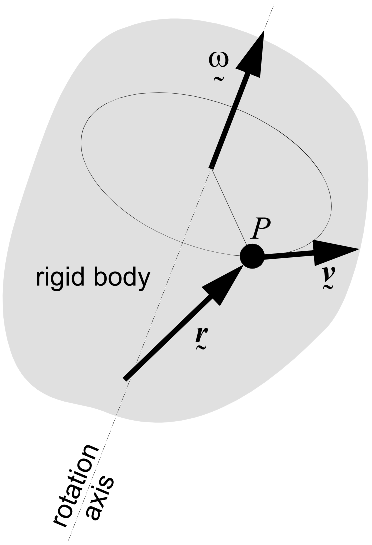
$$(\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{b}})_i = -\varepsilon_{ijk} a_j b_k \tag{11.38}$$

Comparing this with Eq. (11.17), it might *appear* that the cross product changes sign for a left-handed basis, but it doesn't. The vector components in Eq. (11.17) are different from those in Eq. (11.38) by a sign change. The negative in Eq. (11.38) compensates for this sign change so that the final resulting vector,

$$\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{b}} = (\underset{\sim}{\boldsymbol{a}} \times \underset{\sim}{\boldsymbol{b}})_i \underset{\sim}{\boldsymbol{e}}_i \tag{11.39}$$

will be *the same* regardless of the handedness of the basis. The direct notation formula for the axial vector in Eq. (11.27) likewise *does not change* — direct notation formulas apply for *any* basis (how you compute the components varies, but the sum of components times base vectors is invariant). Note that we have denoted the *tensor* $\underset{\approx}{\xi}$ using a symbol that differs from the symbol for its right-handed RCS components, ε_{ijk}. This notational choice is also why we denote the RCS components of the identity tensor $\underset{\approx}{\boldsymbol{I}}$ by δ_{ij} instead of I_{ij}. Unlike the permutation tensor, the identity tensor components remain equal to δ_{ij} even for a left-handed orthonormal basis, but they *are not* δ_{ij} for a non-orthonormal basis.

Physical application of axial vectors



The operation of Eq. (11.27) occurs quite frequently in mechanics. Suppose, for example, that $\underset{\sim}{r}$ is the vector from the origin to a point P in a rigid body. If the rigid body rotates about the origin with an angular velocity vector $\underset{\sim}{\omega}$, then the velocity of the point P is given by

$$\underset{\sim}{v} = \underset{\sim}{\omega} \times \underset{\sim}{r} \qquad (11.40)$$

Equation (11.18) implies that this relationship may be written alternatively in terms of an angular velocity *tensor* $\underset{\approx}{\Omega}$. Namely,

$$\underset{\sim}{v} = \underset{\approx}{\Omega} \bullet \underset{\sim}{r}, \qquad (11.41)$$

where

$$[\underset{\approx}{\Omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \qquad (11.42)$$

The usefulness of $\underset{\approx}{\Omega}$ instead of $\underset{\sim}{\omega}$ becomes apparent by considering rotation. Rotation can be defined in terms of an axis and an angle of rotation, but multiple rotations about different axes do not commute (*i.e.,* if you rotate about one axis and then about the other, the result will not be the same if you switch the order of rotation). Rotation can nevertheless be shown to be a *linear* transformation. In other words, given a collection of vectors, you can rotate them all individually and then take a linear combination of the results or you can *first* take the same linear combination of starting vectors and *then* rotate the result — either way, the final answer will be the same. Hence, the Representation Theorem of Eq. (9.7) says there exists a tensor $\underset{\approx}{R}$ such that rotating an initial vector $\underset{\sim}{r}_o$ to obtain a new vector $\underset{\sim}{r}$ can be written

$$\underset{\sim}{r} = \underset{\approx}{R} \bullet \underset{\sim}{r}_o \qquad (11.43)$$

If the rotation varies in time, we can take time rates of this equation to obtain the velocity, $\underset{\sim}{v} = \dot{\underset{\sim}{r}}$. Specifically, noting that the *initial* vector $\underset{\sim}{r}_o$ is (obviously) not a function of time, the rate of Eq. (11.43) gives

$$\underset{\sim}{v} = \dot{\underset{\approx}{R}} \bullet \underset{\sim}{r}_o \qquad (11.44)$$

Noting from Eq. (11.43) that $\underset{\sim}{r}_o = \underset{\approx}{R}^{-1} \bullet \underset{\sim}{r}$, Eq. (11.44) may be written

$$\underset{\sim}{v} = \underset{\approx}{\Omega} \bullet \underset{\sim}{r}, \text{ where } \underset{\approx}{\Omega} \equiv \dot{\underset{\approx}{R}} \bullet \underset{\approx}{R}^{-1} \qquad (11.45)$$

It can be shown that a rigid rotation tensor must be orthogonal* and hence

$$\underset{\approx}{R}^{-1} = \underset{\approx}{R}^T \qquad (11.46)$$

It can be shown [7] that this property in turn implies that $\underset{\approx}{\Omega}$ must be skew symmetric. The associated axial vector is the angular velocity vector $\underset{\sim}{\omega}$.

Symmetric and skew-symmetric tensors

Note that the components in Eq. (11.28) have the following property:

$$A_{ij} = -A_{ji} \quad \text{or, in direct notation, } \underset{\approx}{A} = -\underset{\approx}{A}^T \tag{11.47}$$

Any tensor with this property is said to be "**skew-symmetric**." Here, the superscript "*T*" denotes the "**transpose**".

A tensor $\underset{\approx}{A}$ is "**symmetric**" if and only if it has the property that

$$A_{ij} = A_{ji} \qquad \text{or, in direct notation,} \qquad \boldsymbol{A} = \boldsymbol{A}^T \tag{11.48}$$

Let $\underset{\approx}{A}$ be any second-order tensor, neither symmetric nor skew symmetric in general. The **symmetric part** of $\underset{\approx}{A}$ is defined

$$\text{sym}\underset{\approx}{A} \equiv \frac{1}{2}(\underset{\approx}{A} + \underset{\approx}{A}^T) \tag{11.49}$$

the **skew-symmetric (or antisymmetric) part** of $\underset{\approx}{A}$ is defined

$$\text{skw}\underset{\approx}{A} \equiv \frac{1}{2}(\underset{\approx}{A} - \underset{\approx}{A}^T) \tag{11.50}$$

With these definitions, we see that any tensor $\underset{\approx}{A}$ can be decomposed additively into symmetric and skew-symmetric parts:

$$\underset{\approx}{A} = \text{sym}\underset{\approx}{A} + \text{skw}\underset{\approx}{A} \tag{11.51}$$

Instead of "sym" and "skw," many authors use superscripts "s" and "a" so that Eq. (11.51) would be written somewhat more compactly as

$$\underset{\approx}{A} = \underset{\approx}{A}^{\text{s}} + \underset{\approx}{A}^{\text{a}} \tag{11.52}$$

Note that the act of taking the symmetric part of a tensor may be itself regarded as a tensor function:

$$P(\underset{\approx}{A}) = \frac{1}{2}(\underset{\approx}{A} + \underset{\approx}{A}^T) \tag{11.53}$$

This operation is a projection because $P(P(\underset{\approx}{A})) = P(\underset{\approx}{A})$. In other words, the symmetric part of a symmetric tensor is just the tensor itself. We can also define a function for taking the skew-symmetric part of a tensor:

$$Q(\underset{\approx}{A}) = \frac{1}{2}(\underset{\approx}{A} - \underset{\approx}{A}^T) \tag{11.54}$$

* For a detailed discussion of orthogonal tensors, including the definition of Euler angles, and how to generate a rotation from an axis and angle, see Ref. [7].

The functions P and Q have the properties listed in Eqs. (6.67) except that, this time, the argument of functions is a tensor rather than a vector. Hence, the fact that any tensor can be split into symmetric and skew-symmetric parts is merely a generalized application of the projection theorem!

Positive definite tensors

A tensor $\underset{\approx}{\boldsymbol{B}}$ is positive definite if and only if

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\boldsymbol{B}} \bullet \underset{\sim}{\boldsymbol{x}} > 0 \text{ for all vectors } \underset{\sim}{\boldsymbol{x}} \qquad (11.55)$$

This equation can be alternatively written as

$$\underset{\approx}{\boldsymbol{B}} \boldsymbol{:} \underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}} > 0 \text{ for all vectors } \underset{\sim}{\boldsymbol{x}}, \qquad (11.56)$$

where "$\boldsymbol{:}$" denotes the tensor inner product, soon to be defined in Eq. (12.1). Noting that the dyad $\underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}$ is a symmetric tensor, we can use the (yet to be presented) result of Eq. (12.26) to note that $\underset{\approx}{\boldsymbol{B}} \boldsymbol{:} \underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}} = (\text{sym}\underset{\approx}{\boldsymbol{B}}) \boldsymbol{:} \underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}$. Therefore, a tensor $\underset{\approx}{\boldsymbol{B}}$ is positive definite if and only if $\text{sym}\underset{\approx}{\boldsymbol{B}}$ is positive definite. The skew-symmetric part of $\underset{\approx}{\boldsymbol{B}}$ has no influence on positive definiteness. Consequently, when assessing definiteness of a tensor, you should always immediately start working with the *symmetric* part of that tensor,

$$\underset{\approx}{\boldsymbol{A}} \equiv \text{sym}\underset{\approx}{\boldsymbol{B}} \qquad (11.57)$$

The tensor $\underset{\approx}{\boldsymbol{A}}$ (and therefore $\underset{\approx}{\boldsymbol{B}}$) is positive definite if and only if its associated component matrix is positive definite. This will happen if and only if all three characteristic invariants* of $\underset{\approx}{\boldsymbol{A}}$ are positive. ***WARNING:*** It is possible to construct a tensor $\underset{\approx}{\boldsymbol{B}}$ that has all positive invariants, but the tensor is *not* positive definite. It is imperative to test the invariants of the *symmetric part* of $\underset{\approx}{\boldsymbol{B}}$ to check for definiteness.

Faster way to check for positive definiteness. Recall from Eqs. (3.130) that the k^{th} characteristic invariant of $\underset{\approx}{\boldsymbol{A}}$ is given by the sum of principal $k \times k$ minors. It turns out that computing all of these principal minors is not necessary in order to test for positive definiteness. All you have to do is test *one* principal minor of each size, making sure that, as you progress in size, the smaller principal minors are nested inside each larger minor. The concept of nesting makes more sense if we explain it starting with the largest principal minor, which is the determinant of the 3×3 matrix itself. For the next principal minor, select any 2×2 submatrix whose diagonal components are also diagonal components of the larger matrix, and verify that the determinant of this 2×2 submatrix is positive. Then test the determinant of any 1×1 submatrix on the diagonal of the 2×2 matrix (i.e., check any diagonal component of that matrix). The following sequence of decreasing sized submatrices are all principal submatrices, but they are *not* nested:

* See Eqs. (3.130) on page 46.

$$\textbf{\textit{Not}}\text{ nested: }\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix},\ \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},\ [A_{33}] \tag{11.58}$$

These are ***not*** nested because the 1×1 submatrix $[A_{33}]$ is *not* a submatrix of the larger 2×2 matrix. Of course, in practice, now that the meaning of "nesting" is clear, it is wiser to check the determinants of the submatrices starting with the smallest one first. If, at any point in this process, you find a negative determinant, then you can stop calculations with the conclusion that the original matrix is *not* positive definite.

Using the simplest nested set of principal minors (those that progress in size starting from the 11 position), the simplest test for positive definiteness of a 3×3 matrix involves merely verifying the following

$$A_{11} > 0 \tag{11.59}$$

$$\det\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} > 0 \tag{11.60}$$

$$\det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} > 0 \tag{11.61}$$

Recalling that you may use any set of nested principal minors, it follows that a positive definite tensor will *always* have positive diagonal components, so this is a good thing to visually check before even looking at the larger determinants. There is nothing, that disallows the *off-diagonals* from being negative, so long as the principal minors always evaluate to a positive number.

Positive *semi*-definite

A tensor $\underset{\approx}{\boldsymbol{B}}$ is **positive *semi*-definite** if and only if

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\boldsymbol{B}} \bullet \underset{\sim}{\boldsymbol{x}} \geq 0 \quad \text{for all vectors } \underset{\sim}{\boldsymbol{x}} \tag{11.62}$$

This means that the invariants (and nested subminors) will all be positive or zero.

Negative definite and negative semi-definite tensors

A tensor $\underset{\approx}{\boldsymbol{C}}$ is **negative definite** if $-\underset{\approx}{\boldsymbol{C}}$ is positive definite. Note that this means that the *odd* invariants I_1 and I_3 must be negative, but the *even* invariant I_2 must be positive. *

* The reason why odd invariants must be negative and even invariants follows from Eq. (3.99) in which the scalar multiple needed to check if $-\underset{\approx}{\boldsymbol{C}}$ is positive definite is $\alpha = -1$, making α^N in Eq. (3.99) equal to -1 if N is odd, but $+1$ if N is even.

from which it follows that its invariants (or principal subminors) will all be simply nonnegative (i.e., positive or zero). A tensor $\underset{\approx}{\boldsymbol{C}}$ is **negative *semi*-definite** if $-\underset{\approx}{\boldsymbol{C}}$ is positive *semi*-definite.

Isotropic and deviatoric tensors

Another operation that looks similar to (but is functionally different from) the sym+skw operation is one that decomposes a tensor into its *deviatoric* and *isotropic* parts. The **deviatoric part** of $\underset{\approx}{\boldsymbol{A}}$ is denoted variously as $\underset{\approx}{\boldsymbol{A}}^{\text{dev}}$ or $\underset{\approx}{\boldsymbol{A}}^{\text{d}}$ or $\underset{\approx}{\boldsymbol{A}}'$ or $\text{dev}\underset{\approx}{\boldsymbol{A}}$, and it is defined

$$\underset{\approx}{\boldsymbol{A}}^{\text{dev}} \equiv \underset{\approx}{\boldsymbol{A}}' \equiv \underset{\approx}{\boldsymbol{A}} - \frac{1}{3}(\text{tr}\underset{\approx}{\boldsymbol{A}})\underset{\approx}{\boldsymbol{I}} \qquad (11.63)$$

Here, $\text{tr}\underset{\approx}{\boldsymbol{A}}$ is a scalar called the "**trace**" of $\underset{\approx}{\boldsymbol{A}}$ defined

$$\text{tr}\underset{\approx}{\boldsymbol{A}} \equiv A_{kk} = A_{11} + A_{22} + A_{33} \qquad (11.64)$$

The **isotropic part** of $\underset{\approx}{\boldsymbol{A}}$ is denoted $\underset{\approx}{\boldsymbol{A}}^{\text{iso}}$ or $\text{iso}\underset{\approx}{\boldsymbol{A}}$, and is defined

$$\underset{\approx}{\boldsymbol{A}}^{\text{iso}} \equiv \frac{1}{3}(\text{tr}\underset{\approx}{\boldsymbol{A}})\underset{\approx}{\boldsymbol{I}} \qquad (11.65)$$

Thus, just as a tensor can be decomposed into symmetric and skew-symmetric parts, it can also be decomposed into deviatoric and isotropic parts:

$$\underset{\approx}{\boldsymbol{A}} = \underset{\approx}{\boldsymbol{A}}^{\text{dev}} + \underset{\approx}{\boldsymbol{A}}^{\text{iso}} \qquad (11.66)$$

This is also an application of the projection theorem! This equation is analogous to Eq. (11.52).

> *"Nothing pains some people more than having to think."* — *Martin Luther King, Jr.*

12. Tensor operations

Second-order tensor inner product

The **inner product** (also called the **double dot product**) between two second-order tensors, $\underset{\approx}{\boldsymbol{A}}$ and $\underset{\approx}{\boldsymbol{B}}$, is a scalar denoted $\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}}$ and defined

$$\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}} \equiv A_{mn}B_{mn} \tag{12.1}$$

Alternatively, this binary operation can be written

$$\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}} = \mathrm{tr}(\underset{\approx}{\boldsymbol{A}}^T \bullet \underset{\approx}{\boldsymbol{B}}) = \mathrm{tr}(\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}^T) \tag{12.2}$$

Beware: the notation $\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}}$ is not universally defined the same way by all people. Some people define the "$\boldsymbol{:}$" operator to mean $A_{mn}B_{nm}$ instead of $A_{mn}B_{mn}$. These are both legitimate scalar-valued operations, and they give identical results if either $\underset{\approx}{\boldsymbol{A}}$ or $\underset{\approx}{\boldsymbol{B}}$ happens to be symmetric. However, for general (possibly non-symmetric) operands, only the definition $A_{mn}B_{mn}$ in Eq. (12.1) is an *inner-product*. The other operation, $A_{mn}B_{nm}$, is well-defined, but it is *not* an inner product. As noted in Table 1.1 on page 12, some authors [24] use the structure $\boldsymbol{A} \bullet \boldsymbol{B}$ to denote the inner product, which seriously conflicts with our structure $\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}$ which means the tensor composition (an entirely different operation). Those authors tend to denote tensor composition by $\boldsymbol{AB}$ (i.e., $A_{ik}B_{kj}$) whereas, in our notational system the similar-looking structure $\underset{\approx}{\boldsymbol{A}}\underset{\approx}{\boldsymbol{B}}$ would denote dyadic tensor multiplication resulting in a fourth-order tensor (with components $A_{ij}B_{kl}$). We shamelessly believe that, in this case, our notational conventions are superior because they are self-defining and generalize seamlessly to higher dimensions [see the discussion on page __].

Our definition of $\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}}$, in Eq. (12.1) is a summation of every component of $\underset{\approx}{\boldsymbol{A}}$ multiplied by the *corresponding* component of $\underset{\approx}{\boldsymbol{B}}$. Consequently, the tensor inner product is analogous to the ordinary dot product between vectors in ordinary physical space (see Eq. 3.60). This is no coincidence. Applying a mathematician's definition of a vector (page 230), the set of all second-order tensors can be shown to be an abstract nine-dimensional vector space. In this space, the tensor inner product serves a role exactly like the vector dot product in 3D physical space. We can define the magnitude of a tensor and the "angle" between two tensors in a manner that is completely analogous to the definitions for ordinary vectors!

An important special case of Eq. (12.1) is the inner product between two dyads:

$$\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}:\underset{\sim}{\boldsymbol{r}}\underset{\sim}{\boldsymbol{s}} = a_m b_n r_m s_n = (\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{r}})(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{s}}) \qquad (12.3)$$

An even more specialized instance of this formula applies it to the tensor inner product of basis dyads:

$$\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j:\underset{\sim}{\boldsymbol{e}}_k\underset{\sim}{\boldsymbol{e}}_l = (\underset{\sim}{\boldsymbol{e}}_i \bullet \underset{\sim}{\boldsymbol{e}}_k)(\underset{\sim}{\boldsymbol{e}}_j \bullet \underset{\sim}{\boldsymbol{e}}_l) = \delta_{ik}\delta_{jl} \qquad (12.4)$$

The double dot product is defined such that it must operate between tensors of *at least* second-order. There is no need for parentheses in Eq. (12.3) because $\underset{\sim}{\boldsymbol{b}}:\underset{\sim}{\boldsymbol{r}}$ would be meaningless — the double dot is understood to “reach out” until it is acting between two tensors (in this case, $\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}$ and $\underset{\sim}{\boldsymbol{r}}\underset{\sim}{\boldsymbol{s}}$). Thus, for example, $\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}}\underset{\sim}{\boldsymbol{c}}:\underset{\approx}{\boldsymbol{T}}$ would be interpreted as the vector $\underset{\sim}{\boldsymbol{a}}$ multiplied by the inner product of $\underset{\sim}{\boldsymbol{b}}\underset{\sim}{\boldsymbol{c}}$ with the second-order tensor $\underset{\approx}{\boldsymbol{T}}$. The indicial expression of this operation would be

$$a_i b_m c_n T_{mn} \qquad (12.5)$$

Note, for future use, that

$$\underset{\sim}{\boldsymbol{u}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{v}} = \underset{\approx}{\boldsymbol{A}}:\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}} \qquad (12.6)$$

A NON-recommended scalar-valued product

GOAL: Explain why a commonly used alternative scalar-valued product should be avoided.

Note that only *one single* tensor in Eq. (12.2) has a transpose operation on it. That equation therefore might appear to have an unsavory asymmetrical quality. A *different* double-dot product, which we will here denote by using two horizontal raised dots, is often defined in the literature as*

$$\underset{\approx}{\boldsymbol{A}}\bullet\bullet\underset{\approx}{\boldsymbol{B}} \equiv A_{mn}B_{nm} = \mathrm{tr}(\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}) \qquad (12.7)$$

With this definition, the special case of the horizontal double dot of two dyads would be $(\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{b}})\bullet\bullet(\underset{\sim}{\boldsymbol{r}}\underset{\sim}{\boldsymbol{s}}) = a_i b_j r_j s_i = (\underset{\sim}{\boldsymbol{a}} \bullet \underset{\sim}{\boldsymbol{s}})(\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{r}})$. The scalar-valued operation in Eq. (12.7) does have an aesthetically appealing symmetry to its definition, and it is a perfectly legitimate operation. However, this scalar valued product must *never* be referred to as an “*inner product*.” The operation $\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{B}}$ in Eq. (12.2) is an inner product, but the operation $\underset{\approx}{\boldsymbol{A}}\bullet\bullet\underset{\approx}{\boldsymbol{B}}$ in Eq. (12.7) is not an inner product.

We first mentioned our tensor inner product $\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{B}}$ in the context of *matrix* analysis (see Eq. 3.60), where it was subsequently shown that the operation could be used to define tensor magnitudes because $\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{A}}$ is always real and never negative. By contrast, the horizontal double-dot product $\underset{\approx}{\boldsymbol{A}}\bullet\bullet\underset{\approx}{\boldsymbol{B}}$ does *not* satisfy the positivity requirement, so it must never

* The actual notation (structure) used for this operation varies. What’s important here is the operation itself, not the notation used for it. It’s not uncommon for writers to define the “ : ” and “ •• ” operators oppositely from our definitions.

be referred to as an inner product. Both scalar-valued products, $\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}}$ and $\underset{\approx}{\boldsymbol{A}}\bullet\bullet\underset{\approx}{\boldsymbol{B}}$, will give the same result if either of the operand tensors is symmetric. Hence, the distinction between them comes when neither of the operands are symmetric. A counterexample showing that the horizontal double-dot fails positivity is

$$[\underset{\approx}{\boldsymbol{A}}] = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{12.8}$$

For this tensor,

$$\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{A}} = 2 > 0 \tag{12.9}$$

but

$$\underset{\approx}{\boldsymbol{A}}\bullet\bullet\underset{\approx}{\boldsymbol{A}} = -2 < 0 \qquad \text{(fails positivity test)} \tag{12.10}$$

Incidentally, the "$\boldsymbol{:}$" operator is also the one that arises naturally in tensor calculus. If s is some scalar function of the components of $\underset{\approx}{\boldsymbol{A}}$, then

$$s = s(A_{11}, A_{12}, \ldots, A_{33}) \tag{12.11}$$

If each component of $\underset{\approx}{\boldsymbol{A}}$ varies with time t, then the chain rule can be used to take the time rate of s:

$$\frac{ds}{dt} = \frac{\partial s}{\partial A_{11}}\frac{dA_{11}}{dt} + \frac{\partial s}{\partial A_{12}}\frac{dA_{12}}{dt} + \ldots + \frac{\partial s}{\partial A_{33}}\frac{dA_{33}}{dt} \tag{12.12}$$

Letting $\dfrac{ds}{d\underset{\approx}{\boldsymbol{A}}}$ denote the tensor whose components are $\dfrac{\partial s}{\partial A_{ij}}$, this result is written

$$\frac{ds}{dt} = \frac{ds}{d\underset{\approx}{\boldsymbol{A}}}\boldsymbol{:}\frac{d\underset{\approx}{\boldsymbol{A}}}{dt} \tag{12.13}$$

The "$\boldsymbol{:}$" operator, not "$\bullet\bullet$", appears naturally in the chain rule; each component of one tensor is multiplied by the *corresponding* component of the other tensor.

Fourth-order tensor inner product

The **inner product** between two fourth-order tensors, $\underset{\approx}{\underset{\approx}{\boldsymbol{X}}}$ and $\underset{\approx}{\underset{\approx}{\boldsymbol{Y}}}$, is a scalar denoted $\underset{\approx}{\underset{\approx}{\boldsymbol{X}}}\boldsymbol{::}\underset{\approx}{\underset{\approx}{\boldsymbol{Y}}}$ and defined

$$\underset{\approx}{\underset{\approx}{\boldsymbol{X}}}\boldsymbol{::}\underset{\approx}{\underset{\approx}{\boldsymbol{Y}}} \equiv X_{ijkl}Y_{ijkl} \tag{12.14}$$

This is a summation of every component of $\underset{\approx}{\underset{\approx}{\boldsymbol{X}}}$ multiplied by the corresponding component of $\underset{\approx}{\underset{\approx}{\boldsymbol{Y}}}$. Consequently, the fourth-order tensor inner product is analogous to the inner products that we have defined for vectors and second-order tensors. Again, this is no coincidence. By applying a mathematician's definition of a vector, the set of all fourth-order tensors can be shown to be an abstract 81-dimensional vector space. Although this view is occasionally useful in applications, we will usually find that fourth-order tensors are most conveniently regarded as operations (such as material constitutive laws) that transform second-order tensors to second-order tensors. Hence, fourth-order tensors may be regarded as *second-order* tensors referenced to nine-dimensional tensor space.

Fourth-order Sherman-Morrison formula

When regarding second-order tensors as nine-dimensional vectors, the inner product is the tensor inner product (i.e., the double-dot product). Many formulas that were derived for ordinary 3D vectors have generalizations to this higher-dimensional space. For example, a rank-one modification of a fourth-order tensor is defined by a formula similar in structure to Eq. (9.80). The fourth-order inverse is given by a formula similar to that in Eq. (9.81). Namely,

$$\text{If} \quad \underset{\approx\approx}{\boldsymbol{B}} = \underset{\approx\approx}{\boldsymbol{A}} + \underset{\approx}{\boldsymbol{V}}\underset{\approx}{\boldsymbol{W}} \qquad (B_{ijkl} = A_{ijkl} + V_{ij}W_{kl}) \tag{12.15}$$

then

$$\underset{\approx\approx}{\boldsymbol{B}}^{-1} = \underset{\approx\approx}{\boldsymbol{A}}^{-1} - \frac{\underset{\approx\approx}{\boldsymbol{A}}^{-1}\boldsymbol{:}\underset{\approx}{\boldsymbol{V}}\underset{\approx}{\boldsymbol{W}}\boldsymbol{:}\underset{\approx\approx}{\boldsymbol{A}}^{-1}}{1 + \underset{\approx}{\boldsymbol{W}}\boldsymbol{:}\underset{\approx\approx}{\boldsymbol{A}}^{-1}\boldsymbol{:}\underset{\approx}{\boldsymbol{V}}} \qquad \left(B^{-1}_{ijkl} = A^{-1}_{ijkl} - \frac{A^{-1}_{ijmn}V_{mn}W_{rs}A^{-1}_{rskl}}{1 + W_{pq}A^{-1}_{pqtu}V_{tu}}\right) \tag{12.16}$$

Structurally, this fourth-order formula is identical to the second-order formula except that the vector inner products (single dot) are replaced with tensor inner products (double dot). This formula is frequently used in plasticity theory.

Higher-order tensor inner product

Notationally, we have defined all of our inner products such that the number of "dots" indicates the number of contracted indices. Clearly, this notation is not practical for higher-order tensors. An alternative notation for an n^{th}-order inner product may be defined as the order n surrounded by a circle. Thus, for example,

$$\underset{\approx}{\underset{\approx}{\boldsymbol{X}}} \textcircled{4} \underset{\approx}{\underset{\approx}{\boldsymbol{Y}}} \text{ means the same thing as } \underset{\approx}{\underset{\approx}{\boldsymbol{X}}}::\underset{\approx}{\underset{\approx}{\boldsymbol{Y}}} \tag{12.17}$$

Some writers [e.g., Ref. 18[*]] prefer always using a single raised dot to denote all inner-products, regardless of the order. These writers demand that meaning of the single-dot operator must be inferred by the tensorial order of the arguments. The reader is further expected to infer the tensorial order of the arguments from the context of the discussion since most writers do not indicate tensor order by the number of under-tildes. These writers tend to define the multiplication of two tensors written side by side (with no multiplication symbol between them) to be the tensor composition. For example, when they write $\boldsymbol{AB}$ between two tensors that have been identified as second-order, then they mean what we would write as $\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}$. When they write $\boldsymbol{UV}$ between two tensors that have been identified as fourth-order, they mean what we would write as $\underset{\approx}{\underset{\approx}{\boldsymbol{U}}}:\underset{\approx}{\underset{\approx}{\boldsymbol{V}}}$. Such notational conventions are undeniably easier to typeset, and they work fine whenever one restricts attention to the small set of conventional tensor operations normally seen in trivial applications. However, more exotic advanced tensor operations become difficult to define under this system. A consistent self-defining system such as the one used in this book is far more convenient and flexible.

Self-defining notation

Throughout this book, our notation is self-defining in the sense that the meaning of an expression can always be ascertained by expanding all arguments in basis form, as discussed on page 128. The following list shows several indicial expressions along with their direct notation expressions under our notation

$U_{mnpq}V_{mnpq}$	$\underset{\approx}{\underset{\approx}{\boldsymbol{U}}}::\underset{\approx}{\underset{\approx}{\boldsymbol{V}}}$
$U_{ijpq}V_{pqkl}$	$\underset{\approx}{\underset{\approx}{\boldsymbol{U}}}:\underset{\approx}{\underset{\approx}{\boldsymbol{V}}}$

[*] We call attention to this reference not because it is the only example, but because it a continuum mechanics textbook that is in common use today and may therefore be familiar to a larger audience. This notation draws from older classic references [e.g., 24]. Older should not always be taken to mean inferior, but we believe that, in this case, the older tensor notation is needlessly flawed. Our notation demands that a different symbol be used for each different inner product for each differently ordered tensor space, whereas the older style overloads *the same symbol* to mean different inner products — operator overloading can be extremely useful in many situations, but we feel it does more harm than good in this case because it precludes self-defining notation.

$$
\begin{array}{ll}
U_{ijkp}V_{plmn} & \underset{\approx}{\underset{\approx}{\mathbf{U}}} \bullet \underset{\approx}{\underset{\approx}{\mathbf{V}}} \\[2ex]
U_{ijkl}V_{mnpq} & \underset{\approx}{\underset{\approx}{\mathbf{U}}}\underset{\approx}{\underset{\approx}{\mathbf{V}}} \\[2ex]
A_{pq}U_{pqij} & \underset{\approx}{\mathbf{A}} : \underset{\approx}{\underset{\approx}{\mathbf{U}}} \\[2ex]
A_{ip}U_{pjkl} & \underset{\approx}{\mathbf{A}} \bullet \underset{\approx}{\underset{\approx}{\mathbf{U}}} \\[2ex]
A_{ij}U_{klmn} & \underset{\approx}{\mathbf{A}}\underset{\approx}{\underset{\approx}{\mathbf{U}}} \\[2ex]
A_{ip}\varepsilon_{pjq}U_{qklm} & -\underset{\approx}{\mathbf{A}} \times \underset{\approx}{\underset{\approx}{\mathbf{U}}}^{*}
\end{array}
\tag{12.18}
$$

Writers who use inconsistent non-self-defining notational structures would be hard-pressed to come up with easily remembered direct notations for all of the above operations. Their only recourse would be to meekly argue that such operations would never be needed in real applications anyway. Before we come off sounding too pompous, we acknowledge that there exist indicial expressions that do not translate elegantly into our system. For example, the equation

$$A_{ijkl}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j\underset{\sim}{\mathbf{e}}_k\underset{\sim}{\mathbf{e}}_l = U_{ikjl}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j\underset{\sim}{\mathbf{e}}_k\underset{\sim}{\mathbf{e}}_l \tag{12.19}$$

would have to be written under our notational system as

$$\underset{\approx}{\underset{\approx}{\mathbf{A}}} = X_2^3\left(\underset{\approx}{\underset{\approx}{\mathbf{U}}}\right) \tag{12.20}$$

where the rather non-intuitive swap operator X_2^3 is defined in Eq. (19.44). Of course, older notation systems have no commonly recognized direct notation for this operation either. This particular operation occurs so frequently that we (later) introduce a new “leafing” operator to denote it by $\underset{\approx}{\underset{\approx}{\mathbf{A}}} = \underset{\approx}{\underset{\approx}{\mathbf{U}}}^{L}$ as an alternative to Eq. (12.20). Even when using the notational scheme that we advocate, writers should always provide indicial expressions to clarify their notations, especially when the operations are rather unusual.

The difficulties with direct notation might seem to suggest that perhaps indicial notation would be the best choice. In some instances, this is true. However, even indicial notation has its pitfalls, principally in regard to operator precedence. For example, the notation

$$\frac{\partial f}{\partial \sigma_{kk}} \tag{12.21}$$

is ambiguous. It could mean

[*] In this equation, the negative appears because the cross-product is defined such that the summed indices on the alternating symbol must be adjacent (making them adjacent involves a negative permutation of ε_{pjq} to make it $-\varepsilon_{jpq}$.

$$\frac{\partial f}{\partial(\mathrm{tr}\underset{\approx}{\sigma})} \quad \text{or} \quad \mathrm{tr}\left(\frac{\partial f}{\partial \underset{\approx}{\sigma}}\right) \tag{12.22}$$

The two operations give different results. Furthermore, we have already seen that the book-keeping needed to satisfy the summation conventions is tedious, error-prone, often limited to Cartesian components, distracting from general physical interpretations, and (in some cases) not well-suited to calculus manipulations. Nonetheless, there are certainly many instances where indicial notation is the most lucid way to go.

Bottom line: in your own work, use the notation you prefer, but in published and presented work, always employ notation that seems most likely to accomplish the desired interpretation by an educated reader. Your goal is to convince them of the truth of a scientific principle, not to intimidate, condescend, or baffle them with your (or our) whacked out notations (that we, of course, regard as brilliant and self-evident <g>).

The magnitude of a tensor or a vector

The magnitude of a second-order tensor $\underset{\approx}{\boldsymbol{A}}$ is a scalar denoted $\|\underset{\approx}{\boldsymbol{A}}\|$ defined

$$\|\underset{\approx}{\boldsymbol{A}}\| \equiv \sqrt{\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{A}}} \tag{12.23}$$

This definition has exactly the same form as the more familiar definition of the **magnitude of a simple vector** $\boldsymbol{v}$:

$$\|\underset{\sim}{\boldsymbol{v}}\| \equiv \sqrt{\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{v}}} \tag{12.24}$$

Though rarely needed, the magnitude of a fourth-order tensor $\underset{\approx\approx}{\boldsymbol{X}}$ is a scalar defined

$$\|\underset{\approx\approx}{\boldsymbol{X}}\| = \sqrt{\underset{\approx\approx}{\boldsymbol{X}}::\underset{\approx\approx}{\boldsymbol{X}}} \tag{12.25}$$

A vector is zero if and only if its magnitude is zero. Likewise, a tensor is zero if and only if its magnitude is zero.

Useful inner product identities

The symmetry and deviator decompositions of tensors are often used in conjunction with the following identities:

$$\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{B}} = \mathrm{sym}\underset{\approx}{\boldsymbol{A}}:\mathrm{sym}\underset{\approx}{\boldsymbol{B}} + \mathrm{skw}\underset{\approx}{\boldsymbol{A}}:\mathrm{skw}\underset{\approx}{\boldsymbol{B}} \tag{12.26}$$

$$\underset{\approx}{\boldsymbol{A}}:\underset{\approx}{\boldsymbol{B}} = \mathrm{dev}\underset{\approx}{\boldsymbol{A}}:\mathrm{dev}\underset{\approx}{\boldsymbol{B}} + \mathrm{iso}\underset{\approx}{\boldsymbol{A}}:\mathrm{iso}\underset{\approx}{\boldsymbol{B}} \tag{12.27}$$

Decomposition of the tensors into its symmetric plus skew symmetric parts ($\underset{\approx}{\boldsymbol{A}} = \mathrm{sym}\underset{\approx}{\boldsymbol{A}} + \mathrm{skw}\underset{\approx}{\boldsymbol{A}}$ and $\underset{\approx}{\boldsymbol{B}} = \mathrm{sym}\underset{\approx}{\boldsymbol{B}} + \mathrm{skw}\underset{\approx}{\boldsymbol{B}}$) represents an orthogonal projection decomposition that is completely analogous to Eq. (10.24). Thus, Eq. (12.26) is a specific application of Eq. (10.26) in which tensors are interpreted in their V_9^1 sense. A similar statement holds for the decomposition of tensors into deviatoric plus isotropic parts.

If $\underset{\approx}{\boldsymbol{B}}$ happens to be a symmetric tensor (*i.e.,* if $\text{skw}\underset{\approx}{\boldsymbol{B}} = \underset{\approx}{\boldsymbol{0}}$) then the inner product between $\underset{\approx}{\boldsymbol{B}}$ any other tensor $\underset{\approx}{\boldsymbol{A}}$ will depend only on the symmetric part of $\underset{\approx}{\boldsymbol{A}}$. Consequently, sometimes researchers will replace $\underset{\approx}{\boldsymbol{A}}$ by its symmetric part *without any loss in generality* — which can save on storage in numerical computations, but is unwise if there is any chance that $\underset{\approx}{\boldsymbol{A}}$ will need to be used in any other context.

Incidentally, note that the "trace" operation defined in Eq. (11.64) can be written as an inner product inner product with the identity tensor:

$$\boxed{\text{tr}\underset{\approx}{\boldsymbol{A}} \equiv \underset{\approx}{\boldsymbol{I}}\boldsymbol{:}\underset{\approx}{\boldsymbol{A}}} \tag{12.28}$$

Also note that $\underset{\approx}{\boldsymbol{I}}\boldsymbol{:}\underset{\approx}{\boldsymbol{I}} = \text{tr}\underset{\approx}{\boldsymbol{I}} = 3$, so Eq. (12.27) may be alternatively written

$$\boxed{\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}} = \underset{\approx}{\boldsymbol{A}}'\boldsymbol{:}\underset{\approx}{\boldsymbol{B}}' + \frac{1}{3}(\text{tr}\underset{\approx}{\boldsymbol{A}})(\text{tr}\underset{\approx}{\boldsymbol{B}})} \tag{12.29}$$

Distinction between an N$^{\text{th}}$-order tensor and an N$^{\text{th}}$-rank tensor

Many authors use the term "N^{th}-rank tensor" to mean what we would call an "N^{th}-order tensor". We don't adopt this practice because the term "rank" has a specific meaning in matrix analysis that applies equally well for tensor analysis. The "rank" of a second-order tensor is here defined to equal the rank of the tensor's Cartesian component matrix (i.e., the number of linearly independent rows or columns). That way, for example, when we speak of rank-1 projectors, we are speaking of second-order tensors that have a matrix rank of 1. Of course, our practice of saying N^{th}-*order* tensors has its downside too because it can cause confusion when discussing tensor polynomials.

When a second-order tensor is regarded as an operation that takes vectors to vectors, then the "rank" of the second-order tensor is the dimension of the range space. For example, if a second-order tensor projects a vector into its part in the direction of some fixed unit vector, then the result of that tensor operation will always be a vector that is a multiple of the unit vector. This range space is one-dimensional, so the rank of this second-order tensor is 1.

Based on well-known matrix theory, we note that a second-order tensor is invertible only if its rank is 3. We have already used the term rank in connection with projections. A rank-2 second-order projection is a tensor that projects vectors to a 2-dimensional space. The rank of the component matrix equals the rank of the projector.

Fourth-order oblique tensor projections

Second-order tensors are themselves 9-dimensional abstract vectors of class V_9^1 with "$\boldsymbol{:}$" denoting the inner product. Consequently, operations that are defined for ordinary 3D vectors have analogs for tensors. Recall that Eq. (6.17) gave the formula for the oblique projection of a vector $\underset{\sim}{\boldsymbol{x}}$ onto a plane perpendicular to a given vector $\underset{\sim}{\boldsymbol{b}}$. The "light rays" defining the projection direction were parallel to the vector $\underset{\sim}{\boldsymbol{a}}$. The analog of Eq. (6.17) for tensors is

$$P(\underset{\approx}{\boldsymbol{X}}) = \underset{\approx}{\boldsymbol{X}} - \frac{\underset{\approx}{\boldsymbol{A}}(\underset{\approx}{\boldsymbol{B}}\boldsymbol{:}\underset{\approx}{\boldsymbol{X}})}{\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}}} . \qquad (12.30)$$

As for the projection in 3-space, this operation represents a linear oblique projection in tensor space. The "surface" to which $\underset{\approx}{\boldsymbol{X}}$ is projected is orthogonal to $\underset{\approx}{\boldsymbol{B}}$ and the oblique projection direction is aligned with $\underset{\approx}{\boldsymbol{A}}$. This projection function appears in the study of plasticity [8] in which a trial stress state is returned to the yield surface via a projection of this form.

The fourth-order projection transformation can be readily verified to have the following properties:

$$P(\alpha\underset{\approx}{\boldsymbol{X}}) = \alpha P(\underset{\approx}{\boldsymbol{X}}) \text{ for all scalars } \alpha . \qquad (12.31)$$

$$P(\underset{\approx}{\boldsymbol{X}} + \underset{\approx}{\boldsymbol{Y}}) = P(\underset{\approx}{\boldsymbol{X}}) + P(\underset{\approx}{\boldsymbol{Y}}) \text{ for all } \underset{\approx}{\boldsymbol{X}} \text{ and } \underset{\approx}{\boldsymbol{Y}} . \qquad (12.32)$$

$$P(P(\underset{\approx}{\boldsymbol{X}})) = P(\underset{\approx}{\boldsymbol{X}}) . \qquad (12.33)$$

The first two properties simply indicate that the projection operation is linear. The last property says that projecting a tensor that has already been projected merely gives the tensor back unchanged.

Finally, the analog of Eqs. (6.34) and (6.35) is the important identity that

$$P(\underset{\approx}{\boldsymbol{X}}) = P(\underset{\approx}{\boldsymbol{Y}}) \qquad \text{if and only if} \qquad \underset{\approx}{\boldsymbol{X}} = \underset{\approx}{\boldsymbol{Y}} + \beta\underset{\approx}{\boldsymbol{A}} . \qquad (12.34)$$

This identity is used, for example, to prove the validity of radial return algorithms in plasticity theory [8].

Leafing and palming operations

GOAL: Introduce a simple, but obscure, higher-order tensor operation

Consider a deck of cards. If there are an even number of cards, you can split the deck in half and (in principle) leaf the cards back together in a perfect shuffle. We would call this a leafing operation. If, for example, there were six cards in the deck initially ordered sequentially, then, after the leafing operation (perfect shuffle), they would be ordered 142536. If the deck had only four cards, they would leaf into the ordering 1324.

We will here define a similar operation that applies to any even order tensor. The structure to indicate application of this leafing operation will be a superscript "L." Let

$$\underset{\approx}{\mathbf{U}} = U_{ijpq}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j\underset{\sim}{\mathbf{e}}_p\underset{\sim}{\mathbf{e}}_q \tag{12.35}$$

denote a fourth-order tensor. The “leaf” of this tensor will be defined

$$\underset{\approx}{\mathbf{U}}^L = U_{ipjq}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_j\underset{\sim}{\mathbf{e}}_p\underset{\sim}{\mathbf{e}}_q \tag{12.36}$$

Note that the leaf was obtained by a perfect shuffle of the indices on U_{ijpq}. In purely indicial notation, we would write

$$U^L_{ijpq} = U_{ipjq} \tag{12.37}$$

Note that shuffling the indices is equivalent to shuffling the dyadic ordering of the base vectors. In other words, the equation

$$\underset{\approx}{\mathbf{U}} = U_{ijpq}\underset{\sim}{\mathbf{e}}_i\underset{\sim}{\mathbf{e}}_p\underset{\sim}{\mathbf{e}}_j\underset{\sim}{\mathbf{e}}_q \tag{12.38}$$

is equivalent to Eq. (12.36). Note that leafing a fourth-order tensor is equivalent to simply swapping the middle two indices.

Derivative of a leafing operation:

$$\frac{\partial U^L_{ijpq}}{\partial U_{mnrs}} = \frac{\partial U_{ipjq}}{\partial U_{mnrs}} = \delta_{im}\delta_{pn}\delta_{jr}\delta_{qs} = \delta_{im}\delta_{jr}\delta_{pn}\delta_{qs} \tag{12.39}$$

The leaf of a sixth-order tensor with components U_{ijkpqr} would be

$$U^L_{ijkpqr} = U_{ipjqkr} \tag{12.40}$$

Now consider a different (screwy) way to shuffle a deck of cards. First the deck is split in half, but then the second half is reversed before shuffling. For example, a six-card deck, originally ordered 123456 would split into halves 123 and 456. After reversing the order of the second half, the halves would be 123 654, and then shuffling would give 162534. We will call the analog of this operation on tensor indices a “palming” operation and denote it with a superscript Γ (i.e., an upside down “L”). Then, for fourth- and sixth-order tensors, the palming operator would give

$$U^\Gamma_{ijkl} = U_{iljk} \tag{12.41}$$

and

$$U^\Gamma_{ijkpqr} = U_{irjqkp} \tag{12.42}$$

The leafing and palming operations have been introduced simply because these types of index re-orderings occur frequently in higher-order analyses, and there is no straightforward way to characterize them in existing direct structural notation. Using these new operations, note that the e-δ identity can be written

$$\underset{\approx}{\underset{\sim}{\varepsilon}} \bullet \underset{\approx}{\underset{\sim}{\varepsilon}} = (\underset{\approx}{\boldsymbol{I}}\underset{\approx}{\boldsymbol{I}})^L - (\underset{\approx}{\boldsymbol{I}}\underset{\approx}{\boldsymbol{I}})^\Gamma \tag{12.43}$$

Here, $\underset{\approx}{\boldsymbol{I}}\underset{\approx}{\boldsymbol{I}}$ is a dyad so that $(\underset{\approx}{\boldsymbol{I}}\underset{\approx}{\boldsymbol{I}})_{ijmn} = \delta_{ij}\delta_{mn}$ and therefore $(\underset{\approx}{\boldsymbol{I}}\underset{\approx}{\boldsymbol{I}})^L_{ijmn} = \delta_{im}\delta_{jn}$ and $(\underset{\approx}{\boldsymbol{I}}\underset{\approx}{\boldsymbol{I}})^\Gamma_{ijmn} = \delta_{in}\delta_{jm}$.

Symmetric Leafing. Given a fourth-order tensor U_{ijkl}, a common operation in materials modeling involves *minor-symmetrizing* the minor indices. Just as a tensor A_{ij} can be symmetrized by $\frac{1}{2}(A_{ij} + A_{ji})$, a fourth-order tensor can be minor-symmetrized by

$$U^\sigma_{ijkl} = U_{(ij)(kl)} = \frac{1}{4}(U_{ijkl} + U_{jikl} + U_{ijlk} + U_{jilk}) \tag{12.44}$$

Here, we have employed a common indicial notation convention that pairs of indices in parentheses are to be expanded in a symmetric sum.

Now consider the leafed tensor $U^L_{ijkl} = U_{ikjl}$. Even if the tensor U_{ijkl} is minor-symmetric, its leaf will not necessarily be minor symmetric. The symmetrized leaf is denoted with a superscript Λ and defined

$$U^\Lambda_{ijkl} = U^L_{(ij)(kl)} = \frac{1}{4}(U_{ikjl} + U_{kijl} + U_{iklj} + U_{kilj}) \tag{12.45}$$

Symmetrized leafs are extremely common in constitutive modeling (especially for anisotropy).

"The chief function of your body is to carry your brain around." — Thomas Edison

13. Coordinate/basis transformations

Change of basis (and coordinate transformations)

Quite frequently, people speak of coordinate transformations as if they were the same thing as *basis* transformations. They're not. The coordinates of a point in 3D space comprise exactly three independent numbers. Each distinct set of coordinates uniquely identifies the location of the point in space. For example, cylindrical *coordinates* are $\{r, \theta, z\}$. In the vast majority of applications, the *base vectors* are defined in a way that is coupled to the choice of coordinates. For example, the base vectors that are natural for use with cylindrical coordinates are $\{\underset{\sim}{\boldsymbol{e}}_r, \underset{\sim}{\boldsymbol{e}}_\theta, \underset{\sim}{\boldsymbol{e}}_z\}$, and these orthonormal vectors are defined to point in the direction of increasing values of the corresponding *coordinate*. The *components* of the position vector $\underset{\sim}{\boldsymbol{x}}$ are the coefficients of the vector when it is expanded as a linear combination of the base vectors. For cylindrical coordinates, the position vector is given by $\underset{\sim}{\boldsymbol{x}} = r\underset{\sim}{\boldsymbol{e}}_r + z\underset{\sim}{\boldsymbol{e}}_z$; note that there is no term in which the second base vector $\underset{\sim}{\boldsymbol{e}}_\theta$ appears. Therefore, even though $\underset{\sim}{\boldsymbol{x}}$ depends on *three* coordinates $\{r, \theta, z\}$, it has only two nonzero *components*. Dependence of the position vector $\underset{\sim}{\boldsymbol{x}}$ on the angular coordinate is contained implicitly in the dependence of the radial base vector $\underset{\sim}{\boldsymbol{e}}_r$ on θ. For rectangular cartesian coordinates, referenced to the laboratory basis, it *does* so happen that the vector components are identical to the vector coordinates $\{x_1, x_2, x_3\}$.

As mentioned above, the choice of basis is almost always motivated by the choice of coordinates so that each base vector points in the direction of increasing values of the associated coordinate. However, there is no divine edict that demands that the base vectors *must* be coupled in any way to the coordinates. If, for example, you were studying the mechanics of a ferris wheel, then you might favor using cylindrical coordinates to identify points in space (with the z axis parallel to the wheel's axis, but *the laboratory basis* (with, say, $\underset{\sim}{\boldsymbol{e}}_2$ perpendicular to the ground) to reference physical vectors such as the acceleration of gravity. Using the cylindrical base vectors, the acceleration of gravity would point along $-(\sin\theta)\underset{\sim}{\boldsymbol{e}}_r - \cos\theta\underset{\sim}{\boldsymbol{e}}_\theta$, but using the lab basis, it is simply proportional to $\underset{\sim}{\boldsymbol{e}}_2$. In this section, we describe the effect of changing the *basis*. This discussion is, in principal, completely unrelated to changing *coordinates*.

Consider two different *orthonormal*[*] triads: a reference (or laboratory) basis $\{\underset{\sim}{\boldsymbol{E}}_1, \hat{\underset{\sim}{\boldsymbol{e}}}_1, \hat{\underset{\sim}{\boldsymbol{e}}}_1\}$ and a “tilde” basis $\{\tilde{\underset{\sim}{\boldsymbol{e}}}_1, \tilde{\underset{\sim}{\boldsymbol{e}}}_2, \tilde{\underset{\sim}{\boldsymbol{e}}}_3\}$. A vector $\underset{\sim}{\boldsymbol{v}}$ can be expressed in terms of either basis as follows:

$$\underset{\sim}{\boldsymbol{v}} = \hat{v}_i\hat{\underset{\sim}{\boldsymbol{e}}}_i\text{, where } \hat{v}_i \text{ are the components of } \underset{\sim}{\boldsymbol{v}} \text{ w.r.t the } \hat{\underset{\sim}{\boldsymbol{e}}}_i \text{ basis} \qquad (13.1)$$

$$\underset{\sim}{\boldsymbol{v}} = \tilde{v}_i\tilde{\underset{\sim}{\boldsymbol{e}}}_i\text{, where } \tilde{v}_i \text{ are the components of } \underset{\sim}{\boldsymbol{v}} \text{ w.r.t the } \tilde{\underset{\sim}{\boldsymbol{e}}}_i \text{ basis} \qquad (13.2)$$

Both expressions represent the *same* vector $\underset{\sim}{\boldsymbol{v}}$. Thus, it must be true that

$$\hat{v}_i\hat{\underset{\sim}{\boldsymbol{e}}}_i = \tilde{v}_i\tilde{\underset{\sim}{\boldsymbol{e}}}_i \qquad (13.3)$$

The “hat” components $\{\hat{v}_1, \hat{v}_2, \hat{v}_3\}$ are not generally equal to the “tilde” components $\{\tilde{v}_1, \tilde{v}_2, \tilde{v}_3\}$, but they are related in a very specific way. Specifically, if the two bases are not equal to each other, then the components with respect to each of these bases must also be different from each other. However, the components must change in response to a basis change in such a way that the *sum* of components times base vectors is invariant — either representation characterizes the *same* vector.

Coordinate and basis transformations are awkward to explain, and everyone has their own way of doing it. Our Reference [7] provides detailed derivations of the procedure for the case of changing from one orthonormal basis $\{\tilde{\underset{\sim}{\boldsymbol{e}}}_1, \tilde{\underset{\sim}{\boldsymbol{e}}}_2, \tilde{\underset{\sim}{\boldsymbol{e}}}_3\}$ to another reoriented, but still orthonormal, basis $\{\hat{\underset{\sim}{\boldsymbol{e}}}_1, \hat{\underset{\sim}{\boldsymbol{e}}}_2, \hat{\underset{\sim}{\boldsymbol{e}}}_3\}$[†]. Coordinate transformation discussions always begin with the definition of a “direction cosine” matrix $[Q]$ whose components are given by

$$\boxed{Q_{ij} = \underset{\sim}{\boldsymbol{E}}_i \bullet \hat{\underset{\sim}{\boldsymbol{e}}}_j} \qquad (13.4)$$

This is called the direction cosine matrix because, applying Eq. (5.5),

$$Q_{ij} = \cos\theta_{ij}, \qquad (13.5)$$

where θ_{ij} is the angle between $\underset{\sim}{\boldsymbol{E}}_i$ and $\hat{\underset{\sim}{\boldsymbol{e}}}_j$. Thankfully, you don’t have to find these angles to construct the $[Q]$ matrix. Just find the components of each reoriented $\hat{\underset{\sim}{\boldsymbol{e}}}_j$ base vector with respect to the reference $\tilde{\underset{\sim}{\boldsymbol{e}}}_k$ basis. Then assemble these components into *columns* of the $[Q]$ matrix so that the j^{th} column contains the tilde components of the $\hat{\underset{\sim}{\boldsymbol{e}}}_j$ base vector. An example is provided at the end of this section.

Some people define the direction cosine matrix alternatively by

$$L_{ij} = \underset{\sim}{\boldsymbol{E}}_j \bullet \hat{\underset{\sim}{\boldsymbol{e}}}_i = \hat{\underset{\sim}{\boldsymbol{e}}}_i \bullet \underset{\sim}{\boldsymbol{E}}_j \qquad (13.6)$$

Note that $L_{ij} = Q_{ji}$ therefore the results that we present below can be readily converted to this alternative definition by simply replacing all occurrences of $[Q]$ by $[L]^T$.

[*] For a discussion of general non-orthonormal bases, see Ref. [6].

[†] In our Reference [7], the notation is different, but the meaning is the same. Specifically, Reference [7] uses $\underset{\sim}{\boldsymbol{E}}_k$ to denote what we are calling $\tilde{\underset{\sim}{\boldsymbol{e}}}_k$. It uses $\underset{\sim}{\boldsymbol{e}}_k$ to denote what we here call $\hat{\underset{\sim}{\boldsymbol{e}}}_k$. It uses v_k to denote what we are here calling $\tilde{v}_k$. Tildes in Ref. [7], become “hats” in this document. The equations listed in this document have already been converted to the present notation.

Our Reference [7] provides proofs for the following key equations governing basis reorientation:

$$\boxed{[Q]^{-1} = [Q]^T} \qquad \text{(i.e., the direction cosine matrix is orthogonal)} \tag{13.7}$$

If the handedness of both bases is the same, then $\det[Q] = +1$. If the handedness of one basis is different from the other, then $\det[Q] = -1$.

$$\boxed{\hat{\underset{\sim}{e}}_i = Q_{ji}\tilde{\underset{\sim}{e}}_j} \tag{13.8}$$

$$\boxed{\tilde{\underset{\sim}{e}}_i = Q_{ij}\hat{\underset{\sim}{e}}_j} \tag{13.9}$$

For a vector $\underset{\sim}{v} = \tilde{v}_i\tilde{\underset{\sim}{e}}_i = \hat{v}_j\hat{\underset{\sim}{e}}_j$, the components are related by

$$\boxed{\hat{v}_j = Q_{ij}\tilde{v}_i}. \qquad \text{In matrix form,} \quad \{\hat{v}\} = [Q]^T\{\tilde{v}\} \tag{13.10}$$

$$\boxed{\tilde{v}_i = Q_{ij}\hat{v}_j}. \qquad \text{In matrix form,} \quad \{\tilde{v}\} = [Q]\{\hat{v}\} \tag{13.11}$$

For a second-order tensor $\underset{\approx}{T} = \tilde{T}_{ij}\tilde{\underset{\sim}{e}}_i\tilde{\underset{\sim}{e}}_j = \hat{T}_{kl}\hat{\underset{\sim}{e}}_k\hat{\underset{\sim}{e}}_l$, the components are related by

$$\hat{T}_{kl} = Q_{ik}Q_{jl}\tilde{T}_{ij}, \text{ or in matrix form, } [\hat{T}] = [Q]^T[\tilde{T}][Q] \tag{13.12}$$

$$\tilde{T}_{ij} = Q_{ik}Q_{jl}\hat{T}_{kl}, \text{ or in matrix form, } [\tilde{T}] = [Q][\hat{T}][Q]^T \tag{13.13}$$

Higher order tensors transform similarly. For example, if $\underset{\approx}{\xi}$ is a third order tensor, then

$$\hat{\xi}_{pqr} = Q_{ip}Q_{jq}Q_{kr}\tilde{\xi}_{ijk} \tag{13.14}$$

$$\tilde{\xi}_{ijk} = Q_{ip}Q_{jq}Q_{kr}\hat{\xi}_{pqr} \tag{13.15}$$

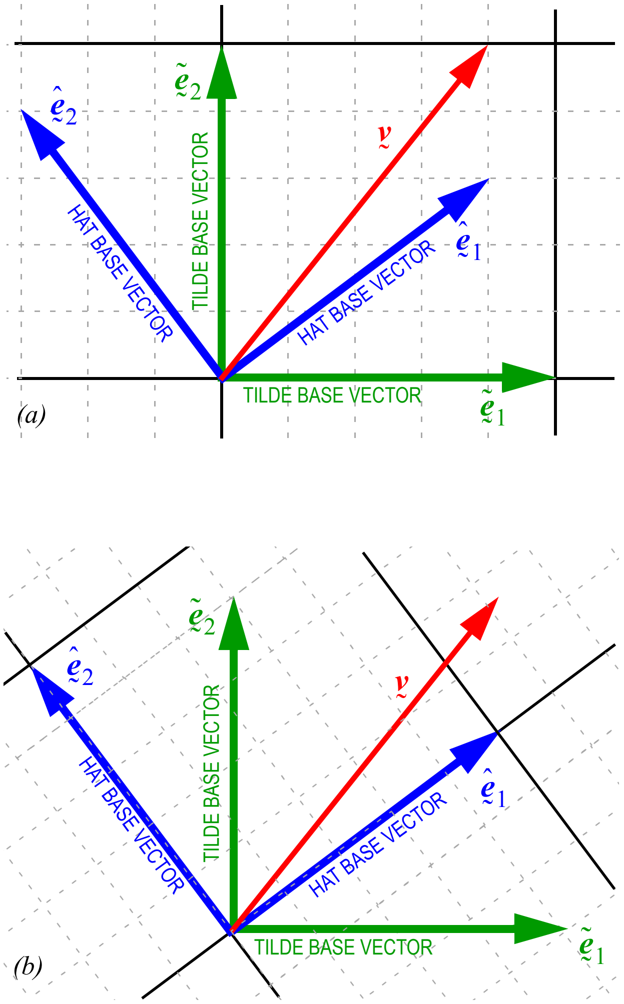
To remember these equations, just recognize that transforming an n-th order tensor requires the presence of n occurrences of Q. Furthermore, the first index on Q will always match the index on the “tilde” component while the second index on Q will always match the index on the hat component.

For transformations within a plane (i.e., when only two base vectors change while the other two reorient in the plane), a graphical tool called Mohr’s circle can be used. Although Mohr’s circle is typically taught only for symmetric matrices, it generalizes easily for non-symmetric matrices (see Ref. [9]).

For changing from one nonorthonormal basis to another, the transformation rules are considerably more complicated (see Ref [6]).

EXAMPLE.

Most of the time, you know the components of one basis with respect to the other. In Fig. 13.1, for example, we show two differently aligned “tilde” and “hat” bases.


$\hat{\underset{\sim}{e}}_2$
$\tilde{\underset{\sim}{e}}_2$
$\underset{\sim}{v}$
HAT BASE VECTOR
TILDE BASE VECTOR
HAT BASE VECTOR
$\hat{\underset{\sim}{e}}_1$
TILDE BASE VECTOR
$\tilde{\underset{\sim}{e}}_1$
(a)
$\tilde{\underset{\sim}{e}}_2$
$\hat{\underset{\sim}{e}}_2$
$\underset{\sim}{v}$
HAT BASE VECTOR
TILDE BASE VECTOR
HAT BASE VECTOR
$\hat{\underset{\sim}{e}}_1$
TILDE BASE VECTOR
$\tilde{\underset{\sim}{e}}_1$
(b)

$$\hat{\underset{\sim}{e}}_1 = \frac{4}{5}\tilde{\underset{\sim}{e}}_1 + \frac{3}{5}\tilde{\underset{\sim}{e}}_2$$

$$\hat{\underset{\sim}{e}}_2 = -\frac{3}{5}\tilde{\underset{\sim}{e}}_1 + \frac{4}{5}\tilde{\underset{\sim}{e}}_2$$

$$\hat{\underset{\sim}{e}}_3 = \tilde{\underset{\sim}{e}}_3$$

$$\underset{\sim}{v} = \frac{4}{5}\tilde{\underset{\sim}{e}}_1 + \tilde{\underset{\sim}{e}}_2$$

$$\tilde{\underset{\sim}{e}}_1 = \frac{4}{5}\hat{\underset{\sim}{e}}_1 - \frac{3}{5}\hat{\underset{\sim}{e}}_2$$

$$\tilde{\underset{\sim}{e}}_2 = \frac{3}{5}\hat{\underset{\sim}{e}}_1 + \frac{4}{5}\hat{\underset{\sim}{e}}_2$$

$$\tilde{\underset{\sim}{e}}_3 = \hat{\underset{\sim}{e}}_3$$

$$\underset{\sim}{v} = 1.24\hat{\underset{\sim}{e}}_1 + 0.32\hat{\underset{\sim}{e}}_2$$

Figure 13.1. Relative basis orientations. In *(a)*, a grid is set up to be aligned with the tilde basis (with dashed grid lines at intervals 1/5 of a unit apart), and the hat basis is expressed in terms of the grid (tilde) basis as shown. In *(b)*, a grid is set up oppositely. Note that the vectors themselves are the same in parts *(a)* and *(b)* — only the orientation of the observer (as indicated by the grid) is different. The vector $\underset{\sim}{v}$ is the same in both drawings, but it has different *components* with respect to the different bases.

In Fig. 13.1*(a)*, we set up a grid to be aligned with the “tilde” basis. Our goal is to change over to the “hat” basis. In particular, we would like to compute the components of the red vector, $\underset{\sim}{v}$, with respect to the “hat” basis. We could solve this problem graphically (and therefore approximately) by simply setting up a new grid [Fig. 13.1*(b)*] that is aligned with the “hat” basis and then just reading off the components of the red vector with respect to this new grid. To solve the problem exactly, we need to use transformation formulas.

The hat basis is expressed in terms of the “tilde” basis as shown in Fig. 13.1*(a)*. By expressing the “hat” basis in terms of the “tilde” basis, we can compute the direction cosines. For example,

$$Q_{12} = \tilde{\underset{\sim}{e}}_1 \bullet \hat{\underset{\sim}{e}}_2 = \tilde{\underset{\sim}{e}}_1 \bullet \left(-\frac{3}{5}\tilde{\underset{\sim}{e}}_1 + \frac{4}{5}\tilde{\underset{\sim}{e}}_2 \right) = -\frac{3}{5} \qquad (13.16)$$

This would be the normal way that you would compute Q_{12}. However, it can be done differently. In Fig. 13.1*(b)*, we have a grid aligned with the “hat” basis, and the tilde basis is expressed in terms of the “hat” basis as shown. In this scenario, the calculation of Q_{12} can be performed by

$$Q_{12} = \tilde{\underset{\sim}{e}}_1 \bullet \hat{\underset{\sim}{e}}_2 = \left(\frac{4}{5}\hat{\underset{\sim}{\mathbf{e}}}_1 - \frac{3}{5}\hat{\underset{\sim}{\mathbf{e}}}_2 \right) \bullet \hat{\underset{\sim}{e}}_2 = -\frac{3}{5} \qquad (13.17)$$

Either approach produces the same result for Q_{12}. Proceeding similarly to compute the other components of $[Q]$ gives

$$[Q] = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (13.18)$$

Note that the columns of the [Q] matrix contain the tilde components of the hat base vectors. The rows of [Q] contain the hat components of the tilde basis. For this reason, the [Q] matrix is often presented in the form of a table that explicitly shows the base vectors. For this example, the direction cosine table would be written

Direction cosine table

	$\hat{\underset{\sim}{e}}_1$	$\hat{\underset{\sim}{e}}_2$	$\hat{\underset{\sim}{e}}_3$
$\tilde{\underset{\sim}{e}}_1$	4/5	–3/5	0
$\tilde{\underset{\sim}{e}}_2$	3/5	4/5	0
$\tilde{\underset{\sim}{e}}_3$	0	0	1

Now consider the red vector $\underset{\sim}{v}$ shown in Fig. 13.1. Let’s suppose that we know how to express this vector with respect to the tilde basis. Namely,

$$\underset{\sim}{v} = \frac{4}{5}\tilde{\underset{\sim}{v}}_1 + \tilde{\underset{\sim}{v}}_2 \qquad (13.19)$$

Therefore

$$\tilde{v}_1 = \frac{4}{5}, \qquad \tilde{v}_2 = 1, \qquad \tilde{v}_3 = 0 \tag{13.20}$$

Our goal is to find the "hat" components of this vector. The direction cosine matrix for this example was already computed in Eq. (13.18). Substituting the transpose of this matrix into Eq. (13.10) gives us the result for the components of this vector with respect to the hat basis:

$$\begin{Bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \hat{v}_3 \end{Bmatrix} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{Bmatrix} 4/5 \\ 1 \\ 0 \end{Bmatrix} = \begin{bmatrix} \frac{4}{5} & \frac{3}{5} & 0 \\ -\frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 4/5 \\ 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 31/25 \\ 8/25 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1.24 \\ 0.32 \\ 0 \end{Bmatrix} \tag{13.21}$$

or

$$\underset{\sim}{\boldsymbol{v}} = 1.24\hat{\underset{\sim}{\boldsymbol{e}}}_1 + 0.32\hat{\underset{\sim}{\boldsymbol{e}}}_2 \tag{13.22}$$

This result is consistent with the graphical estimate shown in Fig. 13.1.

Definition of a vector and a tensor. Many authors use the vector transformation relation in Eq. (13.10) as a *definition* of the term "**vector**" for ordinary 3D physical space. They say that a vector is any set of three numbers that are defined with respect to a basis in such a way that the numbers change according to Eq. (13.11) upon a change of basis. In this document, we are more precise by saying that these three numbers are *components* of a vector. The vector itself is defined to be the sum of the components times the base vectors; this sum is invariant upon a change of basis. This definition of "vector" eliminates the ambiguity associated with the sloppy definition of a vector as "something with length and direction."

On page 230, we explain that a mathematician will likely puff up proudly and define a vector to be a "member of a vector space." This pompous definition is not circular so long as the phrase "vector space" has already been defined. Specifically, as discussed on page 230, a vector space is a collection of objects that satisfy certain properties under addition and multiplication. The definition of a vector according to its transformation properties is merely a *discerning criterion* for identifying members of a particular candidate collection of objects. To be mathematically justified in using our transformation definition of a vector, we must follow through and test that vector addition and scalar multiplication do indeed satisfy the mathematician's required axioms (they are indeed satisfied). The mathematician's definition of a vector requires scalars (which they call members of a *field*) to be defined separately. Formally, a scalar is a zero-order tensor and a vector is a first-order tensor.

Engineering texts typically define second-order tensors to be ordered 3×3 arrays of numbers referenced to a basis such that those numbers change according to Eq. (13.12) upon a change of basis. In this document, we call the numbers themselves the *components* of the tensor. We define the tensor itself to be the sum of components times basis dyads, which is invariant under a change of basis.

Basis coupling tensor. Recall that

$$Q_{ij} = \tilde{\underset{\sim}{\boldsymbol{e}}}_i \bullet \hat{\underset{\sim}{\boldsymbol{e}}}_j \tag{13.23}$$

We can always construct a basis-coupling tensor $\underset{\approx}{\boldsymbol{Q}}$ as

$$\underset{\approx}{\boldsymbol{Q}} = \hat{\underset{\sim}{\boldsymbol{e}}}_k \tilde{\underset{\sim}{\boldsymbol{e}}}_k \tag{13.24}$$

We may write

$$\underset{\approx}{\boldsymbol{Q}} = \delta_{ij} \hat{\underset{\sim}{\boldsymbol{e}}}_i \hat{\underset{\sim}{\boldsymbol{e}}}_j = \delta_{ij} \tilde{\underset{\sim}{\boldsymbol{e}}}_i \hat{\underset{\sim}{\boldsymbol{e}}}_j = Q_{ij} \tilde{\underset{\sim}{\boldsymbol{e}}}_i \tilde{\underset{\sim}{\boldsymbol{e}}}_j = Q_{ij} \hat{\underset{\sim}{\boldsymbol{e}}}_i \hat{\underset{\sim}{\boldsymbol{e}}}_j \tag{13.25}$$

It follows that

$$\hat{\underset{\sim}{\boldsymbol{e}}}_k = \underset{\approx}{\boldsymbol{Q}} \bullet \tilde{\underset{\sim}{\boldsymbol{e}}}_k \tag{13.26}$$

or, equivalently,

$$\hat{\underset{\sim}{\boldsymbol{e}}}_i = Q_{ji} \tilde{\underset{\sim}{\boldsymbol{e}}}_j = Q_{ij}^T \tilde{\underset{\sim}{\boldsymbol{e}}}_j \tag{13.27}$$

We have *not* accidentally inserted a transpose in the last equation here. The basis coupling tensor is a linear transformation that operates on the tilde basis to produce the hat basis. The basis coupling tensor corresponds to a rigid rotation of space and all vectors in space. By contrast, the component transformations we have discussed in this chapter deal with holding space and all vectors fixed while we reorient the *observer*. You would use the basis coupling tensor if you wanted to transform material vectors so that they are aligned with the reoriented basis.

"When I'm working on a problem, I never think about beauty. I think only how to solve the problem. But, when I have finished, if the solution is not beautiful, I know it's wrong."

— Buckminster Fuller

14. Tensor (and Tensor function) invariance

This section may be skipped, but it does provide some useful insight into what is meant by a tensor.

What's the difference between a matrix and a tensor?

This is one of the most common questions for new students of tensor analysis. The answer comes down to the fact that tensor matrices must correspond to components with respect to an underlying basis, and the component matrix must change according to the tensor transformation rules of the previous section upon a change of basis. If this does not hold true, then the matrix is just a matrix, not a tensor component matrix.

Suppose that a tensor is defined by giving a "*rule*" for constructing its components with respect to any convenient orthonormal basis. We use the term "**tensor invariance**" to mean that the following:

> If you apply the "rule" in the two bases separately, then the two results must be related to each other by the transformation rule appropriate for the order of the result.

Stated in a slightly different way, if you want to know the result of the "rule" in a second basis, you can do one of the following:

- You can apply the rule in the first basis and then transform the result to the second basis.
- You can *first* transform the *arguments* of the rule to the second basis, and then apply the rule directly in the second basis.

Either way, you should obtain the same result. If you don't, then the "rule" violates tensor invariance.

When we talk about *tensor* invariance or when we use the phrase "valid *tensor* rule," we are using the word *tensor* in its generalized sense for which scalars are regarded as 0-order tensors, vectors are 1^{st} order tensors, etc. Therefore, the above statements also apply for the question "*What is the difference between a vector and a set of three numbers*?" Suppose your next door neighbor told you that (for whatever reason) he was working with a set of three numbers defined to be two times the index of the number. Then the first number would be 2, the second 4, and the third 6. He would have an array of three numbers given by $\{2, 4, 6\}$. To determine if your neighbor is actually working with a *vector*, you would first have to ask him if the indices of his rule had anything to do with a particular set of directions. Since this tutorial limits its scope to *orthonormal* bases, we will presume that he told you that the three directions corresponded to east, north, and up (which are orthogonal). You would then ask your neighbor if his index doubling rule would still apply if he were to instead use three *different* orthogonal reference directions such as north-east, north-west, and up. If his answer is "*yes, the doubling rule applies for any set of orthonormal reference directions*," then you would conclude that he is merely working with an array of three numbers, not a vector. To prove that your neighbor's doubling rule does not correspond to a vector, you would only have to find one counter-example that violates the invariance criterion. Let $\{\underset{\sim}{e}_1, \underset{\sim}{e}_2, \underset{\sim}{e}_3\}$ denote the directions $\{\text{east, north, up}\}$. Now let $\{\underset{\sim}{E}_1, \underset{\sim}{E}_2, \underset{\sim}{E}_3\}$ denote the directions $\{\text{north-east, north-west, up}\}$. In other words, let

$$\underset{\sim}{E}_1 = \frac{\underset{\sim}{e}_1 + \underset{\sim}{e}_2}{\sqrt{2}}, \qquad \underset{\sim}{E}_2 = \frac{-\underset{\sim}{e}_1 + \underset{\sim}{e}_2}{\sqrt{2}}, \qquad \underset{\sim}{E}_3 = \underset{\sim}{e}_3 \tag{14.1}$$

Then, the doubling rule in this system would give a vector

$$2\underset{\sim}{E}_1 + 4\underset{\sim}{E}_2 + 6\underset{\sim}{E}_3 \tag{14.2}$$

or

$$\frac{-4\underset{\sim}{e}_1 + 6\underset{\sim}{e}_2}{\sqrt{2}} + 6\underset{\sim}{e}_3 \tag{14.3}$$

Thus, the doubling rule applied directly in the second system does *not* result in the same vector as would be obtained by applying that rule in the first system from the outset (namely, $2\underset{\sim}{e}_1 + 4\underset{\sim}{e}_2 + 6\underset{\sim}{e}_3$), and you must therefore conclude that your neighbor's rule gives a well-defined *array*, but (alas) not a vector.

Suppose, on the other hand, that your neighbor had told you that the doubling rule was never intended to apply in a different system. You would then have to ask him to provide you with a general rule that applies in *any* system. If the general rule has the property that you will end up with the same vector, no matter what system you use, then his general rule would correspond to a vector, not just an array. In general, if you have a rule that gives you three numbers v_1, v_2, v_3 with respect to $\{\underset{\sim}{e}_1, \underset{\sim}{e}_2, \underset{\sim}{e}_3\}$ and the definition of the rule is such that it gives you three *different* numbers V_1, V_2, V_3 when applied to any other orthonormal triad $\{\underset{\sim}{E}_1, \underset{\sim}{E}_2, \underset{\sim}{E}_3\}$, then the rule corresponds to a vector if and only if

$$v_1\underset{\sim}{e}_1 + v_2\underset{\sim}{e}_2 + v_3\underset{\sim}{e}_3 = V_1\underset{\sim}{E}_1 + V_2\underset{\sim}{E}_2 + V_3\underset{\sim}{E}_3 \tag{14.4}$$

In other words, even though the *components* will generally vary upon a change of basis, the *sum* of components times base vectors does not change. That's why the term *invariance* is used. It says that a *vector* (the sum of components times basis) is invariant under a change of basis. If the basis changes, then the components must change in a *fully compensating manner* so that Eq. (14.4) remains true. For second-order tensors, a similar statement applies. Namely, the set of *nine* basis dyads ($\underset{\sim}{e}_1\underset{\sim}{e}_1, \underset{\sim}{e}_1\underset{\sim}{e}_2, \underset{\sim}{e}_1\underset{\sim}{e}_3, \underset{\sim}{e}_2\underset{\sim}{e}_1, \ldots, \underset{\sim}{e}_3\underset{\sim}{e}_3$) forms a basis for second-order tensors. A set of nine numbers $(a_{11}, a_{12}, a_{13}, a_{21}, \ldots, a_{33})$ corresponds to a second-order tensor if and only if the rule that assigns these numbers changes upon a change of basis in such a manner that

$$a_{ij}\underset{\sim}{e}_i\underset{\sim}{e}_j = A_{ij}\underset{\sim}{E}_i\underset{\sim}{E}_j \tag{14.5}$$

where $\{\underset{\sim}{E}_1, \underset{\sim}{E}_2, \underset{\sim}{E}_3\}$ is any other (orthonormal) basis and $(A_{11}, A_{12}, A_{13}, A_{21}, \ldots, A_{33})$ are the nine new numbers resulting from applying the rule in the new basis. If the above equation does not hold true, then the rule, and any numbers it spits out, do not correspond to a tensor.

Thankfully, the triplets and ordered 3×3 arrays that pop up in physics typically *do* satisfy the vector and tensor invariance requirements of Eqs. (14.4) and (14.5). That's no surprise, since any physical phenomenon should be unaffected by the coordinate frame that we adopt to describe it. The components do not mean much by themselves, but the *sum* of components times bases (being invariant) *does* have strong physical significance. It is the sum that means something as a *single* entity. Even though describing a velocity requires supplying three numbers, you'll never hear a person say "the velocity *are* …". Being a vector, velocity *is* (and should be) regarded as a single entity. Physical phenomena are routinely described conceptually using direct notation. For particular problems, or in computer codes, component expressions are used.

Whenever a "rule" is proved to satisfy invariance, we always endow it with a structured *direct* (Gibbs) notation. Whenever a new operation is defined exclusively in terms of invariant operations, then the new operation is immediately known to be *itself* invariant.

Any expression that can be written in direct notation is automatically known to be invariant itself.

Example of a "scalar rule" that satisfies tensor invariance

Given a vector $\underset{\sim}{v}$ with components $\{\hat{v}_1, \hat{v}_2, \hat{v}_3\}$ with respect to a basis $\{\hat{\underset{\sim}{e}}_1, \hat{\underset{\sim}{e}}_2, \hat{\underset{\sim}{e}}_3\}$, we propose constructing a scalar by applying the following "rule"

$$s = \hat{v}_1^2 + \hat{v}_2^2 + \hat{v}_3^2 \tag{14.6}$$

or, using summation notation,

$$s = \hat{v}_i\hat{v}_i \tag{14.7}$$

Now consider a second basis $\{\tilde{\underset{\sim}{e}}_1, \tilde{\underset{\sim}{e}}_2, \tilde{\underset{\sim}{e}}_3\}$. If we apply the "rule" directly in this second basis, we get

$$s = \tilde{v}_i\tilde{v}_i \tag{14.8}$$

To demonstrate that the "rule" satisfies tensor invariance, we must demonstrate that Eqs. (14.7) and (14.8) give the *same answer* for the scalar s. Recalling Eq. (13.11) we know there exists an orthogonal matrix [Q] such that

$$\hat{v}_j = \tilde{v}_i Q_{ij} \tag{14.9}$$

Substituting this into Eq. (14.7), being extra careful to introduce two distinct dummy summation indices, gives

$$s = \tilde{v}_m Q_{mi} \tilde{v}_k Q_{ki} \tag{14.10}$$

Since $[Q]$ is orthogonal, we know that $Q_{mi} Q_{ki} = \delta_{mk}$. Hence

$$s = \tilde{v}_m \delta_{mk} \tilde{v}_k = \tilde{v}_m \tilde{v}_m \tag{14.11}$$

which is identical to Eq. (14.8). Thus, this "rule" for constructing the scalar s satisfies tensor invariance. Now that we know that this rule is invariant, we are permitted to give it a *direct notation symbol*. Of course, the symbol for this operation is already defined — it is simply the dot product:

$$s = \underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{v}} \tag{14.12}$$

Example of a "scalar rule" that violates tensor invariance

Scalars *themselves* are invariant under a coordinate transformation. Newcomers to tensor analysis therefore often jump to the conclusion that *any* "rule" that produces a real number must be invariant. This section shows why this is not the case. If you look back at our definition of an invariant rule, you will see that you must be able to apply the rule in any two coordinate systems, and the results must differ from each other by the transformation between the systems. If the rule produces a scalar, then invariance demands that both systems must produce the *same* value for the scalar.

Given a vector $\underset{\sim}{\boldsymbol{v}}$ with components $\{v_1, v_2, v_3\}$ with respect to a basis $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$, let's consider the following "scalar-valued rule:"

$$s = v_1 + v_2 + v_3, \tag{14.13}$$

In other words, the scalar is constructed by simply adding up the vector's components. To show that this rule is not invariant, all we need is a *counterexample*. Consider a vector

$$\underset{\sim}{\boldsymbol{v}} = 3\tilde{\underset{\sim}{\boldsymbol{e}}}_1 + 4\tilde{\underset{\sim}{\boldsymbol{e}}}_2 \tag{14.14}$$

Consider a second basis

$$\hat{\underset{\sim}{\boldsymbol{e}}}_1 = \frac{1}{5}(3\tilde{\underset{\sim}{\boldsymbol{e}}}_1 + 4\tilde{\underset{\sim}{\boldsymbol{e}}}_2), \qquad \hat{\underset{\sim}{\boldsymbol{e}}}_2 = \frac{1}{5}(-4\tilde{\underset{\sim}{\boldsymbol{e}}}_1 + 3\tilde{\underset{\sim}{\boldsymbol{e}}}_2), \qquad \text{and} \qquad \hat{\underset{\sim}{\boldsymbol{e}}}_3 = \tilde{\underset{\sim}{\boldsymbol{e}}}_3 \tag{14.15}$$

In terms of this second basis, the vector $\underset{\sim}{\boldsymbol{v}}$ is just

$$\underset{\sim}{\boldsymbol{v}} = 5\hat{\underset{\sim}{\boldsymbol{e}}}_1 \tag{14.16}$$

Now, if we apply the rule of Eq. (14.13) to Eq. (14.14) we obtain $s = 7$. However, if we apply the rule to Eq. (14.16), we get $s = 5$. The final results do not agree! Consequently, summation of the components of a vector is not an invariant operation. This is an interesting conclusion since we found in the previous section that summation of the *squares* of components *is* invariant.

Example of a 3x3 matrix that does *not* correspond to a tensor

If you look up the definition of the term "tensor," many textbooks (especially older engineering books) will say that a tensor is a 3×3 matrix of components referenced to a basis such that the components will change in a particular manner upon a change of basis. Specifically, the components must change according to the basis transformation rule described Eq. (13.13). To help clarify this concept, let's present something that can be defined as a 3×3 matrix, but which is *not* the component matrix for any tensor.

In rigid body dynamics, the moments of inertia of a body Ω describe the amount of resistance the body has against being rotated. For example, the moment of inertia ψ_{zz} is defined by

$$\psi_{zz} = \iiint_{\Omega} (x^2 + y^2)\rho dV \qquad (14.17)$$

Here, $\{x, y, z\}$ are the Cartesian coordinates of a point in the body. Thus, the quantity $(x^2 + y^2)$ is the square of the distance of a point from the z axis. Intuitively, the farther points are from the z axis, the harder it is to rotate those points body about the z axis. Hence, ψ_{zz} is a good measure of the resistance of the body to rotation about the z axis. The moments of inertia about the other two axes are defined similarly:

$$\psi_{xx} = \iiint_{\Omega} (y^2 + z^2)\rho dV \qquad (14.18)$$

$$\psi_{yy} = \iiint_{\Omega} (z^2 + x^2)\rho dV \qquad (14.19)$$

The *products* of inertia are defined in many books by

$$\psi_{xy} = \iiint_{\Omega} (xy)\rho dV, \qquad \psi_{yz} = \iiint_{\Omega} (yz)\rho dV, \qquad \text{and} \qquad \psi_{zx} = \iiint_{\Omega} (zx)\rho dV \qquad (14.20)$$

Though less intuitive, these quantities measure how much "wobble" you would feel when rotating the body about an axis.

It would *seem* natural to arrange these inertias into a 3×3 matrix as

$$[\psi] = \begin{bmatrix} \psi_{xx} & \psi_{xy} & \psi_{xz} \\ \psi_{yx} & \psi_{yy} & \psi_{yz} \\ \psi_{zx} & \psi_{zy} & \psi_{zz} \end{bmatrix} \qquad (14.21)$$

Here, we have defined $\psi_{yx}=\psi_{xy}$, etc. to construct a symmetric matrix. It is perfectly legitimate for us to arrange the inertias into a matrix form. The question arises, however:

QUESTION: Does the $[\Psi]$ matrix correspond to a tensor?
ANSWER: No, it does not!

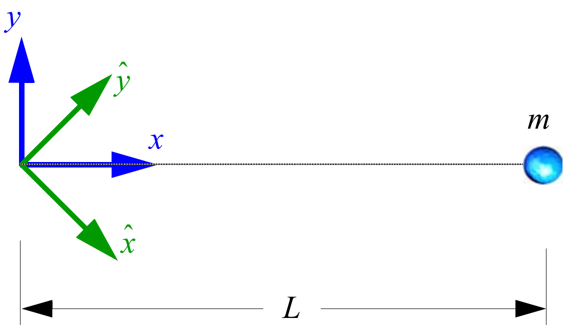
Whenever you wish to prove that something is *false*, all you need is a *counterexample*. Consider a body Ω that is a tiny ball of mass m located on the x axis at a distance L from the origin. Treating the ball as a point mass so that $x \approx L$, $y \approx z \approx 0$, the inertia values are

$$\psi_{yy} = \psi_{zz} = mL^2 \text{ and all other } \psi_{ij} \approx 0 \qquad (14.22)$$

Thus,

$$[\psi] = mL^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{w.r.t. the } \{xyz\} \text{ coordinate system.} \qquad (14.23)$$

Now consider a different "hat" coordinate system as shown. In this coordinate system, $\hat{x} = \hat{y} \approx L/\sqrt{2}$ and $z \approx 0$. Applying the definitions of the inertia components in this hat system gives



$$\psi_{\hat{x}\hat{x}} = \psi_{\hat{y}\hat{y}} = \frac{1}{2}mL^2 \qquad (14.24)$$

$$\psi_{zz} = mL^2 \qquad (14.25)$$

$$\psi_{xy} = \frac{1}{2}mL^2 \qquad (14.26)$$

Thus,

$$[\psi] = mL^2 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{w.r.t. the } \{\hat{x}\hat{y}\hat{z}\} \text{ coordinate system.} \qquad (14.27)$$

Now the question becomes: *would we have obtained Eq. (14.27) if we had merely performed a coordinate transformation of Eq. (14.23)?* Let's show that the answer is "no." The direction cosine matrix for the transformation from the $\{xyz\}$ system to the $\{\hat{x}\hat{y}\hat{z}\}$ system is

$$[Q] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (14.28)$$

If we assume (incorrectly) that the $[\psi]$ matrix corresponds to a tensor, then the following relationship *should* be true

$$[\psi]_{\substack{\text{w.r.t.} \\ \{\hat{x}\hat{y}\hat{z}\}}} = [Q]^T [\psi]_{\substack{\text{w.r.t.} \\ \{xyz\}}} [Q] \qquad (14.29)$$

Thus, if $[\psi]$ corresponds to a tensor,* then it *should* be true that

$$\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textit{This equation is false!} \qquad (14.30)$$

Multiplying this out, however, shows that equality does *not* hold. Consequently, this counterexample proves that the matrix of inertias defined in Eq. (14.21) is *not* the matrix associated with a tensor!

The inertia TENSOR

Now we are left with the question: *Is there any way to organize the inertias into a matrix form that* <u>*does*</u> *correspond to a tensor?* The answer to this question is "yes!" Simply putting *negatives* on the off-diagonals turns out to define a properly invariant tensor matrix:

$$[\Phi] = \begin{bmatrix} \Psi_{xx} & -\Psi_{xy} & -\Psi_{xz} \\ -\Psi_{yx} & \Psi_{yy} & -\Psi_{yz} \\ -\Psi_{zx} & -\Psi_{zy} & \Psi_{zz} \end{bmatrix} \qquad (14.31)$$

This new definition corresponds to a tensor if and only if, for any two Cartesian coordinate systems,

$$[\Phi]_{\substack{\text{w.r.t.} \\ \{\hat{x}\hat{y}\hat{z}\}}} = [Q]^T[\Phi]_{\substack{\text{w.r.t.} \\ \{xyz\}}}[Q], \qquad \textit{Must hold true for all [Q].} \qquad (14.32)$$

where $[Q]$ is the direction cosine matrix.

Let's first check whether placing negatives on the off-diagonals fixes our previous counterexample. With our new definition for the inertia tensor, we now ask whether negatives on the off-diagonals of the inertia tensor in Eq. (14.30) will make that equation true. In other words, is the following matrix equation true?

$$\begin{bmatrix} 1/2 & -1/2 & 0 \\ -1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textit{This equation is true!} \qquad (14.33)$$

Multiplying this out shows that equality holds. Passing this test is necessary, *but not sufficient* for Eq. (14.32) to hold. The fact that one example (i.e., one choice for [Q]) worked out okay does not prove that our new definition really does correspond to a tensor. We don't know for sure (yet) whether or not some other counterexample might prove that our new definition isn't a tensor either. It is always harder to prove something *true* than to prove it false!

[*] Keep in mind: we are trying to prove that it doesn't.

It turns out that our new definition does indeed transform properly for all possible coordinate changes. One of the simplest ways to prove that a matrix construction rule is in fact a properly invariant *tensor* rule is to find some way to write out the definition in *direct* (Gibbs) notation using only operations (such as the dot product and dyadic multiplication) that have already been proved to be valid tensor operations.

We recognize that the expression $x^2 + y^2$ that appears in the definition of ψ_{zz} can be written as $(x^2 + y^2 + z^2) - z^2$, or simply $\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}} - z^2$, where the position vector is $\underset{\sim}{\boldsymbol{x}} = x\underset{\sim}{\boldsymbol{e}}_x + y\underset{\sim}{\boldsymbol{e}}_y + z\underset{\sim}{\boldsymbol{e}}_z = \hat{x}\underset{\sim}{\boldsymbol{e}}_{\hat{x}} + \hat{y}\underset{\sim}{\boldsymbol{e}}_{\hat{y}} + \hat{z}\underset{\sim}{\boldsymbol{e}}_{\hat{z}}$. This observation lets us introduce at least a *bit* of direct notation into the formula. Note that the off-diagonals of the dyad $-\underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}$ are identical to the off-diagonals of our new inertia matrix. The diagonal components of $-\underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}$ are $-x^2$, $-y^2$, and $-z^2$, which are exactly the same as the terms we used to rewrite the diagonals of the new inertia matrix. For example, the zz component $x^2 + y^2$ was re-written as $\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}} - z^2$. With a bit of thought, we therefore recognize that the inertia tensor may be written in the following clean direct notation:

$$\underset{\approx}{\Phi} = \iiint_{\Omega} [(\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}})\underset{\approx}{\boldsymbol{I}} - \underset{\sim}{\boldsymbol{x}}\underset{\sim}{\boldsymbol{x}}]\rho dV \tag{14.34}$$

By discovering this *direct notation* formula, we have relieved ourselves of the burden of proving invariance under coordinate transformations. The direct notation operations *themselves* (namely: dyadic multiplication, tensor addition, scalar multiplication, vector dot product) have all been individually proven invariant in separate analyses. Any object constructed using these invariant operations must *itself* be invariant!

> *"Engineers, like artists, often fall in love with their models."* — *Tom Bement*

15. Scalar invariants and spectral analysis

Invariants of vectors or tensors

A scalar invariant of a vector or tensor is any scalar-valued function of the vector or tensor that gives the same result even upon a change in basis.

The sum $v_1 + v_2 + v_3$ is *not* an invariant of a vector $\underset{\sim}{v}$ because the sum $\tilde{v}_1 + \tilde{v}_2 + \tilde{v}_3$ with will generally give a different result when computed with respect to a different basis. However, the sum of the *squares* is an invariant, as was proved on page 179.

The sum $T_{11}^2 + T_{22}^2 + T_{33}^2$ of the squares of the *diagonal* components of a tensor, $\underset{\approx}{T}$ is *not* an invariant of a tensor because the result is generally different for different bases. However, the sum of the diagonal components $T_{11} + T_{22} + T_{33}$ alone *does* turn out to be an invariant because, for any orthogonal Q_{ij} direction cosine matrix,

$$\begin{aligned} T_{11} + T_{22} + T_{33} \quad & = T_{kk} \\ & = \tilde{T}_{ij} Q_{ik} Q_{jk} \qquad \text{(now use orthogonality)} \\ & = \tilde{T}_{ij} \delta_{ij} \\ & = \tilde{T}_{ii} \\ & = \tilde{T}_{11} + \tilde{T}_{22} + \tilde{T}_{33} \end{aligned} \tag{15.1}$$

Primitive invariants

As mentioned earlier, the magnitude of a vector is an invariant. Likewise, the *square* of the magnitude is also an invariant. These two invariants are not independent. One can always be computed from the other. In general, if α and β are invariants, then any scalar-valued function of them will also be invariant itself.

Primitive invariants of a vector or tensor are any minimal set of invariants such that all *other* invariants may be expressed as functions of the primitive invariants.

A vector has only one primitive invariant — its magnitude. *Symmetric* tensors (of class V_3^2) have three primitive invariants. This follows because symmetric tensors have three eigenvalues. Since an invariant may be computed in any basis with the same result, all invariants of symmetric tensors must be expressible as functions of the tensor's eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$. For example, the magnitude of a symmetric tensor is an invariant that may be written as $\lambda_1^2 + \lambda_2^2 + \lambda_3^2$. *Nonsymmetric* tensors have more than three primitive invariants. For nonsymmetric tensors, the magnitude is itself an independent invariant that cannot be expressed as a function of the eigenvalues. To prove this statement, consider a tensor whose component matrix with respect to some particular basis is given by

$$\begin{bmatrix} \lambda_1 & a & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{15.2}$$

The eigenvalues are $\{\lambda_1, \lambda_2, \lambda_3\}$, but the tensor magnitude is $\lambda_1^2 + \lambda_2^2 + \lambda_3^2 + a^2$, which depends on the "12" component W. Different values of a will give different magnitudes. Hence, the magnitude must not be expressible as a function of the eigenvalues. Tensor magnitude is an independent fourth invariant for nonsymmetric tensors! Don't let anyone tell you that a tensor is zero if all of its eigenvalues are zero — that statement is true only for *symmetric* tensors!

Trace invariants

Three popular invariants of a second-order tensor are

$$\begin{aligned} I &= \mathrm{tr}\underset{\approx}{\boldsymbol{T}} \\ II &= \mathrm{tr}(\underset{\approx}{\boldsymbol{T}}^2) \\ III &= \mathrm{tr}(\underset{\approx}{\boldsymbol{T}}^3) \end{aligned} \tag{15.3}$$

For symmetric tensors, these form an acceptable set of primitive invariants. A *symmetric* tensor is zero if and only if these three invariants are all zero. This statement if false for nonsymmetric tensors, as can be readily verified by putting $\lambda_1 = \lambda_2 = \lambda_3 = 0$ and $a = 1$ in Eq. (15.2). If the eigenvalues are known, then the trace invariants are given by

$$\begin{aligned} I &= \lambda_1 + \lambda_2 + \lambda_3 \\ II &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ III &= \lambda_1^3 + \lambda_2^3 + \lambda_3^3 \end{aligned} \tag{15.4}$$

Characteristic invariants

The characteristic invariants of a second-order tensor $\underset{\approx}{\boldsymbol{T}}$ are defined by

$$I_k = \text{the sum of all possible } k \times k \text{ principal subminors} \tag{15.5}$$

Here, a $k \times k$ principal subminor is the determinant of a matrix formed by striking out rows and columns of the component matrix $[\underset{\approx}{T}]$ until all that is left is a $k \times k$ matrix whose diagonal components are also diagonal components of the original tensor $[\underset{\approx}{T}]$. The I_k principal invariant requires you to find all of the possible ways to construct a $k \times k$ principal subminor and to sum their determinants.

For a tensor referenced to 3D space, the component matrix is 3×3 and the characteristic invariants are

$$\begin{aligned} I_1 &= T_{11} + T_{22} + T_{33} \\ I_2 &= \det\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} + \det\begin{bmatrix} T_{22} & T_{23} \\ T_{32} & T_{33} \end{bmatrix} + \det\begin{bmatrix} T_{11} & T_{13} \\ T_{31} & T_{33} \end{bmatrix} \\ I_3 &= \det\begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \end{aligned} \tag{15.6}$$

These are called the characteristic invariants (or principal invariants [24]) because they are coefficients in the characteristic equation that gives the eigenvalues of $\underset{\approx}{T}$. Namely, the eigenvalues of a tensor $\underset{\approx}{T}$ are the solutions of

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0 \tag{15.7}$$

This equation is true even if the tensor $\underset{\approx}{T}$ is nonsymmetric. For *symmetric* tensors, the characteristic invariants are an acceptable set of primitive invariants if the tensor is symmetric. In other words, all other invariants of a symmetric tensor may be expressed as functions of the characteristic invariants. A *symmetric* tensor is zero if and only if its characteristic invariants are all zero. This statement if false for nonsymmetric tensors, as can be readily verified by putting $\lambda_1 = \lambda_2 = \lambda_3 = 0$ and $a = 1$ in Eq. (15.2).

If the eigenvalues of the tensor are known, then the characteristic invariants are given by

$$\begin{aligned} I_1 &= \lambda_1 + \lambda_2 + \lambda_3 \\ I_2 &= \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1 \\ I_3 &= \lambda_1\lambda_2\lambda_3 \end{aligned} \tag{15.8}$$

Comparing with Eq. (15.4), note that the characteristic invariants are related to the trace invariants by

$$\begin{aligned} \mathrm{tr}(\underset{\approx}{\boldsymbol{T}}) &= I_1 \\ \mathrm{tr}(\underset{\approx}{\boldsymbol{T}}^2) &= I_1^2 - 2I_2 \\ \mathrm{tr}(\underset{\approx}{\boldsymbol{T}}^3) &= I_1^3 - 3I_1I_2 + 3I_3 \end{aligned} \tag{15.9}$$

These relationships hold only for second-order tensors referenced to a 3D space (class V_3^2). However, the basic definitions of the characteristic invariants and their relationship with the characteristic equation extend analogously to other dimensions. For second-order tensors referenced to a 4D space (i.e., class V_4^2 tensors, whose component matrices are 4×4), the characteristic equation is $\lambda^4 - I_1\lambda^3 + I_2\lambda^2 - I_3\lambda + I_4 = 0$. For second-order tensors referenced to a 2D space (class V_2^2, for which the component matrices are dimensioned 2×2), the characteristic equation is

$$\lambda^2 - I_1\lambda + I_2 = 0\,, \qquad \text{where} \qquad I_1 = T_{11} + T_{22} \qquad \text{and} \qquad I_2 = \det\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \tag{15.10}$$

Applying this formula is the fastest and least error-prone way to quickly write down the characteristic equation of a 2×2 matrix.

Direct notation definitions of the characteristic invariants.

The invariant definitions that we have given so far demand that the underlying basis must be orthonormal. Some more advanced texts (e.g., [21]) cite more abstract, and therefore more broadly applicable, definitions of the characteristic invariants. These definitions are phrased in terms of the triple scalar product, $[(\ ),(\ ),(\ )]$, defined in Eq. (5.36). Namely, in 3D, I_1, I_2, and I_3 are defined such that, *for any vectors,* $\underset{\sim}{\mathbf{u}}$*,* $\underset{\sim}{\mathbf{v}}$*, and* $\underset{\sim}{\mathbf{w}}$*,*

$$[\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}]+[\underset{\sim}{\mathbf{u}},\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}]+[\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{v}}] = I_1\ [\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}] \tag{15.11}$$

$$[\underset{\sim}{\mathbf{u}},\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{v}},\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{w}}]+[\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{u}}\ ,\underset{\sim}{\mathbf{v}},\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{w}}]+[\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{u}},\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}] = I_2\ [\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}] \tag{15.12}$$

$$[\underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{u}}\ ,\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{v}},\ \underset{\approx}{\mathbf{T}}\bullet\underset{\sim}{\mathbf{w}}] = I_3\ [\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}] \tag{15.13}$$

Note that the left hand side in the equation for I_1 sums over every possible way the tensor $\underset{\approx}{\mathbf{T}}$ can operate on exactly *one* of the vectors in the triple scalar product. Similarly, I_2 involves sums over every possible way the tensor can act on *two* of the vectors and I_3 involves sums over every way (only one way) for the tensor to act on all *three* vectors in the triple scalar product.

Admittedly, these definitions are rather strange looking, and we have omitted the proof that the right hand side can, in every instance simplify to a scalar (the invariant) times the triple scalar product $[\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}]$. Nonetheless, expressions like these show up occasionally in analyses, and it is essential to recognize how beautifully they simplify. Furthermore, these definitions are a nice place to start when attempting to deduce how to compute the tensor invariants for irregular (non-orthonormal) bases.

The cofactor in the triple scalar product. We close this discussion with a final identity, involving the cofactor tensor. Recall the direct notation definition of the cofactor, given in Eq. (9.46):

$$\underset{\approx}{\mathbf{G}}^C\bullet(\underset{\sim}{\mathbf{u}}\times\underset{\sim}{\mathbf{v}}) = (\underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{u}})\times(\underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{v}}) \qquad \text{for all vectors } \underset{\sim}{\mathbf{u}} \text{ and } \underset{\sim}{\mathbf{v}} \tag{15.14}$$

Dotting both sides of this equation by a third arbitrary vector $\underset{\sim}{\mathbf{w}}$ gives

$$[\underset{\approx}{\mathbf{G}}^C\bullet(\underset{\sim}{\mathbf{u}}\times\underset{\sim}{\mathbf{v}})]\bullet\underset{\sim}{\mathbf{w}} = [(\underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{u}})\times(\underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{v}})]\bullet\underset{\sim}{\mathbf{w}} \tag{15.15}$$

The right-hand side is the triple scalar product $[\underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{u}},\underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{v}},\underset{\sim}{\mathbf{w}}]$. Using Eq. (8.17), The left-hand side can be written $[(\underset{\sim}{\mathbf{u}}\times\underset{\sim}{\mathbf{v}})\bullet\underset{\approx}{\mathbf{G}}^{CT}]\bullet\underset{\sim}{\mathbf{w}}$ or $(\underset{\sim}{\mathbf{u}}\times\underset{\sim}{\mathbf{v}})\bullet[\underset{\approx}{\mathbf{G}}^{CT}\bullet\underset{\sim}{\mathbf{w}}]$, which is the triple scalar product $[\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\underset{\approx}{\mathbf{G}}^{CT}\bullet\underset{\sim}{\mathbf{w}}]$. Thus

$$\boxed{[\underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{u}},\ \underset{\approx}{\mathbf{G}}\bullet\underset{\sim}{\mathbf{v}},\ \ \underset{\sim}{\mathbf{w}}] = [\underset{\sim}{\mathbf{u}},\underset{\sim}{\mathbf{v}},\ \ \underset{\approx}{\mathbf{G}}^{CT}\bullet\underset{\sim}{\mathbf{w}}]} \tag{15.16}$$

This result shows that, when a tensor acts on *two* arguments in a triple scalar product, it can be recast as the transpose of the cofactor acting on the previously un-transformed (lonely) third vector. By the cyclic property of the triple scalar product, we can assert that this statement holds true when the tensor is acting on *any* two vectors in a triple scalar product.

Applying the above identity to the left-hand-side of Eq. (15.12), and then applying the definition of trace given in Eq. (15.11) shows that

$$\boxed{I_2 = \mathrm{tr}\,\underset{\approx}{\boldsymbol{T}}^c} \tag{15.17}$$

Of course, this result also follows directly from Eqs. (15.8) and (3.94).

Invariants of a sum of two tensors

Consider the sum $\underset{\approx}{\boldsymbol{T}} = \underset{\approx}{\boldsymbol{A}} + \underset{\approx}{\boldsymbol{B}}$ of two tensors. The first invariant is simple enough because the trace of a sum is the sum of the traces. Thus

$$I_1^{A+B} = \mathrm{tr}\underset{\approx}{\boldsymbol{A}} + \mathrm{tr}\underset{\approx}{\boldsymbol{B}} = I_1^A + I_1^B \tag{15.18}$$

Solving Eq. (15.9*b*) for I_2 gives a formula for the second invariant:

$$I_2 = \frac{1}{2}(I_1^2 - \mathrm{tr}(\underset{\approx}{\boldsymbol{T}}^2)) \tag{15.19}$$

When applied to $\underset{\approx}{\boldsymbol{T}} = \underset{\approx}{\boldsymbol{A}} + \underset{\approx}{\boldsymbol{B}}$, the result is

$$I_2^{A+B} = \frac{1}{2}((\mathrm{tr}\underset{\approx}{\boldsymbol{A}} + \mathrm{tr}\underset{\approx}{\boldsymbol{B}})^2 - \mathrm{tr}(\underset{\approx}{\boldsymbol{A}} + \underset{\approx}{\boldsymbol{B}})^2) \tag{15.20}$$

which simplifies to

$$I_2^{A+B} = I_2^A + I_2^B + I_1^A I_1^B - \mathrm{tr}(\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}) \tag{15.21}$$

Without proof, the determinant of a sum can be written

$$I_3^{A+B} = \det(\underset{\approx}{\boldsymbol{A}} + \underset{\approx}{\boldsymbol{B}}) = I_3^A + I_3^B + \underset{\approx}{\boldsymbol{A}}^c \boldsymbol{:} \underset{\approx}{\boldsymbol{B}} + \underset{\approx}{\boldsymbol{A}} \boldsymbol{:} \underset{\approx}{\boldsymbol{B}}^c \tag{15.22}$$

CASE: invariants of the sum of a tensor plus a dyad. Now suppose that the tensor $\underset{\approx}{\boldsymbol{B}}$ is actually a dyad:

$$\underset{\approx}{\boldsymbol{B}} = \underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}} \tag{15.23}$$

Then

$$\underset{\approx}{\boldsymbol{B}}^c = \underset{\approx}{\boldsymbol{0}} \tag{15.24}$$

$$I_1^B = \underset{\sim}{\boldsymbol{u}} \bullet \underset{\sim}{\boldsymbol{v}} \tag{15.25}$$

$$I_2^B = 0 \tag{15.26}$$

$$I_3^B = 0 \tag{15.27}$$

Thus, the invariants of $\underset{\approx}{\boldsymbol{T}} = \underset{\approx}{\boldsymbol{A}} + \underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}}$ are

$$I_1^T = \mathrm{tr}(\underset{\approx}{\boldsymbol{A}} + \underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}}) = \mathrm{tr}\underset{\approx}{\boldsymbol{A}} + \underset{\sim}{\boldsymbol{u}} \bullet \underset{\sim}{\boldsymbol{v}} \tag{15.28}$$

$$I_2^T = I_2^A + (\mathrm{tr}\underset{\approx}{\boldsymbol{A}})(\underset{\sim}{\boldsymbol{u}} \bullet \underset{\sim}{\boldsymbol{v}}) - \underset{\sim}{\boldsymbol{v}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{u}} \tag{15.29}$$

$$I_3^T = \det(\underset{\approx}{\mathbf{A}} + \underset{\sim}{\mathbf{u}}\underset{\sim}{\mathbf{v}}) = \det\underset{\approx}{\mathbf{A}} + \underset{\sim}{\mathbf{u}} \bullet \underset{\approx}{\mathbf{A}}^c \bullet \underset{\sim}{\mathbf{v}} \qquad (15.30)$$

The last equation is especially useful in the analysis of the acoustic tensor for determining plastic wave speeds and modes.

The Cayley-Hamilton theorem:

A tensor satisfies its own characteristic equation. That is,

$$\underset{\approx}{\boldsymbol{T}}^3 - I_1\underset{\approx}{\boldsymbol{T}}^2 + I_2\underset{\approx}{\boldsymbol{T}} - I_3\underset{\approx}{\boldsymbol{I}} = \underset{\approx}{\boldsymbol{0}}, \tag{15.31}$$

where $\underset{\approx}{\boldsymbol{I}}$ is the identity tensor, and $\underset{\approx}{\boldsymbol{0}}$ is the zero tensor. This theorem is trivial to prove if the tensor is symmetric, but the proof is quite subtle if the tensor is non-symmetric [see Ref. 26].

The above theorem provides a means of writing powers of a tensor in terms of second or lower powers. For example, multiplying Eq. (15.31) by $\underset{\approx}{\boldsymbol{T}}$ and solving for $\underset{\approx}{\boldsymbol{T}}^4$ gives

$$\begin{aligned} \underset{\approx}{\boldsymbol{T}}^4 &= I_1\underset{\approx}{\boldsymbol{T}}^3 - I_2\underset{\approx}{\boldsymbol{T}}^2 + I_3\underset{\approx}{\boldsymbol{T}} \\ &= I_1(I_1\underset{\approx}{\boldsymbol{T}}^2 - I_2\underset{\approx}{\boldsymbol{T}} + I_3\underset{\approx}{\boldsymbol{I}}) - I_2\underset{\approx}{\boldsymbol{T}}^2 + I_3\underset{\approx}{\boldsymbol{T}} \\ &= (I_1^2 - I_2)\underset{\approx}{\boldsymbol{T}}^2 + (I_3 - I_1 I_2)\underset{\approx}{\boldsymbol{T}} + I_1 I_3\underset{\approx}{\boldsymbol{I}} \end{aligned} \tag{15.32}$$

where, in the second step, we used Eq. (15.31) to substitute for $\underset{\approx}{\boldsymbol{T}}^3$.

CASE: Expressing the inverse in terms of powers and invariants.

We can multiply both sides of Eq. (15.31) by $\underset{\approx}{\boldsymbol{T}}^{-1}$ and then solve for $\underset{\approx}{\boldsymbol{T}}^{-1}$ to obtain

$$\underset{\approx}{\boldsymbol{T}}^{-1} = \frac{\underset{\approx}{\boldsymbol{T}}^2 - I_1\underset{\approx}{\boldsymbol{T}} + I_2\underset{\approx}{\boldsymbol{I}}}{I_3} \tag{15.33}$$

CASE: Expressing the cofactor in terms of powers and invariants.

Knowing that the inverse can also be written $\underset{\approx}{\boldsymbol{T}}^{-1} = \dfrac{\underset{\approx}{\boldsymbol{T}}^{CT}}{\det\underset{\approx}{\boldsymbol{T}}}$, the above result show us that

$$\underset{\approx}{\boldsymbol{T}}^{CT} = \underset{\approx}{\boldsymbol{T}}^2 - I_1\underset{\approx}{\boldsymbol{T}} + I_2\underset{\approx}{\boldsymbol{I}} \tag{15.34}$$

Eigenvalue problems

Consider a general tensor, $\underset{\approx}{\boldsymbol{A}}$. A vector $\underset{\sim}{\boldsymbol{p}}$ is called a "right" **eigenvector** of $\underset{\approx}{\boldsymbol{A}}$ if and only if $\underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{p}}$ is proportional to $\underset{\sim}{\boldsymbol{p}}$. In other words, there must exist a scalar λ, called the "**eigenvalue**," such that

$$\underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{p}} = \lambda\underset{\sim}{\boldsymbol{p}} \tag{15.35}$$

Eigenvectors are also called **principal directions**. Note that if an eigenvector can be found, its magnitude is arbitrary because any scalar multiple of an eigenvector would also be an eigenvector. Thus, if we say that an eigenvector is *unique*, we really mean that its *direction* is unique.

Rewriting Eq. (15.35), we seek $\underset{\sim}{\boldsymbol{p}}$ and λ such that

$$[\underset{\approx}{A} - \lambda\underset{\approx}{I}] \bullet \underset{\sim}{p} = \underset{\sim}{0} \tag{15.36}$$

The only way that this equation can hold is if the tensor in brackets is singular. Hence, λ must satisfy

$$\det[\underset{\approx}{A} - \lambda\underset{\approx}{I}] = 0 \tag{15.37}$$

Since we are speaking of ordinary tensors in ordinary 3D physical space, expansion of this determinant gives a cubic equation for the eigenvalue λ. Specifically, it can be shown that this "characteristic equation" is

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0 \tag{15.38}$$

where I_i are the invariants of Eq. (15.6). Once an eigenvalue is found, the corresponding eigenvector is determined by enforcing the definition of Eq. (15.35). For two-dimensional problems, a graphical tool called **Mohr's circle**, is also very useful for performing eigenvalue analyses.*

Rebecca: discuss possibility of complex eigenvalues

Because the equation is cubic, there are up to three distinct eigenvalues, $\{\lambda_1, \lambda_2, \lambda_3\}$. Hence there is the *potential* for having up to three associated eigenvectors, $\{\underset{\sim}{p}_1, \underset{\sim}{p}_2, \underset{\sim}{p}_3\}$, but there might be fewer (as explained below). Any linear algebra textbook contains the proof that the eigenvectors corresponding to distinct eigenvalues will be linearly independent. In other words,

If $\lambda_i \neq \lambda_j$, then $\underset{\sim}{p}_i$ and $\underset{\sim}{p}_j$ are linearly independent. (15.39)

For symmetric tensors, the following stronger statement holds

For **symmetric tensors**,
if $\lambda_i \neq \lambda_j$, then $\underset{\sim}{p}_i$ and $\underset{\sim}{p}_j$ are orthogonal. (15.40)

Since the magnitudes of eigenvectors are arbitrary, it is customary to scale them to unit length so that distinct eigenvectors for symmetric tensors are not only orthogonal, but also ortho*normal*. As discussed later, a different convention applies for non-symmetric tensors.

Algebraic and geometric multiplicity of eigenvalues

If all three eigenvalues are distinct (*i.e.,* if there are no repeated roots), then the three eigenvectors are linearly independent and may be used as a basis for the 3D physical space.

When two or more eigenvalues are equal, however, the situation becomes more complicated. The *algebraic multiplicity* of an eigenvalue is the number of times the eigenvalue repeats as a solution to Eq. (15.38). The *geometric multiplicity* of an eigenvalue is the number of linearly independent eigenvectors associated with the eigenvalue. There is

* An extremely detailed exposition on Mohr's circle, including its definition for *nonsymmetric* tensors is provided in Ref. [9]. More elementary discussions may be found in virtually any undergraduate text on mechanics of materials.

always at least one eigenvector associated with each eigenvalue. For repeat roots, the geometric multiplicity is always less than or equal to the algebraic multiplicity. The distinction between geometric and algebraic multiplicity is important only for nonsymmetric tensors because (as any good matrix analysis will prove),

For a real symmetric tensor, the geometric multiplicity always equals the algebraic multiplicity.

Consequently, a symmetric tensor will always have a complete set of linearly independent eigenvectors that may be used as a basis for 3D space. Such a basis is sometimes called an **eigenbasis**.

For repeated eigenvalues, the associated eigenvectors do not have unique directions. If the geometric multiplicity is two, then there are exactly two associated linearly independent eigenvectors. These two eigenvectors may be used to define a plane in 3D space. The eigenvectors associated with the eigenvalue of geometric multiplicity two are unique to the degree that they must lie in that plane, but their directions (and, of course, magnitudes) are arbitrary. If desired, they may be scaled such that they are any two orthonormal vectors in the plane. These statements apply even to nonsymmetric tensors. For symmetric tensors, in light of Eq. (15.40), we note that

A symmetric tensor always has an *orthonormal* eigenbasis. (15.41)

A normalized eigenbasis is often denoted with different symbols, $\{\underset{\sim}{\delta}_1, \underset{\sim}{\delta}_2, \underset{\sim}{\delta}_3\}$ to emphasize that it has been selected such that

$$\underset{\sim}{\delta}_i \bullet \underset{\sim}{\delta}_j = \delta_{ij} \tag{15.42}$$

Although not guaranteed, it may so happen that a particular *non*-symmetric tensor of interest also has all of its geometric multiplicities equal to the algebraic multiplicities. In such a case, that tensor also possesses a “spanning” set of eigenvectors $\{\underset{\sim}{\boldsymbol{p}}_1, \underset{\sim}{\boldsymbol{p}}_2, \underset{\sim}{\boldsymbol{p}}_3\}$ that may serve as a basis for 3D space. However,

If a *non*-symmetric tensor happens to have an eigenbasis,
then the eigenbasis will *not* be orthogonal. (15.43)

A non-orthogonal eigenbasis $\{\underset{\sim}{\boldsymbol{p}}_1, \underset{\sim}{\boldsymbol{p}}_2, \underset{\sim}{\boldsymbol{p}}_3\}$ always has a “dual” basis $\{\underset{\sim}{\boldsymbol{p}}^1, \underset{\sim}{\boldsymbol{p}}^2, \underset{\sim}{\boldsymbol{p}}^3\}$ that is defined such that

$$\underset{\sim}{\bar{\boldsymbol{p}}}^i \bullet \underset{\sim}{\boldsymbol{p}}_j = \delta^i_j , \tag{15.44}$$

where the δ^i_j is an alternative symbol for the Kronecker delta and $\underset{\sim}{\bar{\boldsymbol{p}}}^i$ is the complex conjugate of $\underset{\sim}{\boldsymbol{p}}^i$. Comparing Eqs. (15.44) with (15.42) shows that the dual basis associated with $\{\underset{\sim}{\delta}_1, \underset{\sim}{\delta}_2, \underset{\sim}{\delta}_3\}$ is just $\{\underset{\sim}{\delta}_1, \underset{\sim}{\delta}_2, \underset{\sim}{\delta}_3\}$ itself, so there is no need to worry about a dual basis for symmetric tensors.

Diagonalizable tensors (the spectral theorem)

Whenever a tensor possesses a spanning set of eigenvectors, $\{\underset{\sim}{\mathbf{p}}_1, \underset{\sim}{\mathbf{p}}_2, \underset{\sim}{\mathbf{p}}_3\}$ (*i.e.,* when a tensor has an eigenbasis), then the tensor is said to be "**diagonalizable**." In matrix analysis, this means that the matrix $[A]_{\mathbf{ee}}$ containing the components of $\underset{\approx}{\mathbf{A}}$ is with respect to the laboratory basis $\{\underset{\sim}{\mathbf{e}}_1, \underset{\sim}{\mathbf{e}}_2, \underset{\sim}{\mathbf{e}}_3\}$ is *similar* to a diagonal matrix $[\Lambda]$ containing the eigenvalues on the diagonal. In other words, there exists a matrix $[L]$ such that

$$[A]_{\mathbf{ee}} = [L][\Lambda][L]^{-1} \tag{15.45}$$

where

$$[\Lambda] = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{15.46}$$

Comparing Eq. (15.35) with (15.45), we note that the columns of the matrix $[L]$ must contain the components of the three eigenvectors $\{\underset{\sim}{\mathbf{p}}_1, \underset{\sim}{\mathbf{p}}_2, \underset{\sim}{\mathbf{p}}_3\}$ with respect to the orthonormal laboratory basis. If the tensor $\underset{\approx}{\mathbf{A}}$ is symmetric, the eigenvectors will be orthonormal and $[L]$ will be orthogonal.

Using vector basis notation, Eq. (15.45) implies that the tensor $\underset{\approx}{\mathbf{A}}$ can be written as the sum of three "eigen-dyads":

$$\underset{\approx}{\mathbf{A}} = \lambda_1 \underset{\sim}{\mathbf{p}}_1 \underset{\sim}{\mathbf{p}}^1 + \lambda_2 \underset{\sim}{\mathbf{p}}_2 \underset{\sim}{\mathbf{p}}^2 + \lambda_3 \underset{\sim}{\mathbf{p}}_3 \underset{\sim}{\mathbf{p}}^3, \tag{15.47}$$

where the eigenvectors are here presumed to satisfy the normalization of Eq. (15.44) so that, indeed, $\underset{\approx}{\mathbf{A}} \bullet \underset{\sim}{\mathbf{p}}_i = \lambda_i \underset{\sim}{\mathbf{p}}_i$ (no sum on i). Incidentally, note that

$$\underset{\sim}{\mathbf{p}}^i \bullet \underset{\approx}{\mathbf{A}} = \lambda^i \underset{\sim}{\mathbf{p}}^i \tag{15.48}$$

The dual vectors $\underset{\sim}{\mathbf{p}}^i$, are sometimes called the "left" eigenvectors by virtue of the above relationship. It is apparent that the left eigenvalues are the same as the right eigenvalues, and the left eigenvectors are the *right* eigenvectors of $\underset{\approx}{\mathbf{A}}^T$.

If the tensor is symmetric, there's no distinction between the two types of base vectors, and Eq. (15.47) can be written as simply

$$\text{If } \underset{\approx}{\mathbf{A}} \text{ is symmetric, then } \underset{\approx}{\mathbf{A}} = \lambda_1 \underset{\sim}{\delta}_1 \underset{\sim}{\delta}_1 + \lambda_2 \underset{\sim}{\delta}_2 \underset{\sim}{\delta}_2 + \lambda_3 \underset{\sim}{\delta}_3 \underset{\sim}{\delta}_3 \tag{15.49}$$

Eigenprojectors

Consider a diagonalizable tensor $\underset{\approx}{\mathbf{A}}$ for which two eigenvalues are repeat roots. Suppose, for example, $\lambda_2 = \lambda_3$. Then Eq. (15.47) can be written

$$\underset{\approx}{\mathbf{A}} = \lambda_1 \underset{\sim}{\mathbf{p}}_1 \underset{\sim}{\bar{\mathbf{p}}}^1 + \lambda_2 (\underset{\sim}{\mathbf{p}}_2 \underset{\sim}{\bar{\mathbf{p}}}^2 + \underset{\sim}{\mathbf{p}}_3 \underset{\sim}{\bar{\mathbf{p}}}^3), \tag{15.50}$$

or

$$\underset{\approx}{\mathbf{A}} = \lambda_1 \underset{\approx}{\mathbf{P}}_1 + \lambda_2 \underset{\approx}{\mathbf{P}}_2 , \tag{15.51}$$

where

$$\begin{aligned} \underset{\approx}{\mathbf{P}}_1 &\equiv \underset{\sim}{\mathbf{p}}_1 \underset{\sim}{\bar{\mathbf{p}}}^1 \\ \underset{\approx}{\mathbf{P}}_2 &\equiv \underset{\sim}{\mathbf{p}}_2 \underset{\sim}{\bar{\mathbf{p}}}^2 + \underset{\sim}{\mathbf{p}}_3 \underset{\sim}{\bar{\mathbf{p}}}^3 . \end{aligned} \tag{15.52}$$

By virtue of Eq. (15.44) we note that

$$\underset{\approx}{\mathbf{P}}_1 \bullet \underset{\approx}{\mathbf{P}}_1 = \underset{\approx}{\mathbf{P}}_1 \quad \text{and} \quad \underset{\approx}{\mathbf{P}}_2 \bullet \underset{\approx}{\mathbf{P}}_2 = \underset{\approx}{\mathbf{P}}_2 \tag{15.53}$$

Consequently, the $\underset{\approx}{\mathbf{P}}_k$ tensors are projectors, called **eigenprojectors**. Furthermore,

$$\underset{\approx}{\mathbf{P}}_1 \bullet \underset{\approx}{\mathbf{P}}_2 = \underset{\approx}{\mathbf{0}} , \tag{15.54}$$

Finally, since the eigenvectors form a basis, it can be shown that

$$\underset{\approx}{\mathbf{P}}_1 + \underset{\approx}{\mathbf{P}}_2 = \underset{\approx}{\mathbf{I}} \tag{15.55}$$

Comparing these results with Eq. (10.32) reveals that the eigenprojectors are complementary projectors and the number M in Eq. (10.32) is equal to the number of distinct eigenvalues. Because $\lambda_2 = \lambda_3$, the directions of eigenvectors $\underset{\sim}{\mathbf{p}}_2$ and $\underset{\sim}{\mathbf{p}}_3$ are not unique — they only need to reside in the plane defined by their span. However, even though the directions of $\underset{\sim}{\mathbf{p}}_2$ and $\underset{\sim}{\mathbf{p}}_3$ are not unique, the *sum*, $\underset{\sim}{\mathbf{p}}_2 \underset{\sim}{\mathbf{p}}^2 + \underset{\sim}{\mathbf{p}}_3 \underset{\sim}{\mathbf{p}}^3$, is unique! In other words, the eigenprojector associated with the double root λ_2 is unique. Furthermore, note that the dimension of the range space for each eigenprojector equals the multiplicity of the eigenvalue.

If all of the eigenvalues of the tensor instead had been distinct, we could have performed a similar analysis to obtain

$$\underset{\approx}{\mathbf{A}} = \lambda_1 \underset{\approx}{\mathbf{P}}_1 + \lambda_2 \underset{\approx}{\mathbf{P}}_2 + \lambda_3 \underset{\approx}{\mathbf{P}}_3 , \tag{15.56}$$

where

$$\begin{aligned} \underset{\approx}{\mathbf{P}}_1 &\equiv \underset{\sim}{\mathbf{p}}_1 \underset{\sim}{\bar{\mathbf{p}}}^1 \\ \underset{\approx}{\mathbf{P}}_2 &\equiv \underset{\sim}{\mathbf{p}}_2 \underset{\sim}{\bar{\mathbf{p}}}^2 \\ \underset{\approx}{\mathbf{P}}_3 &\equiv \underset{\sim}{\mathbf{p}}_3 \underset{\sim}{\bar{\mathbf{p}}}^3 \end{aligned} \tag{15.57}$$

and the properties of Eq. (10.32) would again hold, this time with $M=3$.

$$\underset{\approx}{\mathbf{P}}_i \bullet \underset{\approx}{\mathbf{P}}_j = \underset{\approx}{\mathbf{P}}_i \text{ if } i=j \text{ so each } \underset{\approx}{\mathbf{P}}_i \text{ is a projector} \tag{15.58}$$

$$\underset{\approx}{\mathbf{P}}_i \bullet \underset{\approx}{\mathbf{P}}_j = \underset{\approx}{\mathbf{0}} \text{ if } i \neq j \tag{15.59}$$

$$\underset{\approx}{\mathbf{P}}_1 + \underset{\approx}{\mathbf{P}}_2 + \underset{\approx}{\mathbf{P}}_3 = \underset{\approx}{\mathbf{I}} \tag{15.60}$$

Likewise, if all three eigenvalues had been equal (triple root), then the number of distinct eigenvalues would be $M=1$, and the same process would (trivially) give

$$\underset{\approx}{\boldsymbol{A}} = \lambda \underset{\approx}{\boldsymbol{I}}, \tag{15.61}$$

where we note that the identity tensor itself is a (degenerate) projector.

These concepts can be generalized to fourth-order tensors, where (instead of eigenvectors) we will find eigen*tensors*. The idea of eigenprojectors is extremely useful in this case, especially in the study of material anisotropy.

> *"Knowledge is a process of piling up facts.*
> *Wisdom lies in their simplification."*
> *— Martin H. Fischer*

16. Geometrical entities

Equation of a plane

A plane in 3D space can be described completely by specifying the plane's orientation by a unit normal $\underset{\sim}{\boldsymbol{n}}$ and the distance "d" of the plane from the origin. Physically, we know that any position vector $\underset{\sim}{\boldsymbol{x}}$ that points from the origin to a location in the plane must have a projection onto the plane's normal that is identically equal to equal to the distance "d". Thus, the direct notation equation for a plane is

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{n}} = d \qquad (16.1)$$

Written out in component form,

$$x_1 n_1 + x_2 n_2 + x_3 n_3 = d \qquad (16.2)$$

Even though a plane is a two dimensional object, it takes *three* independent parameters to specify it. The unit normal has two independent components (the third being restricted to ensure that $\|\underset{\sim}{\boldsymbol{n}}\| = 1$) and the distance d makes the third parameter. When the plane passes through the origin ($d{=}0$), it is a *linear manifold*. Otherwise, it is called an **affine** space.

If the plane does *not* pass through the origin (i.e., if $d \neq 0$), then the equation for the plane may be written

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{u}} = 1\text{, where } \underset{\sim}{\boldsymbol{u}} \equiv \frac{\underset{\sim}{\boldsymbol{n}}}{d} \qquad (16.3)$$

In this case, $1/u_i$ equals the intercept of the plane on the i^{th} axis of the Cartesian coordinate triad. If the plane *does* pass through the origin, then the equation of the plane is just

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{n}} = 0 \qquad (16.4)$$

The equation of a plane passing through a particular point $\underset{\sim}{\boldsymbol{p}}$ is therefore

$$(\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{p}}) \bullet \underset{\sim}{\boldsymbol{n}} = 0 \qquad (16.5)$$

Equation of a line

Consider a straight line that passes through one point $\underset{\sim}{x}_1$ and a second point $\underset{\sim}{x}_2$. The easiest way to define the set of position vectors $\underset{\sim}{x}$ that extend from the origin to some point on the line is to write the equations in parametric form by simply expressing the fact that the vector from $\underset{\sim}{x}_1$ to any point on the line must be parallel to the vector from $\underset{\sim}{x}_1$ to $\underset{\sim}{x}_2$. Letting the proportionality factor be denoted by "*t*", the parametric equation for a line can be written

$$\boxed{(\underset{\sim}{x}-\underset{\sim}{x}_1) = t(\underset{\sim}{x}_2-\underset{\sim}{x}_1)} \tag{16.6}$$

Since this is a vector expression, it represents a set of *three separate equations*. If the parameter t equals zero, then $\underset{\sim}{x}$ is located at $\underset{\sim}{x}_1$. If $t=1$, then the point $\underset{\sim}{x}$ is located at $\underset{\sim}{x}_2$. Of course, values of t outside the interval from 0 to 1 are permissible as well, and will result in position vectors outside the line segment from $\underset{\sim}{x}_1$ to $\underset{\sim}{x}_2$.

An alternative parametric equation for a line is

$$\underset{\sim}{y} = \alpha\underset{\sim}{n}$$

where

$$\underset{\sim}{y} \equiv \underset{\sim}{x}-\tfrac{1}{2}(\underset{\sim}{x}_1+\underset{\sim}{x}_2), \qquad \underset{\sim}{n} \equiv \frac{\underset{\sim}{x}_2-\underset{\sim}{x}_1}{L}, \quad \text{and} \qquad L = \|\underset{\sim}{x}_2-\underset{\sim}{x}_1\| \tag{16.7}$$

The parameter α varies from –L/2 to +L/2 as $\underset{\sim}{x}$ varies from $\underset{\sim}{x}_1$ to $\underset{\sim}{x}_2$.

More generally, one can define a line by any point $\underset{\sim}{p}$ on the line and the orientation $\underset{\sim}{n}$ of the line. Then the equation of the line can be written

$$\boxed{\underset{\sim}{y} = \alpha\underset{\sim}{n}}, \tag{16.8}$$

where

$$\underset{\sim}{y} = \underset{\sim}{x}-\underset{\sim}{p}, \quad \text{and } \underset{\sim}{n} \text{ defines the orientation of the line} \tag{16.9}$$

Dotting both sides of Eq. (16.8) by $\underset{\sim}{n}$ shows that

$$\alpha = \underset{\sim}{y} \bullet \underset{\sim}{n} \tag{16.10}$$

This expression for α may be substituted back into Eq. (16.8) to give a non-parametric version of the equation of a line:

$$\boxed{\underset{\sim}{y} = (\underset{\sim}{y} \bullet \underset{\sim}{n})\underset{\sim}{n}} \tag{16.11}$$

Physically, this says that the projection of $\underset{\sim}{y}$ onto $\underset{\sim}{n}$ equals $\underset{\sim}{y}$ itself. Even though Eq. (16.11) is non-parametric, it is still a vector equation, so it really represents *three* simultaneous equations. These three equations are not independent. Given arbitrary choices for y_1 and y_2, it is not generally possible to solve Eq. (16.11) for y_3. Certain solvability conditions must be met in order for a solution to exist; namely, the above equation may be solved for y_3 if and only if $y_1n_2 = y_2n_1$. This solvability condition is expressing the requirement that y_1 and y_2 must fall on the "shadow" cast by the line onto the 1-2 plane.

Equation (16.11) is non-parametric, but it is still a set of three distinct equations. It can be expressed as a *single* equation by noting that two vectors $\underset{\sim}{\boldsymbol{a}}$ and $\underset{\sim}{\boldsymbol{b}}$ are equal if and only if $(\underset{\sim}{\boldsymbol{a}}-\underset{\sim}{\boldsymbol{b}})\bullet(\underset{\sim}{\boldsymbol{a}}-\underset{\sim}{\boldsymbol{b}}) = 0$. Therefore, an equivalent version of Eq. (16.11) is

$$\boxed{\underset{\sim}{\boldsymbol{y}}\bullet\underset{\sim}{\boldsymbol{y}} = (\underset{\sim}{\boldsymbol{y}}\bullet\underset{\sim}{\boldsymbol{n}})^2} \tag{16.12}$$

Any of the above boxed equations may be used as an equation of a line in 3D space. All but the last equation are vector expressions, so they actually represent a set of three equations, each of which involves a linear combination of the position vector components. The last equation is a single (scalar-valued) expression for a line, but the price paid for having a single equation is that the position components appear in quadratic form — pretty odd given that it represents the equation for a linear (straight) line!

Equation of a sphere

A sphere is defined as the set of points that are equidistant from the sphere's center. Therefore, if a sphere of radius R is centered at a point $\underset{\sim}{\boldsymbol{p}}$, then its equation is simply

$$\boxed{(\underset{\sim}{\boldsymbol{x}}-\underset{\sim}{\boldsymbol{p}})\bullet(\underset{\sim}{\boldsymbol{x}}-\underset{\sim}{\boldsymbol{p}}) = R^2} \tag{16.13}$$

Equation of an ellipsoid

Suppose you wish to define an ellipsoid in 3D space that is centered at the origin and whose principle axes are parallel to three vectors $\{\underset{\sim}{\boldsymbol{a}}_1, \underset{\sim}{\boldsymbol{a}}_2, \underset{\sim}{\boldsymbol{a}}_3\}$ and whose principal radii are given by the magnitudes of these vectors. With respect to a coordinate system that is aligned with the principle directions, the equation of the ellipsoid is

$$\frac{x_1^2}{\alpha_1^2}+\frac{x_2^2}{\alpha_2^2}+\frac{x_3^2}{\alpha_3^2} = 1\text{, where } \alpha_k = \|\underset{\sim}{\boldsymbol{a}}_k\| \tag{16.14}$$

If, however, the principle axes are *not* coincident with your basis, then the equation for the ellipsoid takes on a more general quadratic form. Namely,

$$\underset{\sim}{\boldsymbol{x}}\bullet\underset{\approx}{\boldsymbol{B}}\bullet\underset{\sim}{\boldsymbol{x}} = 1, \tag{16.15}$$

where the tensor $\underset{\approx}{\boldsymbol{B}}$ is defined by the inverse of the dyad sum

$$\underset{\approx}{\boldsymbol{A}} = \underset{\sim}{\boldsymbol{a}}_k\underset{\sim}{\boldsymbol{a}}_k \tag{16.16}$$

Since the $\underset{\sim}{\boldsymbol{a}}_k$ are (by premise) mutually perpendicular, we know that they must be eigenvectors of the tensor $\underset{\approx}{\boldsymbol{A}}$. Consequently, the eigenvalues of the tensor are $\alpha_k \equiv \|\underset{\sim}{\boldsymbol{a}}_k\|$ and the inverse of the tensor is

$$\underset{\approx}{\boldsymbol{B}} = \sum_{k=1}^{3}\frac{\underset{\sim}{\boldsymbol{a}}_k\underset{\sim}{\boldsymbol{a}}_k}{\alpha_k^4} \tag{16.17}$$

Example. Suppose that

$$\underset{\sim}{\boldsymbol{a}}_1 = \underset{\sim}{\boldsymbol{e}}_1 + \underset{\sim}{\boldsymbol{e}}_2 \tag{16.18}$$

$$\underset{\sim}{\boldsymbol{a}}_2 = 4(-\underset{\sim}{\boldsymbol{e}}_1 + \underset{\sim}{\boldsymbol{e}}_2) \tag{16.19}$$

$$\underset{\sim}{\boldsymbol{a}}_3 = 3\underset{\sim}{\boldsymbol{e}}_3 \tag{16.20}$$

The magnitudes are

$$\alpha_1 = \sqrt{2} \tag{16.21}$$

$$\alpha_2 = 4\sqrt{2} \tag{16.22}$$

$$\alpha_3 = 3 \tag{16.23}$$

The dyads are

$$\underset{\sim}{\boldsymbol{a}}_1\underset{\sim}{\boldsymbol{a}}_1 = \underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1 + \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_1 + \underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_2 + \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2 \qquad \text{-->} \qquad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{16.24}$$

$$\underset{\sim}{\boldsymbol{a}}_2\underset{\sim}{\boldsymbol{a}}_2 = 16(\underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1 - \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_1 - \underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_2 + \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2) \qquad \text{-->} \qquad \begin{bmatrix} 16 & -16 & 0 \\ -16 & 16 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{16.25}$$

$$\underset{\sim}{\boldsymbol{a}}_1\underset{\sim}{\boldsymbol{a}}_1 = 9\underset{\sim}{\boldsymbol{e}}_3\underset{\sim}{\boldsymbol{e}}_3 \qquad \text{-->} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 9 \end{bmatrix} \tag{16.26}$$

Thus,

$$[\underset{\sim}{A}^{-1}] = \frac{1}{(\sqrt{2})^4}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{(4\sqrt{2})^4}\begin{bmatrix} 16 & -16 & 0 \\ -16 & 16 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{3^4}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 9 \end{bmatrix} \tag{16.27}$$

or

$$[\underset{\sim}{A}^{-1}] = \begin{bmatrix} \frac{17}{64} & \frac{15}{64} & 0 \\ \frac{15}{64} & \frac{17}{64} & 0 \\ 0 & 0 & \frac{1}{9} \end{bmatrix} \tag{16.28}$$

Thus, the equation for the ellipsoid is

$$\frac{17}{64}x_1^2 + \frac{17}{64}x_2^2 + \frac{1}{9}x_3^2 + \frac{15}{64}x_1x_2 + \frac{15}{64}x_2x_1 = 1 \tag{16.29}$$

Of course, the last two terms, which come from the off-diagonals, may be combined, but we left them separated to emphasize the structure of the solution.

Equation of a cylinder with an ellipse-cross-section

The equation for a cylinder whose cross section is an ellipse could be regarded as a degenerate case of the above ellipsoid for which one of the specified vectors, say $\underset{\sim}{\boldsymbol{a}}_3$, is infinite in magnitude. In this case, only the first two direction vectors, $\underset{\sim}{\boldsymbol{a}}_1$ and $\underset{\sim}{\boldsymbol{a}}_2$ are needed to specify the geometry of the cylindrical ellipsoid. In this case, Eq. (16.17) becomes

$$\underset{\approx}{\boldsymbol{B}} = \frac{\underset{\sim}{\boldsymbol{a}}_1\underset{\sim}{\boldsymbol{a}}_1}{\alpha_1^4} + \frac{\underset{\sim}{\boldsymbol{a}}_2\underset{\sim}{\boldsymbol{a}}_2}{\alpha_2^4} \qquad (16.30)$$

Equation of a right circular cylinder

The special case of a *circular* cylinder of radius r follows from Eq. (16.30) by taking

$$\underset{\sim}{\boldsymbol{a}}_1 = \underset{\sim}{\boldsymbol{a}}_2 \qquad (16.31)$$

where

$$\alpha_1 = \alpha_2 = r \qquad (16.32)$$

Then Eq. (16.30) becomes

$$\underset{\approx}{\boldsymbol{B}} = \frac{\underset{\sim}{\boldsymbol{e}}_1\underset{\sim}{\boldsymbol{e}}_1 + \underset{\sim}{\boldsymbol{e}}_2\underset{\sim}{\boldsymbol{e}}_2}{r^2} = \frac{\underset{\approx}{\boldsymbol{I}} - \underset{\sim}{\boldsymbol{n}}\underset{\sim}{\boldsymbol{n}}}{r^2} \qquad (16.33)$$

Here, $\underset{\sim}{\boldsymbol{e}}_i$ is the unit vector in the direction of $\underset{\sim}{\boldsymbol{a}}_i$ and $\underset{\sim}{\boldsymbol{n}}$ is the unit vector in the direction of the cylinder axis. The final step in Eq. (16.33) followed from the fact that the cylinder axis is perpendicular to the first two directions. With that result, Eq. (16.15) becomes simply

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}} - (\underset{\sim}{\boldsymbol{n}} \bullet \underset{\sim}{\boldsymbol{x}})^2 = r^2 \qquad (16.34)$$

or

$$\| \underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{n}}\underset{\sim}{\boldsymbol{n}} \bullet \underset{\sim}{\boldsymbol{x}} \| = r \qquad (16.35)$$

Equation of a general quadric (including hyperboloid)

A quadric surface is governed by an equation of the form

$$\underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\boldsymbol{B}} \bullet \underset{\sim}{\boldsymbol{x}} = 1 \qquad (16.36)$$

As already discussed, this form is capable of describing an ellipsoid. However, that is just one class of surfaces described by the form. Without loss, the $\underset{\approx}{\boldsymbol{B}}$ may be presumed symmetric (if not, then it must be symmetrized in order for the comments in this section to remain true).

An ellipsoid corresponds to the case that the $\underset{\approx}{\boldsymbol{B}}$ tensor is positive definite. If the $\underset{\approx}{\boldsymbol{B}}$ tensor possesses negative eigenvalues, then the quadric surface is a hyperboloid.

Generalization of the quadratic formula and "completing the square"

Consider a spatially varying field given by an equation of the following form:

$$\psi = \underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{x}} + \underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{x}} + c \tag{16.37}$$

As long as $\underset{\approx}{\boldsymbol{A}}$ is invertible, we claim that there exists a symmetric tensor $\underset{\approx}{\boldsymbol{B}}$ such that this field can be written in the form

$$\psi = (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{p}}) \bullet \underset{\approx}{\boldsymbol{B}} \bullet (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{p}}) - d \tag{16.38}$$

Expanding this out, recognizing the $\underset{\approx}{\boldsymbol{B}}$ may be presumed symmetric without loss, gives

$$\psi = \underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\boldsymbol{B}} \bullet \underset{\sim}{\boldsymbol{x}} - 2\underset{\sim}{\boldsymbol{p}} \bullet \underset{\approx}{\boldsymbol{B}} \bullet \underset{\sim}{\boldsymbol{x}} + \underset{\sim}{\boldsymbol{p}} \bullet \underset{\approx}{\boldsymbol{B}} \bullet \underset{\sim}{\boldsymbol{p}} - d \tag{16.39}$$

Equating this expression with Eq. (16.37) gives

$$\underset{\approx}{\boldsymbol{B}} = \underset{\approx}{\boldsymbol{A}} \tag{16.40}$$

$$-2\underset{\sim}{\boldsymbol{p}} \bullet \underset{\approx}{\boldsymbol{B}} = \underset{\sim}{\boldsymbol{b}} \tag{16.41}$$

$$\underset{\sim}{\boldsymbol{p}} \bullet \underset{\approx}{\boldsymbol{B}} \bullet \underset{\sim}{\boldsymbol{p}} - d = c \tag{16.42}$$

The last two equations can be solved for $\underset{\sim}{\boldsymbol{p}}$ and d. Thus we have the result:

$$\text{If } \psi = \underset{\sim}{\boldsymbol{x}} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{x}} + \underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{x}} + c$$

$$\text{then } \psi = (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{p}}) \bullet \underset{\approx}{\boldsymbol{A}} \bullet (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{p}}) + d,$$

$$\text{where } \underset{\sim}{\boldsymbol{p}} = -\frac{1}{2}\underset{\approx}{\boldsymbol{A}}^{-1} \bullet \underset{\sim}{\boldsymbol{b}} \text{ and } d = \frac{1}{4}\underset{\sim}{\boldsymbol{b}} \bullet \underset{\approx}{\boldsymbol{A}}^{-1} \bullet \underset{\sim}{\boldsymbol{b}} - c \tag{16.43}$$

Written out in full,

$$\psi = \left\{\underset{\sim}{\boldsymbol{x}} + \frac{1}{2}\underset{\approx}{\boldsymbol{A}}^{-1} \bullet \underset{\sim}{\boldsymbol{b}}\right\} \bullet \underset{\approx}{\boldsymbol{A}} \bullet \left\{\underset{\sim}{\boldsymbol{x}} + \frac{1}{2}\underset{\approx}{\boldsymbol{A}}^{-1} \bullet \underset{\sim}{\boldsymbol{b}}\right\} + \left(\frac{1}{4}\underset{\sim}{\boldsymbol{b}} \bullet \underset{\approx}{\boldsymbol{A}}^{-1} \bullet \underset{\sim}{\boldsymbol{b}} - c\right), \tag{16.44}$$

We will be interested in describing the shape of surfaces of constant ψ. Such surfaces can be described by considering only the case that $\psi=0$, because nonzero ψ values of ψ can always "absorbed" into the parameter "c." For the case that $\psi=0$, Eq. (16.37) is like a tensor version of a quadratic formula, so inquiring minds want to know what is the analog of the quadratic formula. In particular, what is the analog of the discriminant? To answer this question, let's define

$$\underset{\sim}{\boldsymbol{y}} \equiv 2\underset{\approx}{\boldsymbol{A}} \bullet \underset{\sim}{\boldsymbol{x}} + \underset{\sim}{\boldsymbol{b}} \tag{16.45}$$

or, solving for $\underset{\sim}{\boldsymbol{x}}$,

$$\underset{\sim}{\boldsymbol{x}} = \frac{1}{2}\underset{\approx}{\boldsymbol{A}}^{-1} \bullet (\underset{\sim}{\boldsymbol{y}} - \underset{\sim}{\boldsymbol{b}}) \tag{16.46}$$

With this change of variables, Eq. (16.44) becomes

$$4\psi = \underset{\approx}{\mathbf{A}}^{-1}\boldsymbol{:}[\underset{\sim}{\mathbf{y}}\underset{\sim}{\mathbf{y}} + \underset{\sim}{\mathbf{b}}\underset{\sim}{\mathbf{b}}] - 4c \tag{16.47}$$

Suppose that $\psi=0$. Then, if this result were for scalars rather than tensors, it would read

$$0 = \frac{1}{a}[y^2 + b^2] - 4c, \qquad \text{or} \qquad y = \pm\sqrt{b^2 - 4ac} \tag{16.48}$$

Thus, we see that the vector $\underset{\sim}{\mathbf{y}}$ plays a role similar to that played by the discriminant in scalar quadratics! As a matter of fact, note that Eq. (16.46) can be written

$$\underset{\sim}{\mathbf{x}} = (2\underset{\approx}{\mathbf{A}})^{-1} \bullet (-\underset{\sim}{\mathbf{b}} + \underset{\sim}{\mathbf{y}}), \tag{16.49}$$

which is the analog of the scalar quadratic formula, $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. In the scalar quadratic equation, there were only two values of y that would make $\psi=0$. By contrast, for the tensor case, setting $\psi=0$ gives

$$0 = \underset{\approx}{\mathbf{A}}^{-1}\boldsymbol{:}[\underset{\sim}{\mathbf{y}}\underset{\sim}{\mathbf{y}} + \underset{\sim}{\mathbf{b}}\underset{\sim}{\mathbf{b}}] - 4c \tag{16.50}$$

or, written in a somewhat different manner

$$\underset{\sim}{\mathbf{y}}^* \bullet \underset{\approx}{\mathbf{A}}^{-1} \bullet \underset{\sim}{\mathbf{y}}^* = 1, \text{ where } \underset{\sim}{\mathbf{y}}^* \equiv \frac{\underset{\sim}{\mathbf{y}}}{\sqrt{\underset{\sim}{\mathbf{b}} \bullet \underset{\approx}{\mathbf{A}}^{-1} \bullet \underset{\sim}{\mathbf{b}} - 4c}} \tag{16.51}$$

which describes a quadric surface (i.e., a *family* of $\underset{\sim}{\mathbf{y}}$ vectors exists that will make $\psi=0$).

> *"Our ability to delude ourselves may be an important survival tool."*
>
> *— Lily Tomlin*

17. Polar decomposition

Singular value decomposition

Let $[F]_{M\times N}$ denote an $M\times N$ matrix. The "singular value decomposition" theorem from matrix analysis [26, 34] states there exist matrices $[q]_{M\times M}$, $[r]_{N\times N}$, and $[\Lambda]_{M\times N}$ for which

(v) $[q]$ and $[r]$ are orthogonal*.

(vi) $[\Lambda]$ is "pseudo-diagonal" (i.e., $\Lambda_{ij} = 0$ whenever $i \neq j$)

(vii) $[\Lambda]$ is positive semi-definite in the sense that its diagonal $(i=j)$ components, Λ_i, are non-negative.

such that $[F]$ can be written

$$[F]_{M\times N} = [q]_{M\times M}[\Lambda]_{M\times N}[r]^T_{N\times N} \tag{17.1}$$

is valid. The numbers Λ_i^2 *make up* the eigenvalues of $[F]^T[F]$ (and the Λ_i are the *non-negative* square roots of Λ_i^2, perhaps with some zeros added to fill out remaining diagonal slots in $[\Lambda]$); the columns of $[r]$ hold the associated "right singular" eigenvectors. Similarly, the numbers Λ_i^2 make up the eigenvalues of $[F][F]^T$ with columns of $[q]$ holding the associated "left singular" eigenvectors. The proof can be found in any good matrix analysis book. The Λ_i are called the "singular values".

Special case: $M = N$**.** Suppose now that all matrices have the same dimensions. Since $[q]$ is orthogonal, we know that

$$[q]^T[q] = [I] \tag{17.2}$$

Therefore, Eq. (17.1), once specialized to square matrices, may be written without loss in the form

$$[F] = [q][\Lambda][q]^T[q][r], \tag{17.3}$$

or

* As with most of the theorems cited in this book, this theorem does have a simple generalization to include the possibility that $[F]$ is complex, in which case $[q]$ and $[r]$ would be *unitary*. For simplicity, this book considers only real matrices so, in this context, unitary means the same thing as orthogonal.

$$[F] = [V][R]\text{, where } [V] \equiv [q][\Lambda][q]^T \text{ and } [R] \equiv [q][r] \tag{17.4}$$

Similarly, since $[r]$ is orthogonal, we know that

$$[r][r]^T = [I] \tag{17.5}$$

Therefore, Eq. (17.1) may be written without loss in the form

$$[F] = [Q][r][r]^T[\Lambda][R], \tag{17.6}$$

or

$$[F] = [R][U]\text{, where } [U] \equiv [r]^T[\Lambda][r] \text{ and } [R] \equiv [q][r] \tag{17.7}$$

Note that $[R]$ is itself orthogonal and that $[U]$ and $[V]$ are symmetric. Thus, the singular value decomposition guarantees the existence of symmetric matrices $[U]$ and $[V]$ as well as an orthogonal matrix $[R]$ such that

$$[F] = [R][U] = [V][R] \tag{17.8}$$

The polar decomposition theorem:

In tensor analysis (especially in continuum mechanics), the singular-value decomposition theorem (specialized to invertible square matrices) is called the “polar decomposition” theorem. Specifically, for each invertible tensor $\underset{\approx}{\boldsymbol{F}}$ there exists a unique orthogonal tensor $\underset{\approx}{\boldsymbol{R}}$, a unique symmetric positive definite “right stretch” $\underset{\approx}{\boldsymbol{U}}$, and a unique symmetric positive definite “left stretch” $\underset{\approx}{\boldsymbol{V}}$ such that

$$\underset{\approx}{\boldsymbol{F}} = \underset{\approx}{\boldsymbol{R}} \bullet \underset{\approx}{\boldsymbol{U}} = \underset{\approx}{\boldsymbol{V}} \bullet \underset{\approx}{\boldsymbol{R}} \tag{17.9}$$

A tensor is called a stretch if it is symmetric and positive definite. Physically, a stretch describes a deformation in which there is an orthonormal triad of material vectors that change length but not orientation. These special vectors are the principal directions of the stretch and the corresponding eigenvalues are the ratios of deformed length to undeformed length. A physical interpretation of the polar decomposition is shown in Fig. 17.1. Note that $\underset{\approx}{\boldsymbol{R}} \bullet \underset{\approx}{\boldsymbol{U}}$ represents a stretch $\underset{\approx}{\boldsymbol{U}}$ followed by a rotation $\underset{\approx}{\boldsymbol{R}}$.

In the polar decomposition theorem, the descriptors “symmetric” and “positive definite” are *requirements*, not consequences. A stretch must be *both* symmetric and positive definite — merely showing that a tensor $\underset{\approx}{\boldsymbol{F}}$ is symmetric is not sufficient to prove that it is a stretch. For example, any rotation of exactly $180°$ will produce a symmetric $\underset{\approx}{\boldsymbol{F}}$ tensor, but it will not be positive definite.

The classic analytical procedure for finding $\underset{\approx}{\boldsymbol{R}}$, $\underset{\approx}{\boldsymbol{U}}$, and $\underset{\approx}{\boldsymbol{V}}$ is as follows:

- Construct a tensor $\underset{\approx}{\boldsymbol{C}} = \underset{\approx}{\boldsymbol{F}}^T \bullet \underset{\approx}{\boldsymbol{F}}$, and perform a spectral analysis on it to find its eigenvalues C_k and eigenprojectors $\underset{\approx}{\boldsymbol{P}}_k$.* Then $\underset{\approx}{\boldsymbol{C}} = C_1\underset{\approx}{\boldsymbol{P}}_1 + C_2\underset{\approx}{\boldsymbol{P}}_2 + C_3\underset{\approx}{\boldsymbol{P}}_3$ and

* If $\underset{\sim}{p}_k$ is the eigenvector associated with C_k, then the eigenprojector $\underset{\approx}{\boldsymbol{P}}_k$ equals the dyad $\underset{\sim}{p}_k \otimes \underset{\sim}{p}_k$, where there is no sum on k.

$\underset{\approx}{\boldsymbol{U}} = \underset{\approx}{\boldsymbol{C}}^{1/2} = \sqrt{C_1}\underset{\approx}{\boldsymbol{P}}_1 + \sqrt{C_2}\underset{\approx}{\boldsymbol{P}}_2 + \sqrt{C_3}\underset{\approx}{\boldsymbol{P}}_3$. The positive square root must be taken to make $\underset{\approx}{\boldsymbol{U}}$ positive definite.*

- Compute $\underset{\approx}{\boldsymbol{R}} = \underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{U}}^{-1}$.
- Compute $\underset{\approx}{\boldsymbol{V}} = \underset{\approx}{\boldsymbol{R}} \bullet \underset{\approx}{\boldsymbol{U}} \bullet \underset{\approx}{\boldsymbol{R}}^T$.

Alternatively, if only $\underset{\approx}{\boldsymbol{V}}$ is desired, it may be computed by $\underset{\approx}{\boldsymbol{V}} = (\underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{F}}^T)^{1/2}$. and then the rotation would be given by $\underset{\approx}{\boldsymbol{R}} = \underset{\approx}{\boldsymbol{V}}^{-1} \bullet \underset{\approx}{\boldsymbol{F}}$. An excellent proof of the polar decomposition theorem is in Ref. [24]. An invalid proof can be found in Ref. [18] where the authors wrongly assert that a symmetric $\underset{\approx}{\boldsymbol{F}}$ is a stretch, which is false because a $180°$ rotation is a symmetric tensor, but is not positive definite and therefore not a stretch; furthermore, Ref. [18] fails to prove uniqueness.

* By computing $\underset{\sim}{\boldsymbol{U}}$ in the principal basis of $\underset{\sim}{\boldsymbol{C}}$, we have ensured that it will be symmetric. In general, an $N \times N$ positive definite symmetric matrix like $\underset{\sim}{\boldsymbol{C}}$ can have an *infinite* number of square roots, of which only 2^N are symmetric and only one is symmetric and positive definite.

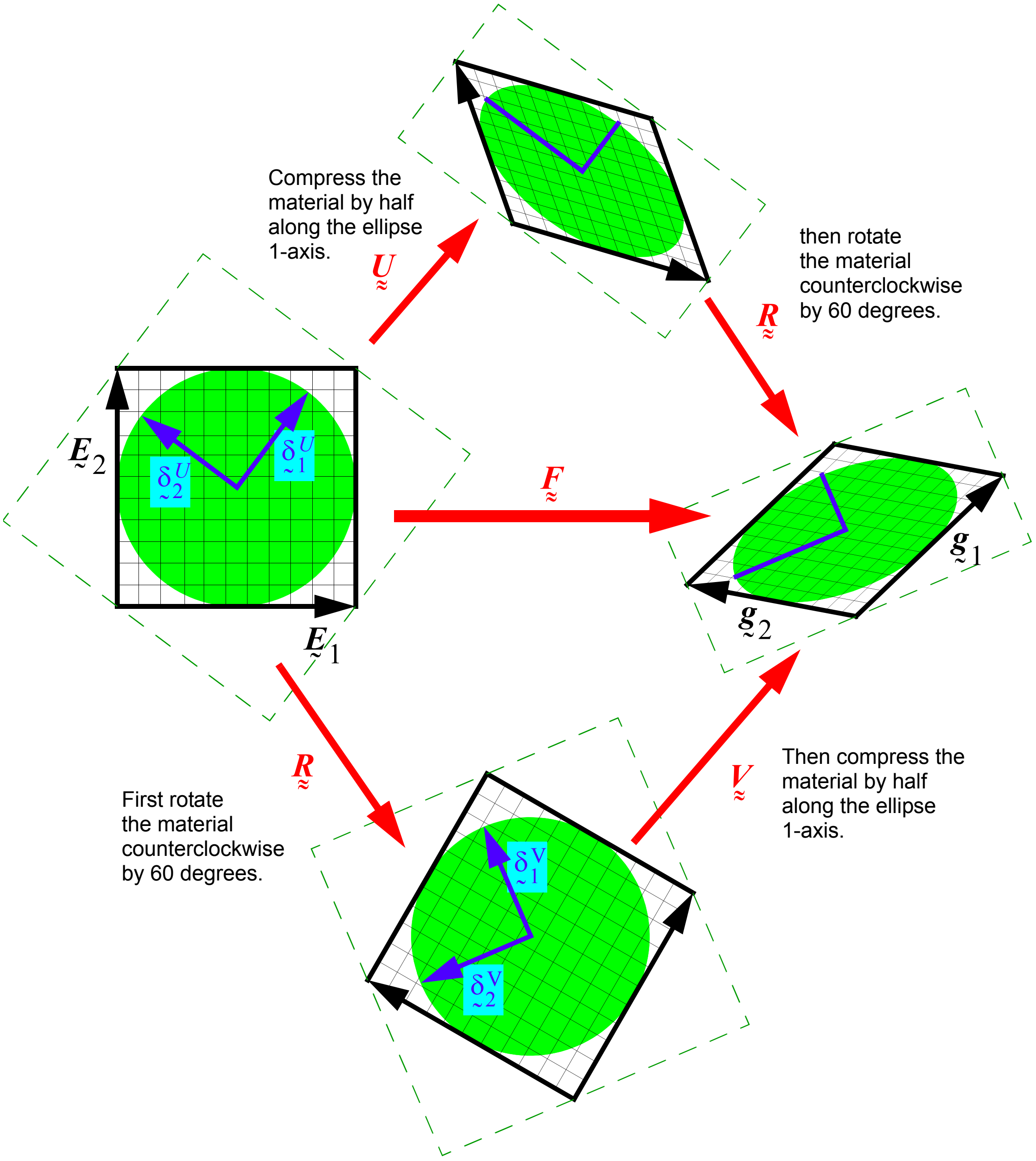


Figure 17.1. Visualization of the polar decomposition. This figure shows that the deformation $\underset{\sim}{F}$ can be visualized as two step process. The upper $\underset{\sim}{R} \bullet \underset{\sim}{U}$ path first applies the stretch $\underset{\sim}{U}$ to compress the material by a factor of $1/2$ in the direction of the vector labeled $\underset{\sim}{\delta}_1^U$ and then the rotation tensor $\underset{\sim}{R}$ rotates counterclockwise by $60°$. The same deformation is achieved on the bottom path by first rotating by $60°$ and *then* compressing by a factor of $1/2$ in the direction of the vector labeled $\underset{\sim}{\delta}_1^V$. In these figures, we have "painted" a circle (or sphere in 3D) on the reference cube to show how it becomes an ellipse (or ellipsoid in 3D). The vectors $\underset{\sim}{\delta}_k^U$ and $\underset{\sim}{\delta}_k^V$ lie on the major axes of the ellipse.

Incidentally, because $\underset{\approx}{\boldsymbol{R}}$ is orthogonal, it follows that $\underset{\approx}{\boldsymbol{R}} = \underset{\approx}{\boldsymbol{R}}^{-T}$. Therefore, the rotation can be alternatively be computed by

$$\underset{\approx}{\boldsymbol{R}} = \underset{\approx}{\boldsymbol{F}}^{-T} \bullet \underset{\approx}{\boldsymbol{U}} \quad \text{or by} \quad \underset{\approx}{\boldsymbol{R}} = \underset{\approx}{\boldsymbol{V}} \bullet \underset{\approx}{\boldsymbol{F}}^{-T} \tag{17.10}$$

which is sometimes more computationally convenient.

Polar decomposition is a *nonlinear* projection

Recall that the reference stretch tensor is computed by

$$\underset{\approx}{\boldsymbol{U}} = P(\underset{\approx}{\boldsymbol{F}}), \tag{17.11}$$

where the operator P is defined for any argument $\underset{\approx}{\boldsymbol{X}}$ by

$$P(\underset{\approx}{\boldsymbol{X}}) = +(\underset{\approx}{\boldsymbol{X}}^T \bullet \underset{\approx}{\boldsymbol{X}})^{1/2} \tag{17.12}$$

We have inserted the "+" sign in front of this equation to reiterate that the square root of a tensor is not unique, but the *positive definite* square root is unique. The transformation P is a projection in the sense defined in Eq. (6.44) because $P(P(\underset{\approx}{\boldsymbol{X}})) = P(\underset{\approx}{\boldsymbol{X}})$. In other words, if $\underset{\approx}{\boldsymbol{F}}$ is already a symmetric positive definite tensor, then its associated polar stretch is *itself*.

Similarly, the rotation tensor, can be regarded purely as a function of $\underset{\approx}{\boldsymbol{F}}$ as follows:

$$\underset{\approx}{\boldsymbol{R}} = \underset{\approx}{\boldsymbol{F}} \bullet \underset{\approx}{\boldsymbol{U}}^{-1} = \underset{\approx}{\boldsymbol{F}} \bullet [+(\underset{\approx}{\boldsymbol{X}}^T \bullet \underset{\approx}{\boldsymbol{X}})^{1/2}]^{-1} = Q(\underset{\approx}{\boldsymbol{F}}) \tag{17.13}$$

The operator Q is defined for any argument $\underset{\approx}{\boldsymbol{X}}$ by

$$Q(\underset{\approx}{\boldsymbol{X}}) = \underset{\approx}{\boldsymbol{X}} \bullet (\underset{\approx}{\boldsymbol{X}}^T \bullet \underset{\approx}{\boldsymbol{X}})^{-1/2} \tag{17.14}$$

The operator Q is a projector because $Q(Q(\underset{\approx}{\boldsymbol{X}})) = Q(\underset{\approx}{\boldsymbol{X}})$. In other words, if $\underset{\approx}{\boldsymbol{F}}$ is already a rotation, then its associated polar rotation tensor is itself.

Neither of the projectors, P or Q, are *linear* operators. The codomain is not a linear manifold for either of these operators. The codomain for P is the set of all positive definite symmetric tensors. It's true that the sum of two positive definite tensors is itself positive definite; however, an *arbitrary* linear combination of positive definite tensors (i.e., a general combination that permits negative coefficients) will not be itself positive definite. The codomain of Q is the set of all orthogonal rotation tensors. This is not a linear space because adding two orthogonal tensors will not result in an orthogonal tensor.

The *FAST* way to do a polar decomposition in 2D

For planar deformations, the deformation gradient tensor has the form

$$[F] = \begin{bmatrix} F_{11} & F_{12} & 0 \\ F_{21} & F_{22} & 0 \\ 0 & 0 & F_{33} \end{bmatrix}, \text{ where } F_{33} > 0 \tag{17.15}$$

We claim by serendipity[*] that the polar decomposition can be performed rapidly by the following formula:

$$[R] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad [U] = [R]^T[F] \tag{17.16}$$

where

$$\cos\theta = \frac{F_{11}+F_{22}}{\sqrt{(F_{11}+F_{22})^2+(F_{21}-F_{12})^2}}$$

$$\sin\theta = \frac{F_{21}-F_{12}}{\sqrt{(F_{11}+F_{22})^2+(F_{21}-F_{12})^2}} \tag{17.17}$$

Beware! You must define $\cos\theta$ and $\sin\theta$ *separately* in order to uniquely determine the rotation angle. It is certainly true that

$$\tan\theta = (F_{21}-F_{12})/(F_{11}+F_{22}), \tag{17.18}$$

but this relation does not uniquely define the rotation angle because there are always *two* angles θ in the range from 0 to 2π that satisfy the above equation. By contrast there is only *one* angle in the range from 0 to 2π that satisfies Eq. (17.17). The following code fragment may be used to compute a polar rotation tensor in 2-D:

```
c = F(1,1)+F(2,2); s = F(2,1)-F(1,2);
d = Sqrt(c*c+ s*s);
c=c/d;s=s/d;
R(1,1)=c;R(2,1)=-s;R(1,2)=s;R(2,2)=c;
```

A fast and accurate numerical 3D polar decomposition

We have already described the classic analytical method for doing polar decomposition in 3D. This method, unfortunately, entails finding the eigenvalues of a 3×3 matrix. Doing that requires solving a third-order characteristic equation. Analytical methods do exist for finding roots of third-order equations, but those methods actually entail a fair amount of computation, which makes them susceptible to round-off error and inefficiency. Below, we outline an iterative method for obtaining a polar decomposition.

Insert algorithm here

[*] It's easy to verify that our formulas yield an orthogonal [R] and a symmetric [U]. It is straightforward, but tedious, to also prove that our formula gives a *positive definite* [U] matrix. This property is essential in order to confirm the validity of our serendipitous formulas.

You might feel uncomfortable about using an iterative algorithm, but you shouldn't. In numerical implementations, iterative algorithms can sometimes give you *more* accurate results than analytical solutions when the analytical solutions involve numerous multiplications, square roots, and other round-off error producing operations. You can obtain maximum accuracy from an iterative algorithm by testing for convergence against the machine precision. The machine precision is the largest number `P` for which `(P+1.0)-1.0` evaluates to zero on your computer. For most computers running double precision, the machine precision `P` is in the neighborhood of 1×10^{-20}. Even analytical solutions cannot hope to achieve any more accuracy than machine precision.

Dilation-Distortion (volumetric-isochoric) decomposition

GOAL: Describe another (less common) multiplicative decomposition that breaks a tensor transformation into two parts, one part that captures size changes and the other part characterizing shape changes.

Like the polar decomposition, a different multiplicative decomposition to be described in this section applies equally well to *any* tensor, but we will explain it in the physical context that the tensor is a deformation gradient $\underset{\approx}{F}$. The operation $\underset{\sim}{x} = \underset{\approx}{F} \bullet \underset{\sim}{X}$ describes a homogenous deformation of space in which the initial position vector $\underset{\sim}{X}$ is deformed into a new location $\underset{\sim}{x}$. A region having volume V_o in the initial configuration will deform into a new region with deformed volume V, and the two volumes are related by

$$V = JV_o\text{, where } J \equiv \det\underset{\approx}{F} \tag{17.19}$$

The "Jacobian" J is sometimes called the **dilatation**. It is improper, in general, to alternatively use the term "dilation" because **dilation** properly means volume change *without* a change in shape.* A deformation tensor generally describes both changes in volume and changes in shape. For example, a cube will deform to a parallelepiped. The deformation will purely dilational (shape preserving) if and only if the deformation gradient is isotropic; hence, it would have to be proportional to the identity tensor, making it of the form $\alpha\underset{\approx}{I}$ for some scalar α. Applying Eq. (3.99) for this special case of a 3×3 matrix, note that the determinant of $\alpha\underset{\approx}{I}$ is given by α^3. Since this determinant must equal the Jacobian J, we can assert that $\alpha = J^{1/3}$ and therefore

$$\text{A tensor } \underset{\approx}{F} \text{ is a pure } \textbf{dilation} \text{ if and only if } \underset{\approx}{F} = J^{1/3}\underset{\approx}{I}\text{, where } J \equiv \det\underset{\approx}{F}. \tag{17.20}$$

On the other extremely we can consider a pure **distortion** deformation (also called an **isochoric** deformation), which is one that permits shape changes, but not volume changes.

$$\text{A tensor } \underset{\approx}{F} \text{ is a pure } \textbf{distortion} \text{ if and only if } \det\underset{\approx}{F} = 1. \tag{17.21}$$

* Your pupils *dilate* when you see mouth watering food or an attractive person — your pupils grow in size, but still remain circular. Contrast this response with a feline's slit-like pupils which *dilatate* (not just dilate) in response to light — a cat's pupils change both size *and* shape.

The identity tensor is the only degenerate case that, according to these definitions would be simultaneously a dilation and a distortion. This case is degenerate because the volume does not change and the shape does not change.

Now consider a tensor $\underset{\approx}{F}$ that is potentially *neither* a pure dilation or a pure distortion. We wish to decompose this general deformation into two separate (multiplicative) steps, a dilation followed by a distortion. That is, we seek an isotropic tensor $\underset{\approx}{A}$ and an isochoric tensor $\underset{\approx}{G}$ such that $\underset{\approx}{F} = \underset{\approx}{A} \bullet \underset{\approx}{G}$. Of course, since $\underset{\approx}{A}$ is isotropic it will also be true that $\underset{\approx}{F} = \underset{\approx}{G} \bullet \underset{\approx}{A}$. Knowing that $\det\underset{\approx}{F} = J$ and $\det\underset{\approx}{G} = 1$ shows that $\det\underset{\approx}{A} = J$ and therefore

$$\underset{\approx}{A} = J^{1/3}\underset{\approx}{I}\text{, where } J \equiv \det\underset{\approx}{F}\,. \qquad (17.22)$$

from which it follows that

$$\underset{\approx}{G} = J^{-1/3}\underset{\approx}{F} \qquad (17.23)$$

Thermomechanics application. The dilation-distortion decomposition (DDD) is useful to resolve one particularly vexing issue in generalizing thermodynamics for gases to solids. In gas thermodynamics, the specific internal energy u is typically regarded as a function of the entropy s and the specific volume ($v \equiv 1/\rho$). Consequently, by using the chain rule, the time rate of the internal energy can be written

$$\dot{u} = T\dot{s} - p\dot{v} \qquad (17.24)$$

where

$$p = -\left(\frac{\partial u}{\partial v}\right)_s \qquad (17.25)$$

and

$$T = \left(\frac{\partial u}{\partial s}\right)_v \qquad (17.26)$$

As written, p and T are merely mathematical entities lacking physical meaning. However, for gas dynamics, you can use the second law to assert that they must be the pressure and temperature respectively. The negative sign is used in Eq. (17.25) because, by convention, pressure is positive in compression. Another commonly used energy-like quantity, the enthalpy is defined

$$h = u + pv \qquad (17.27)$$

When generalizing to thermoelasticity of *solids* similar analysis techniques are used. The internal energy of a thermoelastic solid is presumed to depend on entropy s and some measure of strain or deformation, which we will denote $\underset{\approx}{V}$. This tensor is not the same as the tensor $\underset{\approx}{V}$ from the polar decomposition, though some people might choose it to be since the polar stretch *is* a measure of deformation. Other people might choose $\underset{\approx}{V}$ to be, say, the Lagrange strain from continuum mechanics theory while others might choose $\underset{\approx}{\mathbf{V}}$ to simply be the deformation gradient $\underset{\approx}{\mathbf{F}}$. Associated with your choice for $\underset{\approx}{\mathbf{V}}$, there must exist a so-called "conjugate" specific stress tensor $\underset{\approx}{\mathbf{P}}$ such that the rate of internal energy can be written

$$\dot{u} = \underset{\approx}{\mathbf{P}} : \dot{\underset{\approx}{\mathbf{V}}} + T\dot{s} \qquad (17.28)$$

This is the solid mechanics analog of Eq. (17.24). There is a difference of sign in the mechanical p-v work term because, unlike gases, stress is typically taken to be positive in *tension* in solid mechanics. It can be shown that the generalization of Eqs. (17.25) and (17.26) are

$$\underset{\approx}{\mathbf{P}} = \left(\frac{\partial u}{\partial \underset{\approx}{\mathbf{V}}}\right)_s \qquad (17.29)$$

and

$$T = \left(\frac{\partial u}{\partial s}\right)_{\underset{\approx}{\mathbf{V}}} \qquad (17.30)$$

Again, there is a sign difference in the first of these equations because of the sign convention for stress. The generalization of the enthalpy in Eq. (17.27) is

$$h = u + \underset{\approx}{\mathbf{P}} : \underset{\approx}{\mathbf{V}} \qquad (17.31)$$

Typically, the tensor $\underset{\approx}{\mathbf{V}}$ chosen to be something *dimensionless* such as a strain or the deformation gradient tensor. Contrast this with the situation in gas dynamics where the analogous variable v has dimensions of 1/mass. In thermoelasticity, the specific internal energy must still be energy/mass. Hence, if $\underset{\approx}{\mathbf{V}}$ is dimensionless, $\underset{\approx}{\mathbf{P}}$ must have dimensions of energy per mass. Stress has dimensions of energy per *volume*. Therefore $\underset{\approx}{\mathbf{P}}$ can be regarded as a stress *divided by density*, which is why we called it a *specific* stress. Using concepts from continuum mechanics, the expression $\underset{\approx}{\mathbf{P}} : \underset{\approx}{\mathbf{V}}$, called the *stress power* must be expressible in any of the following forms:

$$\underset{\approx}{\mathbf{P}} : \dot{\underset{\approx}{\mathbf{V}}} = \frac{1}{\rho}\underset{\approx}{\sigma} : \underset{\approx}{\mathbf{D}} \qquad (17.32)$$

$$\underset{\approx}{\mathbf{P}} : \dot{\underset{\approx}{\mathbf{V}}} = \frac{1}{\rho_o}\underset{\approx}{\bar{s}} : \dot{\underset{\approx}{\bar{\varepsilon}}} \qquad (17.33)$$

$$\underset{\approx}{\mathbf{P}} : \dot{\underset{\approx}{\mathbf{V}}} = \frac{1}{\rho_o}\underset{\approx}{\mathbf{t}} : \dot{\underset{\approx}{\mathbf{F}}} \qquad (17.34)$$

Here, $\underset{\approx}{\sigma}$ is the Cauchy stress tensor (it is the stress tensor that undergraduate engineering students learn about, defined to be force per unit deformed area), $\underset{\approx}{\mathbf{D}}$ is the symmetric part of the velocity gradient, ρ_o is the *initial density* (equal to the current density ρ times the Jacobian J), $\underset{\approx}{\bar{\varepsilon}}$ is the Lagrange strain [defined $\underset{\approx}{\bar{\varepsilon}} = \frac{1}{2}(\underset{\approx}{\mathbf{F}}^T \bullet \underset{\approx}{\mathbf{F}} - \underset{\approx}{\mathbf{I}})$], $\underset{\approx}{\bar{s}}$ is the Second-Piola-Kirchhoff stress [defined $\underset{\approx}{\bar{s}} = J\underset{\approx}{\mathbf{F}}^{-1} \bullet \underset{\approx}{\sigma} \bullet \underset{\approx}{\mathbf{F}}^{-T}$], $\underset{\approx}{\mathbf{F}}$ is the deformation gradient tensor, and $\underset{\approx}{\mathbf{t}}$ is the First-Piola-Kirchhoff stress [defined $\underset{\approx}{\mathbf{t}} = J\underset{\approx}{\sigma} \bullet \underset{\approx}{\mathbf{F}}^{-T} = \underset{\approx}{\sigma} \bullet \underset{\approx}{\mathbf{F}}^{C}$].

Although the tensor $\underset{\approx}{\mathbf{D}}$ is somewhat “rate-like” it is a well-known result from continuum mechanics that there exists no tensor $\underset{\approx}{\mathbf{V}}$ such that $\underset{\approx}{\mathbf{D}} = \dot{\underset{\approx}{\mathbf{V}}}$. Consequently, Eq. (17.32) provides no guidance for assigning a mathematically well-founded meaning for $\underset{\approx}{\mathbf{V}}$. On the other hand, Eq. (17.33) is of the proper structure to permit us to choose

$$\underset{\approx}{\mathbf{V}} = \underset{\approx}{\bar{\varepsilon}} \text{ and } \underset{\approx}{\mathbf{P}} = \frac{1}{\rho_o}\underset{\approx}{\bar{s}} \tag{17.35}$$

Likewise, Eq. (17.34) admits the alternative choice

$$\underset{\approx}{\mathbf{V}} = \underset{\approx}{\mathbf{F}} \text{ and } \underset{\approx}{\mathbf{P}} = \frac{1}{\rho_o}\underset{\approx}{\mathbf{t}} \tag{17.36}$$

Is there anything that makes either of these two choices preferable? Either of these conjugate pairs will give the same result for the stress power. However, these two choices will give *different* results for the product $\underset{\approx}{\mathbf{P}}\mathbf{:}\underset{\approx}{\mathbf{V}}$ that appears in the enthalpy Eq. (17.31). Consider, for example, a purely dilational (nondistortional) deformation for which $\underset{\approx}{\mathbf{F}} = J^{1/3}\underset{\approx}{\mathbf{I}}$ and (for an isotropic material) $\underset{\approx}{\sigma} = -p\underset{\approx}{\mathbf{I}}$. In this special case, noting that $\rho_o = \rho J = J/v$,

$$\frac{1}{\rho_o}\underset{\approx}{\bar{s}}\mathbf{:}\underset{\approx}{\bar{\varepsilon}} = -\frac{1}{2}pv(1-J^{-2/3}) \tag{17.37}$$

and

$$\frac{1}{\rho_o}\underset{\approx}{\mathbf{t}}\mathbf{:}\underset{\approx}{\mathbf{F}} = -3pv \tag{17.38}$$

Neither of these versions of $\underset{\approx}{\mathbf{P}}\mathbf{:}\underset{\approx}{\mathbf{V}}$ reduce to the same value, nor do they reduce to $-pv$ from gas dynamics. Consequently, thermoelasticity theory with these conjugate pairs won’t reduce down to classic gas and inviscid fluid thermodynamics when the material and the deformation are both isotropic. Another strain choice, the logarithmic or Hencky strain, also comes close to reducing to the gas equations, but using it entails serious computational overhead. At the root of the problem is the fact that gas thermodynamics is cast in terms of *volumetric* strains whereas solid mechanics is cast in terms of *linear* strains.

> *"Advice is what we ask for when we already know the answer, but wish we didn't."*
>
> *— Erica Jong*

18. Material symmetry

The symmetry of a material is measured by how its properties vary with material orientation. If the material properties are unaffected by material orientation, then the material is said to be **proper isotropic**. If the material properties are additionally unchanged upon a *reflection*, then the tensor is **strictly isotropic**. If the material properties are unaffected by rotation about some given vector $\underset{\sim}{a}$, as for unidirectional fiber-reinforced plastics or plywood, then the material is **transversely isotropic**. The tensor theory of symmetries is covered in a rigorous and elegant manner through the use of group theory. Here will only give a simple overview of the results of greatest use in material modeling.

Similar concepts apply to tensor analysis. If the components of a tensor are unchanged upon an orthonormal right-handed change of basis, then the tensor is said to be **proper isotropic**. If the tensor components are also unchanged upon reflections, then the tensor is **strictly isotropic**. In physical applications, there are many instances when reflections are not permitted (e.g., in materials modeling, that would entail turning a material inside-out, which is not of interest). Any tensor that is strictly isotropic will also be proper isotropic. Proper-isotropy is a broader property of significant physical interest, so you should always characterize proper-isotropy constraints before investigating strict-isotropy. If the components of a tensor are unchanged for any rotation of the basis about some given vector $\underset{\sim}{a}$, then the tensor is **transversely isotropic**.

What is isotropy?

GOAL: Describe two competing definitions of isotropy and the relative merits of each.

There are two possible ways to define isotropy.

(i) Definition 1: a tensor is *strictly*-isotropic if its components are unchanged upon any orthonormal change in basis

(ii) Definition 2: a tensor is *proper*-isotropic if its components are unchanged upon any *same-handed* change in basis

Consider a second-order tensor $\underset{\approx}{A}$ of class V_3^2. According to definition #1, this tensor is isotropic if

$$\text{Strict isotropy: } Q_{ip}Q_{jq}A_{pq} = A_{ij} \text{ for any orthogonal matrix } [Q] \tag{18.1}$$

According to definition number 2, the tensor is *proper-isotropic* if

$$\text{Proper isotropy: } R_{ip}R_{jq}A_{pq} = A_{ij} \text{ for any proper orthogonal matrix } [R] \qquad (18.2)$$

A proper orthogonal tensor (i.e., an orthogonal tensor with determinate $+1$) is a rotation operation. When a tensor is proper-isotropic, we are saying that it “looks the same” no matter what orientation you view it from. Stated differently, if you hold yourself fixed, then the tensor “looks the same” no matter how much you turn it. For proper-isotropy, there is no guarantee that the tensor won’t look different if you invert it (i.e., if you switch to a left-handed basis). Strict isotropy insists that the tensor must also look the same for both rotations and reflections.

So which definition is more useful? Clearly, strict-isotropy is more restrictive because satisfaction of Eq. 18.1 automatically guarantees satisfaction of 18.2, but not vice-versa. Which definition is more useful from a practical standpoint? We contend that *proper-isotropy* is more meaningful for engineering applications. In the vast majority of physics applications, you want to know when something will be unchanged upon a rotation, but you don’t care what happens upon a (non-physical) reflection. Knowledge of proper-isotropy (even when strict-isotropy does not hold) is very useful information and should be tested first in order to not discount important tensors. Unless otherwise stated, we take the term “isotropic” to mean “proper-isotropic.”

As a general rule, to determine the most general form for an isotropic tensor, you should consider restrictions placed on the components of a tensor for *particular choices of the rotation tensor, which will simplify your general analysis for exploring the set of all possible rotations.* Good “particular” choices for the rotations are 90° rotations about the coordinate axes:

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \qquad \text{and} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad (18.3)$$

The component restrictions arising from these special choices will only give you *necessary* conditions for isotropy. However, frequently, these necessary conditions turn out to also be sufficient conditions.

For example, to deduce the most general form for an isotropic *vector* $\underset{\sim}{v}$, you would demand that its components satisfy the equation $R_{ip}v_p = v_i$ or, in matrix form,

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \qquad (18.4)$$

This must hold for all proper-orthogonal matrices $[R]$. Consequently, it must hold for any of the special cases in Eq. 18.3. Considering the first case,

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \qquad \text{or} \qquad \begin{bmatrix} -v_2 \\ v_1 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{18.5}$$

With this simple test, we have already learning an important *necessary* condition for a vector to be isotropic. Namely, v_2 must equal $-v_1$ and v_2 must also equal v_1. The only way one number can equal another number *and the negative* of that other number is if both numbers are zero. Thus, an isotropic vector would have to be of the form $<0, 0, v_3>$. We can use this information in a second necessary condition of isotropy where we use the second choice in Eq. 18.3:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v_3 \end{bmatrix}, \qquad \text{or} \qquad \begin{bmatrix} v_3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v_3 \end{bmatrix} \tag{18.6}$$

This result tells us that v_3 itself must be zero. In other words, a *necessary* requirement for a vector to be isotropic is that the vector's components must all be zero (vectors are boring). Now that these two specialized tests have greatly simplified the nature of vectors that can be isotropic, it is time to use these *necessary* conditions in the general equation 18.4. Trivially, if a vector is zero, then Eq. 18.4 is satisfied. Thus, the zero vector is the *only* isotropic vector.

Looking now at second-order tensors, note that any linear combination of isotropic tensors is itself isotropic. By this we mean if $\underset{\approx}{\mathbf{A}}$ and $\underset{\approx}{\mathbf{B}}$ (each of class V_3^2) are isotropic, then

$$R_{ip}R_{jq}A_{pq} = A_{ij} \qquad \text{and} \qquad R_{ip}R_{jq}B_{pq} = B_{ij} \tag{18.7}$$

and it follows that any linear combination $\alpha\underset{\approx}{\mathbf{A}} + \beta\underset{\approx}{\mathbf{B}}$ will be isotropic because

$$R_{ip}R_{jq}(\alpha A_{pq} + \beta B_{pq}) = \alpha(R_{ip}R_{jq}A_{pq}) + \beta(R_{ip}R_{jq}B_{pq}) = \alpha A_{ij} + \beta B_{ij} \tag{18.8}$$

Important consequence. Since any linear combination of isotropic tensors (of a given class) will itself be isotropic, it follows that the set of isotropic tensors (of that class) is a linear subspace. Thus, the zero tensor will always be an isotropic tensor. More importantly, if there exist *any* non-trivial (i.e., nonzero) isotropic tensors, then there must exist a basis for the subspace of all isotropic tensors of that type. For example, below we prove that a second-order tensor of class V_3^2 is isotropic if and only if it is some multiple of the identity tensor. Consequently, the identity tensor itself is a basis for the subspace of isotropic tensors of class V_3^2. There's only one base tensor, so this must be a one-dimensional space. Projecting an arbitrary tensor onto this space (using the projection techniques covered in this document) gives you the *isotropic part* of that tensor. Interestingly, we will find that the dimension of the space of isotropic tensors depends on both the order of the tensor and on the dimension of physical space. Given tensors of class V_m^n, the space of isotropic tensors depends on both m and n.

Isotropic second-order tensors in 3D space

In earlier sections, we introduced the notion of vector and tensor invariants. Ordinarily, vector and tensor components will change upon a change in basis and hence components are numbers, but not scalar invariants. Now we ask what relationships must be satisfied if the components themselves are to remain unchanged upon a change of basis. Such relationships define isotropic tensors. A vector or tensor is called **isotropic** if its components are the same with respect to *any* orthonormal basis. Any linear combination of isotropic tensors is itself an isotropic tensor. Therefore, the set of all isotropic tensors (of a given order) forms a subspace. By this, we mean that there exists a finite number of "primitive" isotropic tensors that serve as a basis such that any isotropic tensor may be expressed as a linear combination of the primitive isotropic base tensors.

First we will prove that the only isotropic vector is the zero vector. Referring to Eq. (13.11), a vector is isotropic if and only if $\tilde{v}_k = \hat{v}_k$ for any coordinate transformation. Consequently, we seek conditions on the vector components such that

$$\begin{aligned} &v_j = v_i Q_{ji} \text{ for all orthogonal } Q_{ji} \\ &\text{or, in matrix form, } \{v\} = [Q]\{v\} \text{ for all orthogonal } [Q] \end{aligned} \qquad (18.9)$$

Since this must hold for all orthogonal $[Q]$, it must also hold for any particular choice for an orthogonal $[Q]$. Choosing

$$[Q] = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (18.10)$$

shows that one restriction on the vector components must be $v_1 = v_2$. Other choices for the orthogonal tensor eventually reveals that not only must the components of the vector all be equal, the only way to satisfy Eq. (18.9) is for *all* of the components to be *zero*. Hence, the *only* isotropic vector is the zero vector.

Referring to Eq. (13.14), a second-order tensor in 3D space is isotropic if and only if

$$[T] = [Q][T][Q]^T \text{ for all orthogonal } [Q] \qquad (18.11)$$

This condition implies that the tensor must be a scalar times the identity tensor $\underset{\approx}{\boldsymbol{I}}$. In other words, the most general form for an isotropic tensor referenced to 3D space is

$$\begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix} \qquad (18.12)$$

where α is an arbitrary scalar. Isotropic tensors in 3D space is therefore one-dimensional since only one scalar is needed. The identity tensor $\underset{\approx}{\boldsymbol{I}}$ is a basis for this space. Any general tensor $\underset{\approx}{\boldsymbol{B}}$ may be projected to its isotropic part by the operation

$$\mathrm{iso}\underset{\approx}{\boldsymbol{B}} = \frac{\underset{\approx}{\boldsymbol{I}}(\underset{\approx}{\boldsymbol{I}}:\underset{\approx}{\boldsymbol{B}})}{\underset{\approx}{\boldsymbol{I}}:\underset{\approx}{\boldsymbol{I}}} \tag{18.13}$$

Note that $\underset{\approx}{\boldsymbol{I}}:\underset{\approx}{\boldsymbol{B}} = \mathrm{tr}\underset{\approx}{\boldsymbol{B}}$ and $\underset{\approx}{\boldsymbol{I}}:\underset{\approx}{\boldsymbol{I}} = \mathrm{tr}\underset{\approx}{\boldsymbol{I}} = 3$. Hence,

$$\mathrm{iso}\underset{\approx}{\boldsymbol{B}} = \frac{1}{3}(\mathrm{tr}\underset{\approx}{\boldsymbol{B}})\underset{\approx}{\boldsymbol{I}} \tag{18.14}$$

This is a very familiar result. The idea of finding the isotropic part by projecting to the space of isotropic tensors becomes less obvious when considering tensors in spaces of different dimensions.

Isotropic second-order tensors in 2D space

GOAL: Demonstrate that the proper-isotropic space is two dimensional for tensors of class V_2^2, in stark contrast to the result for V_2^3. The strict-isotropic space is one-dimensional.

To consider both proper-isotropy and strict-isotropy of tensors of class V_2^2 (i.e., tensors in two-dimensions that can be represented by 2×2 matrices), we first need to identify the general form for an orthogonal tensor in this space.

$$\text{Let } Q = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{18.15}$$

We seek restrictions on the components $(a, b, c, \text{and } d)$ such that

$$[Q]^T[Q] = [I] \tag{18.16}$$

or

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{18.17}$$

Multiplying this out shows that the components must satisfy

$$\begin{aligned} a^2 + c^2 &= 1 \\ b^2 + d^2 &= 1 \\ ab + dc &= 0 \end{aligned} \tag{18.18}$$

We can satisfy the first two constraints automatically by setting

$$\begin{aligned} a &= \cos\alpha,\ c = \sin\alpha \\ b &= \cos\beta,\ d = \sin\beta \end{aligned} \tag{18.19}$$

Then satisfying the last constraint requires that

$$\cos\alpha\cos\beta + \sin\alpha\sin\beta = 0 \tag{18.20}$$

or

$$\cos(\beta - \alpha) = 0 \tag{18.21}$$

or

$$\beta = \alpha \pm \frac{\pi}{2} \tag{18.22}$$

Putting this result back into Eq. 18.19 using $\beta = \alpha + \frac{\pi}{2}$ gives

$$\begin{aligned} a &= \cos\alpha\,,\ c = \sin\alpha \\ b &= -\sin\alpha\,,\ d = \cos\alpha \end{aligned} \tag{18.23}$$

Putting this result back into Eq. 18.15 yields a proper orthogonal matrix, so we will denote it by $[R]$. Namely,

$$[R] = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \tag{18.24}$$

On the other hand, substituting Eq. 18.22 back into Eq. 18.19 using $\beta = \alpha - \frac{\pi}{2}$ gives

$$\begin{aligned} a &= \cos\alpha\,,\ c = \sin\alpha \\ b &= \sin\alpha\,,\ d = -\cos\alpha \end{aligned} \tag{18.25}$$

Putting this result back into Eq. 18.15 yields an improper orthogonal matrix,

$$[Q] = \begin{bmatrix} \cos\alpha & \sin\alpha \\ \sin\alpha & -\cos\alpha \end{bmatrix} \tag{18.26}$$

Equation 18.24 is the most general form for a proper orthogonal matrix in 2D and 18.26 is the most general form for an improper matrix. With this result in hand, we are ready to explore the nature of isotropic tensors in 2D. For a second order tensor to be proper isotropic its components must satisfy

$$\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{18.27}$$

Considering, as a special case, $\alpha = \frac{\pi}{2}$ gives

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{18.28}$$

or

$$A_{11} = A_{22} \quad \text{and} \quad A_{12} = -A_{21} \tag{18.29}$$

Since this result was obtained by considering a special rotation, we only know it a *necessary* condition for isotropy. However, substituting this condition back into Eq. 18.27 shows that it is also a *sufficient* condition.

Consequently, the most general form for an isotropic tensor referenced to 2D space is of the form

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} \tag{18.30}$$

where a and b are arbitrary parameters. Any tensor in 2D space that is of this proper-isotropic form may be expressed as a linear combination of the following two primitive base tensors:

$$[\underset{\approx}{\boldsymbol{I}}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad [\underset{\approx}{\varepsilon}] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{18.31}$$

Note that $\underset{\approx}{\varepsilon}$ is the 2D version of the permutation symbol; namely, ε_{ij} is zero if $i=j$, it is +1 if $ij=12$, and it is –1 if $ij=21$. In two dimensions, the (proper) isotropic part of a second-order tensor $\underset{\approx}{\boldsymbol{F}}$ would be obtained by projecting the tensor onto the space spanned by the basis in Eq. (18.31). This basis is orthogonal, but not normalized, so the appropriate projection operation is

$$\begin{aligned}
\text{iso}\underset{\approx}{\boldsymbol{F}} &= \frac{\underset{\approx}{\boldsymbol{I}}(\underset{\approx}{\boldsymbol{I}}\boldsymbol{:}\underset{\approx}{\boldsymbol{F}})}{\underset{\approx}{\boldsymbol{I}}\boldsymbol{:}\underset{\approx}{\boldsymbol{I}}} + \frac{\underset{\approx}{\varepsilon}(\underset{\approx}{\varepsilon}\boldsymbol{:}\underset{\approx}{\boldsymbol{F}})}{\underset{\approx}{\varepsilon}\boldsymbol{:}\underset{\approx}{\varepsilon}} \\
&= \frac{1}{2}\underset{\approx}{\boldsymbol{I}}(\underset{\approx}{\boldsymbol{I}}\boldsymbol{:}\underset{\approx}{\boldsymbol{F}}) + \frac{1}{2}\underset{\approx}{\varepsilon}(\underset{\approx}{\varepsilon}\boldsymbol{:}\underset{\approx}{\boldsymbol{F}}) \\
&= \frac{\underset{\approx}{\boldsymbol{I}}(F_{11}+F_{22})}{2} + \frac{\underset{\approx}{\varepsilon}(F_{12}-F_{21})}{2}
\end{aligned} \tag{18.32}$$

In component form,

$$\text{iso}\underset{\approx}{\boldsymbol{F}} = \frac{1}{2}\begin{bmatrix} F_{11}+F_{22} & F_{12}-F_{21} \\ F_{21}-F_{12} & F_{11}+F_{22} \end{bmatrix} \tag{18.33}$$

Incidentally, the 2D polar rotation tensor associated with any 2D $\underset{\approx}{\boldsymbol{F}}$ tensor is proportional to $\text{iso}\underset{\approx}{\boldsymbol{F}}$. Specifically,

$$\underset{\approx}{\boldsymbol{R}} = \frac{1}{\sqrt{(F_{11}+F_{22})^2+(F_{12}-F_{21})^2}}\begin{bmatrix} F_{11}+F_{22} & F_{12}-F_{21} \\ F_{21}-F_{12} & F_{11}+F_{22} \end{bmatrix}$$
$$\text{for 2D space only!.} \tag{18.34}$$

This formula provides a particularly convenient (and barely known) method for finding the polar decomposition for 2D problems.

Recall that proper isotropy is less constraining than strict-isotropy. To be strictly isotropic, a tensor must first be proper-isotropic, which means it must be of the form

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} \tag{18.35}$$

This result guarantees that the components of the tensor will be unchanged under a proper (same-handed) basis transformation. To be proper isotropic, it must also satisfy transformations using the matrix $[Q]$ from Eq. 18.26. Namely

$$\begin{bmatrix} \cos\alpha & \sin\alpha \\ \sin\alpha & -\cos\alpha \end{bmatrix} \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ \sin\alpha & -\cos\alpha \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \tag{18.36}$$

Considering $\alpha = 0$ shows that b must be zero, which also is easily confirmed to be a sufficient condition to satisfy Eq. 18.36. Consequently, for *strict* isotropy, a second-order tensor in 2D must be a multiple of the identity. That is, the permutation tensor $\underset{\approx}{\varepsilon}$ is proper-isotropic, but not strictly isotropic. Its components will change sign upon changing to a differently handed basis. In dyadic form, the permutation tensor may be written

$$\underset{\approx}{\varepsilon} = \underset{\sim}{e}_1 \underset{\sim}{e}_2 - \underset{\sim}{e}_2 \underset{\sim}{e}_1 \tag{18.37}$$

Changing to a different-handed basis (i.e., exchanging $\underset{\sim}{e}_1$ and $\underset{\sim}{e}_2$) will automatically change the sign of the components.

Isotropic fourth-order tensors

Turning our attention back to tensors in 3D space, the most general form for a *fourth-order* isotropic tensor c_{ijrs} is

$$c_{ijrs} = \alpha(\delta_{ij}\delta_{rs}) + \beta(\delta_{ir}\delta_{js}) + \gamma(\delta_{is}\delta_{jr}) \tag{18.38}$$

For a proof, see, for example, Ref. [2]. The above expression has three arbitrary parameters, so the space of isotropic fourth-order tensors is three-dimensional,* with base tensors given by the parentheses above. The basis tensors in the parentheses of Eq. (18.38) are only one choice for the basis. In material modelling, a more convenient basis for the space of isotropic fourth-order tensors is

$$P^{\text{iso}}_{ijkl} = \frac{1}{3}\delta_{ij}\delta_{kl} \tag{18.39a}$$

$$P^{\text{symdev}}_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) - \frac{1}{3}\delta_{ij}\delta_{kl} \tag{18.39b}$$

$$P^{\text{skew}}_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}) \tag{18.39c}$$

Note that these are all constructed from linear combinations of the primitive basis of Eq. (18.38). Even though the component formulas for this alternative basis are considerably more complicated, the properties of this basis are irresistible. Specifically, the basis of Eq. (18.39) consists of *complementary projectors*! By this we mean

$$\boxed{P^{\text{iso}}_{ijkl}P^{\text{iso}}_{klmn} = P^{\text{iso}}_{ijmn}}, \quad P^{\text{iso}}_{ijkl}P^{\text{symdev}}_{klmn} = 0, \quad P^{\text{iso}}_{ijkl}P^{\text{skew}}_{klmn} = 0 \tag{18.40}$$

$$P^{\text{symdev}}_{ijkl}P^{\text{iso}}_{klmn} = 0^{\text{iso}}, \quad \boxed{P^{\text{symdev}}_{ijkl}P^{\text{symdev}}_{klmn} = P^{\text{symdev}}_{ijmn}}, \quad P^{\text{symdev}}_{ijkl}P^{\text{skew}}_{klmn} = 0 \tag{18.41}$$

$$P^{\text{skew}}_{ijkl}P^{\text{iso}}_{klmn} = 0, \quad P^{\text{skew}}_{ijkl}P^{\text{symdev}}_{klmn} = 0, \quad \boxed{P^{\text{skew}}_{ijkl}P^{\text{skew}}_{klmn} = P^{\text{skew}}_{ijmn}} \tag{18.42}$$

* The space is two dimensional if one imposes a minor symmetry restriction that $c_{ijrs}=c_{jirs}=c_{ijsr}$. That's why isotropic elastic materials have only two independent stiffness moduli.

Recall that second-order tensors don’t really take on any meaning until they act on a vector. Likewise, the meaning of a fourth-order tensor should be inferred by operating on a second-order tensor. For any tensor $\underset{\approx}{\boldsymbol{B}}$, note that

$$P^{\text{iso}}_{ijkl}B_{kl} = \frac{1}{3}B_{kk}\delta_{ij} \tag{18.43}$$

$$P^{\text{symdev}}_{ijkl} = \frac{1}{2}(B_{ij}+B_{ji}) - \frac{1}{3}B_{kk}\delta_{ij} \tag{18.44}$$

$$P^{\text{skew}}_{ijkl}B_{kl} = \frac{1}{2}(B_{ij}-B_{ji}) \tag{18.45}$$

Thus, P^{iso}_{ijkl} returns the isotropic part of $\underset{\approx}{\boldsymbol{B}}$, P^{symdev}_{ijkl} returns the symmetric-deviatoric part of $\underset{\approx}{\boldsymbol{B}}$, and P^{skew}_{ijkl} returns the skew-symmetric part of skew-symmetric part of $\underset{\approx}{\boldsymbol{B}}$.

Finding the isotropic part of a fourth-order tensor

GOAL: Reiterate the concept of projections by showing an advanced, higher-order, application.

Fourth-order tensors are of class V_3^4, but they are also of class V_{81}^1. In other words, they may be regarded as 81-dimensional vectors. The set of isotropic fourth-order engineering tensors (IFOET) is closed under tensor addition and scalar multiplication. This means that any linear combination of IFOET tensors will itself be IFOET. Therefore, the set of all IFOET tensors forms a subspace of general fourth-order engineering tensor space.

In the previous section, we showed that any IFOET tensor can be written as a linear combination of P^{iso}_{ijkl}, P^{symdev}_{ijkl}, and P^{skew}_{ijkl}. These three tensors therefore form a basis for the set of all IFOET tensors, and the IFOET subspace must be 3-dimensional. This basis is orthogonal (e.g., $P^{\text{iso}}_{ijkl}P^{\text{skew}}_{ijkl} = 0$), but it is not normalized. We can define an orthonormal basis for IFOET tensors as

$$\hat{P}^{\text{iso}}_{ijkl} \equiv P^{\text{iso}}_{ijkl} \tag{18.46a}$$

$$\hat{P}^{\text{symdev}}_{ijkl} \equiv \frac{P^{\text{symdev}}_{ijkl}}{\sqrt{5}} \tag{18.46b}$$

$$\hat{P}^{\text{skew}}_{ijkl} \equiv \frac{P^{\text{skew}}_{ijkl}}{\sqrt{3}} \tag{18.46c}$$

The denominators in this equation are the magnitudes of the tensors in the numerators, obtained by taking the square root of the inner product of the tensors with themselves. The denominators can be easily remembered because for ordinary second-order engineering tensors (class V_3^2) isotropic tensors have 1 independent component, symmetric deviatoric tensors have 5 independent components, and skew-symmetric tensors have three independent components.

If three m-dimensional *orthonormal* vectors $\underline{\boldsymbol{n}}_1$, $\underline{\boldsymbol{n}}_2$ and $\underline{\boldsymbol{n}}_3$, form a basis for a 3D subspace embedded in the m-dimensional (class V_m^1) vector space, then any vector $\underline{\boldsymbol{x}}$ in the higher dimensional space can be projected to the 3D subspace by applying the operation

$$\underline{\boldsymbol{x}}^{\text{projected}} = \underline{\underline{\boldsymbol{P}}} * \underline{\boldsymbol{x}} \tag{18.47}$$

where

$$\underline{\underline{\boldsymbol{P}}} = \underline{\boldsymbol{n}}_1 \underline{\boldsymbol{n}}_1 + \underline{\boldsymbol{n}}_2 \underline{\boldsymbol{n}}_2 + \underline{\boldsymbol{n}}_3 \underline{\boldsymbol{n}}_3 \tag{18.48}$$

and the “*” denotes the inner product in the V_m^1 space.

We are interested in finding the IFOET part of a general fourth-order engineering tensor X_{ijkl}. This is accomplished by projecting that tensor to the IFOET subspace. Using Eq. (18.47), this operation is found by

$$X_{ijkl}^{\text{IFOET}} = P_{ijklpqrs} X_{pqrs} \tag{18.49}$$

where the components of the sixth-order (V_3^6) tensor are found by using Eq. (18.46) as the orthonormal basis in Eq. (18.48). Namely,

$$P_{ijklpqrs} = P_{ijkl}^{\text{iso}} P_{pqrs}^{\text{iso}} + \frac{1}{5} P_{ijkl}^{\text{symdev}} P_{pqrs}^{\text{symdev}} + \frac{1}{3} P_{ijkl}^{\text{skew}} P_{pqrs}^{\text{skew}} \tag{18.50}$$

A scalar measure of “percent anisotropy”

GOAL:

Add this topic

Transverse isotropy

A material property is said to be “transversely isotropic” about a given unit vector $\underset{\sim}{\boldsymbol{a}}$ if the material can be rotated about this vector without changing the material property. If the property is a scalar invariant, then by definition it cannot change when the basis is rotated. The concept of transverse isotropy becomes increasingly complicated for vector and higher order tensor properties.

In what follows, we often consider a “material orthogonal basis” $\{\underset{\sim}{\boldsymbol{p}}_1, \underset{\sim}{\boldsymbol{p}}_2, \underset{\sim}{\boldsymbol{p}}_3\}$ such that $\underset{\sim}{\boldsymbol{p}}_3 = \underset{\sim}{\boldsymbol{a}}$. This basis will not generally coincide with the “laboratory basis” $\{\underset{\sim}{\boldsymbol{e}}_1, \underset{\sim}{\boldsymbol{e}}_2, \underset{\sim}{\boldsymbol{e}}_3\}$ used in general calculations. Components with respect to the material basis are called material components. Components with respect to the laboratory basis are called laboratory components. Our goal in what follows is to show the simplified material components of transversely isotropic vectors and tensors and then to show how those tensors may be expressed *directly* in the laboratory components without having to perform a coordinate transformation.

A *vector* $\underset{\sim}{\boldsymbol{u}}$ is transversely isotropic if and only if its material components are of the form

$$\{\underset{\sim}{\boldsymbol{u}}\} = \begin{Bmatrix} 0 \\ 0 \\ u_3 \end{Bmatrix} \quad \text{in the material basis } \{\underset{\sim}{\boldsymbol{p}}_i\} \tag{18.51}$$

In other words, the only vector that remains unchanged when rotated about the material symmetry direction $\underset{\sim}{\boldsymbol{a}}$ is one that is itself proportional to $\underset{\sim}{\boldsymbol{a}}$. Thus, since $\underset{\sim}{\boldsymbol{a}}$ is a unit vector, the most general form for a transversely isotropic vector $\boldsymbol{u}$ is

$$\underset{\sim}{\boldsymbol{u}} = u_3\underset{\sim}{\boldsymbol{a}} \tag{18.52}$$

Thus, in the laboratory basis,

$$\underset{\sim}{\boldsymbol{u}} = u_3\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} \quad \text{in the laboratory basis } \{\underset{\sim}{\boldsymbol{e}}_i\} \tag{18.53}$$

Now consider a general *second-order tensor* $\underset{\approx}{\boldsymbol{A}}$. To be transversely-isotropic, the material components of must be of the form

$$[\underset{\approx}{\boldsymbol{A}}] = \begin{bmatrix} A_t & -A_s & 0 \\ A_s & A_t & 0 \\ 0 & 0 & A_a \end{bmatrix} \quad \text{in the material basis } \{\underset{\sim}{\boldsymbol{p}}_i\} \tag{18.54}$$

Here, the subscripts “t” and “a” indicate transverse and axial (relative to the symmetry axis), and the subscript “s” indicates shear. If the tensor is symmetric, then it cannot have shear terms and its most general form is

$$[\underset{\approx}{\boldsymbol{A}}] = \begin{bmatrix} A_t & 0 & 0 \\ 0 & A_t & 0 \\ 0 & 0 & A_a \end{bmatrix} \quad \text{in the material basis } \{\underset{\sim}{\boldsymbol{p}}_i\} \tag{18.55}$$

In other words, the tensor must be diagonal in the material basis, with the 11 and 22 transverse components equal. Thus, transverse isotropy means the tensor must be isotropic in the transverse plane. In dyadic notation, the above equation may be written

$$\underset{\approx}{\boldsymbol{A}} = A_t(\underset{\sim}{\boldsymbol{p}}_1\underset{\sim}{\boldsymbol{p}}_1 + \underset{\sim}{\boldsymbol{p}}_2\underset{\sim}{\boldsymbol{p}}_2) + A_a\underset{\sim}{\boldsymbol{p}}_3\underset{\sim}{\boldsymbol{p}}_3 \tag{18.56}$$

Recall that Eq. (9.24) is true in *any* orthogonal system; therefore it holds for the material coordinate system and Eq. (18.56) can therefore be written

$$\underset{\approx}{\boldsymbol{A}} = A_t(\underset{\approx}{\boldsymbol{I}} - \underset{\sim}{\boldsymbol{p}}_3\underset{\sim}{\boldsymbol{p}}_3) + A_a\underset{\sim}{\boldsymbol{p}}_3\underset{\sim}{\boldsymbol{p}}_3 \tag{18.57}$$

or, since $\underset{\sim}{\boldsymbol{p}}_3 = \underset{\sim}{\boldsymbol{a}}$

$$\underset{\approx}{\boldsymbol{A}} = A_t(\underset{\approx}{\boldsymbol{I}} - \underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{a}}) + A_a\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{a}} \tag{18.58}$$

Rearranging a bit gives the most general *direct notation* form for a transversely isotropic second-order tensor.

$$\underset{\approx}{\mathbf{A}} = A_t \underset{\approx}{\mathbf{I}} + (A_a - A_t)\underset{\sim}{\mathbf{a}}\underset{\sim}{\mathbf{a}} \tag{18.59}$$

The tremendous advantage of Eq. (18.59) over the matrix representation of Eq. (18.54) is that Eq. (18.59) can be computed *directly* in the laboratory basis without having to perform a coordinate transformation. Namely, in the laboratory basis,

$$[\underset{\approx}{\mathbf{A}}] = A_t \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + (A_a - A_t) \begin{bmatrix} a_1 a_1 & a_1 a_2 & a_1 a_3 \\ a_2 a_1 & a_2 a_2 & a_2 a_3 \\ a_3 a_1 & a_3 a_2 & a_3 a_3 \end{bmatrix} \tag{18.60}$$

Furthermore, the Eq. (18.59) is more useful when the transverse axis is changing dynamically because Eq. (18.59) can be differentiated with respect to time.

> *“Don’t worry about people stealing an idea. If it’s original, you will have to ram it down their throats.” — Howard Aiken*

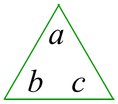
19. Abstract vector/tensor algebra

Structures

GOAL: What are structures, and why use them? Motivate writing down operations in purposely abstract, counter-intuitive, notation to ensure that you don’t introduce your own bias into the analysis.

On page 2, we cited familiar examples of structures, which we defined loosely to be a collection of symbols written down on the page in a carefully prescribed manner and subject to immutable assembly and manipulation rules. The notion of structure *type* also came up. Deciding if one thing is of the same type as something else is often subtle. An orange, for example might be considered to be of the same type as a lemon (they are both citrus fruits). An apple doesn’t fall into the citrus category, but apples, lemons, and oranges *all* fall into the broader category of “fruit.” When exploring structures, you typically want to identify the largest category that will satisfy all of the structure rules (postulates).

For illustration purposes, let’s just “invent” a few new structures. Let’s put three objects (each of the same type) inside a pyramid and declare that this structure represents a a new object of the same type:



(19.1)

With structures, you will often want to avoid endowing anything with physical meaning. For example, when considering the natural logarithm structure, $\ln x$, you might want to replace it with $f(x)$ so that you won’t accidentally assert anything that does not follow from primitive postulates (e.g., you would be less likely to replace $f(x^3)$ by $3f(x)$ unless you had a rule about f that would permit it; working with $\ln x^3$, you might accidentally impose what you know about logarithms to say this equals $3\ln x$ without really proving it). Writing an operation into a new, unfamiliar, notation is an important step towards discovering if other operations might obey identical structure rules. If so, anything you prove about the first operation will automatically be true for the other structure.

In addition to our triangle structure, let’s suppose that we also have other structures with equally odd-looking notation (we will call them the lower and upper box structures)

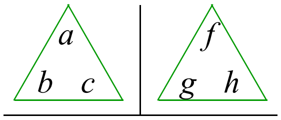
$$\underline{r \mid s} \qquad \text{and} \qquad \overline{p \mid q} \tag{19.2}$$

Again, we will require that these structures represent objects of the same "type" as the individual parts (e.g., if r and s are fruits, then $\underline{r \mid s}$ is a fruit too — perhaps the result of cross breeding). Without ever defining what the structures mean, we could start asserting "rules of behavior" (postulates taken to be true without question). For example, we could demand that the upper and lower box structures must satisfy a commutivity property:

$$\underline{r \mid s} = \underline{s \mid r} \qquad \text{and} \qquad \overline{p \mid q} = \overline{q \mid p} \tag{19.3}$$

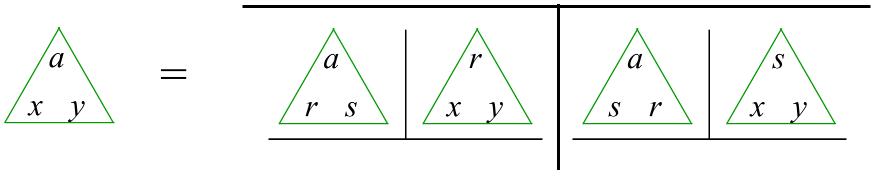
Incidentally, a method for deciding equality of two objects of the same type is presumed to exist by postulate (apples and oranges are both of the same type, but they aren't equal if our criterion for equality is color of the skin; by this criterion, a navel orange equals a Valencia orange equals a tangerine).

As we did with the triangle, we will demand (by postulate) that the structures in Eq. (19.2) result in something of the same type as the individual parts. With this assertion, we can state (theorem, not postulate) that the superstructure



(19.4)

is well defined as long as all of the parts are of the same type. We have purposefully avoided assigning meaning to these structures. Manipulating structures is an essential aspect of *abstract* mathematics — the goal is to define syntax (structure construction and manipulation rules), and see what sorts of conclusions (theorems) can be inferred without biasing ourselves by inserting what we *know* to be true based on the *meanings* of the structures. Mathematicians seek to understand what truths follow strictly from the primitive structure rules. A new primitive postulate is introduced only if the mathematicians become convinced that a certain desired property cannot be obtained from the already existing primitives. For example, they might be forced to assert (immutable postulate) that



(19.5)

Casting everything into abstract structural form permits mathematicians to eventually draw connections between seemingly unrelated operations. Suppose, for example we were to investigate whether or not our triangle operation in Eq. (19.1) might represent partial differentiation so that Eq. (19.1) means the same thing as the partial of a with respect to b holding c constant:

$$\left(\frac{\partial a}{\partial b}\right)_c \tag{19.6}$$

To decide if this *particular* structure could be equivalenced with the "triangle" structure,* we would need to further assign meaning to the two operations in Eq. (19.2). If we let these operations represent multiplication and addition, respectively, then Eq. (19.2) would become

$$rs \qquad \text{and} \qquad p+q \tag{19.7}$$

From here, we can no longer arbitrarily endow further meanings to the structures — we must follow through and check if the other primitive postulates are satisfied. With these interpretations of the operations, we see that Eq. (19.3) is satisfied because

$$rs = sr \text{ and } p+q = q+p \tag{19.8}$$

Furthermore, Eq. (19.5) would become

$$\left(\frac{\partial a}{\partial x}\right)_y = \left(\frac{\partial a}{\partial r}\right)_s\left(\frac{\partial r}{\partial x}\right)_y + \left(\frac{\partial a}{\partial s}\right)_r\left(\frac{\partial s}{\partial x}\right)_y , \tag{19.9}$$

This is the chain rule of partial differentiation, which we know to be true from direct study of differentiation. Assuming that the structure rules specified above form the complete set of primitive postulates, then any superstructures or conclusions that the mathematicians obtained in the abstract setting we could *immediately* know to be true for our *specific* interpretation of the triangle structure. We would not have to go through the effort to prove that the triangle and box structure theorems hold for partial derivatives because the mathematicians have already done the proofs for us — all we had to do was prove that partial differentiation, multiplication, and addition are suitable specific instances of triangle and box structures. Then we get to benefit from the (underpaid) sweat of mathematicians to instantly obtain a new set of truths. In the next section, a less contrived instance of this mindset is covered where we look at the mathematicians' *structural* definition of a vector and then verify that our low-brow engineering vectors are particular instances of their general vector structures.

* The fancy phrase is "isomorphism." Two structures are isomorphic to each other if their structure rules are identical. In this case, we want to determine if partial derivative notation is isomorphic to triangle structure notation. If so, then any theorems derived using only the structure rules for one system will also apply to the other structure.

Definition of an abstract vector

Mathematicians define vectors as being "members of a vector space." A vector space consists of certain basic components:

A1. A field R must exist. *(For this book, the field is the set of reals; for an excellent definition of the term "field", see Ref. [30].) We will here refer to a member of a field as a scalar.**

A2. There must be a discerning definition of membership in a set V.

A3. There must be an operation $f(\alpha, \underset{\sim}{\mathbf{v}})$ that is *proved* closed in V. In other words, if $\alpha \in R$ and $\underset{\sim}{\mathbf{v}} \in V$ then axiom A2 must be used to prove that $f(\alpha, \underset{\sim}{\mathbf{v}}) \in V$. Similarly, there must be an operation $f^*(\underset{\sim}{\mathbf{v}}, \alpha)$ that is also closed in V for every $\alpha \in R$ and $\underset{\sim}{\mathbf{v}} \in V$.

A4. There must be a second operation $g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}})$ that is *proved* closed in V:
If $\underset{\sim}{\mathbf{v}} \in V$ and $\underset{\sim}{\mathbf{w}} \in V$ then $g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}}) \in V$

A5. There must be a well defined process for determining whether two members of V are equal.

A6. The f and g functions must satisfy the following rules:

$$g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}}) = g(\underset{\sim}{\mathbf{w}}, \underset{\sim}{\mathbf{v}}) \text{ and } f(\alpha, \underset{\sim}{\mathbf{v}}) = f^*(\underset{\sim}{\mathbf{v}}, \alpha)$$

$$g(\underset{\sim}{\mathbf{u}}, g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}})) = g(g(\underset{\sim}{\mathbf{u}}, \underset{\sim}{\mathbf{v}}), \underset{\sim}{\mathbf{w}})$$

There must exist a special $\underset{\sim}{\boldsymbol{Z}} \in V$ such that $g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\boldsymbol{Z}}) = \underset{\sim}{\mathbf{v}}$.

The above axioms are purposefully stated using abstract notation to ensure that they are each verified carefully without introducing accidental bias in your analyses stemming from what you "know" about the physical meanings of the operators. *After* you carefully verify that the above axioms are satisfied, only then are you allowed to refer to members of V as **vectors** — don't use this term until you know that all axioms are satisfied. Once all axioms are confirmed, you can with good conscience declare your right to replace the abstract operators f, f^*, and g with the standard structures for scalar multiplication and vector addition. Namely,

$$f(\alpha, \underset{\sim}{\mathbf{v}}) \quad \text{can be replaced with the structure } \alpha\underset{\sim}{\mathbf{v}} \tag{19.10}$$

$$f^*(\underset{\sim}{\mathbf{v}}, \alpha) \quad \text{can be replaced with the structure } \underset{\sim}{\mathbf{v}}\alpha \tag{19.11}$$

$$g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}}) \quad \text{can be replaced with the structure } \underset{\sim}{\mathbf{v}} + \underset{\sim}{\mathbf{w}} \tag{19.12}$$

$$\text{The special vector } \underset{\sim}{\boldsymbol{Z}} \text{ can now be denoted by } \underset{\sim}{\boldsymbol{0}}. \tag{19.13}$$

* Using "scalar" as a synonym for "number" is not strictly correct. In continuum mechanics, a scalar is a number that is unaffected by superposition of a rigid rotation, which is subtly different from an orthogonal change of basis. The trace of the deformation gradient tensor is *not* a scalar because its value changes when there is superimposed rotation. By contrast, the trace of the deformation gradient tensor does *not* change when you change the basis (it's an invariant).

You should avoid stating the vector space axioms using these structures from the outset because it would be “bad citizenship” to use those structures until you finish verifying that *all* the axioms are true. Of course, once the axioms are all known to be true, the conditions of the last axiom can be recast in using familiar addition and multiplication structures as

$$\underset{\sim}{\mathbf{v}} + \underset{\sim}{\mathbf{w}} = \underset{\sim}{\mathbf{w}} + \underset{\sim}{\mathbf{v}} \text{ and } \alpha\underset{\sim}{\mathbf{v}} = \underset{\sim}{\mathbf{v}}\alpha$$

$$\underset{\sim}{\mathbf{u}} + (\underset{\sim}{\mathbf{v}} + \underset{\sim}{\mathbf{w}}) = (\underset{\sim}{\mathbf{u}} + \underset{\sim}{\mathbf{v}}) + \underset{\sim}{\mathbf{w}}$$

There must exist a zero vector $\underset{\sim}{\mathbf{0}} \in V$ such that $\underset{\sim}{\mathbf{v}} + \underset{\sim}{\mathbf{0}} = \underset{\sim}{\mathbf{v}}$.

If we had written the last axiom like this from the outset, it would have looked “obviously” true to you — but that would have been your bias creeping in prematurely. First you have to *prove* that the *g* operation satisfies the rules of addition before you are permitted to denote the operation with the plus sign.

What does this mathematician's definition of a vector have to do with the definition used in applied mechanics? Unfortunately, textbooks seem to fixate on item A6, completely neglecting the far more subtle and difficult items A2, A3, A4, and A5. Engineering vectors are more than something with length and direction. They are also more than simply an array of three numbers. When people define vectors according to the way their components change upon a change of basis, they are implicitly addressing axiom A2.[*] In general, axiom A2 is the most difficult axiom to satisfy when discussing *specific* vector spaces. Verifying axiom A2 demands introducing rather awkward specialized tests that apply to particular sets. To go through the axiom list for ordinary engineering vectors, the process might go like this:

A1. A field *R* must exist: real numbers.

A2. Discerning definition of membership in a set *V*: A member of *V* will be defined to be an array of three numbers $\{v_1, v_2, v_3\}$ defined with respect to a set of mutually perpendicular reference directions (assumed by postulate to exist). Furthermore, to be a member of *V*, the three numbers must change in a particular way if the reference directions are changed to some other mutually perpendicular directions. The new numbers must become $\{v_1^*, v_2^*, v_3^*\}$ where $v_k^* = Q_{ik}v_i$ and Q_{ik} is the cosine of the angle between the i^{th} old direction and the k^{th} new direction.

A3. This axiom requires definition of the operations $f(\alpha, \underset{\sim}{\mathbf{v}})$ and $f^*(\underset{\sim}{\mathbf{v}}, \alpha)$. Anticipating the properties these functions must satisfy, we define them to be ordinary scalar multiplication of a three-component array. If $\underset{\sim}{\mathbf{v}} \in V$, then our discerning definition of membership in V tells us that we must have an array $\{v_1, v_2, v_3\}$ and we will declare that, regardless of the choice of reference directions, both $f(\alpha, \underset{\sim}{\mathbf{v}})$ and $f^*(\underset{\sim}{\mathbf{v}}, \alpha)$ will be computed by $\{\alpha v_1, \alpha v_2, \alpha v_3\}$. So far, we have only defined the operations, but the axiom requires us to prove that the result will be a member of *V*. That's where the component transformation rule aspect of the "discerning definition of membership" comes into play again. Since $\underset{\sim}{\mathbf{v}} \in V$, we know that $v_k^* = Q_{ik}v_i$ and we can just multiply both sides of this by α to assert that $f(\alpha, \underset{\sim}{\mathbf{v}})$ will satisfy the same transformation behavior. Hence, $f(\alpha, \underset{\sim}{\mathbf{v}}) \in V$.

A4. This axiom requires definition of a second operation $g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}})$. If $\underset{\sim}{\mathbf{v}} \in V$ and $\underset{\sim}{\mathbf{w}} \in V$, then we know they have component arrays, and we will define $g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}})$ to have a components $w_i + v_i$. Given that $\underset{\sim}{\mathbf{v}}$ and $\underset{\sim}{\mathbf{w}}$ are, by premise members of *V*, then their conformance to the component transformation rule ensures conformance of $g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}})$, proving that it that this operation is closed in *V*.

A5. There must be a well defined process for determining whether two members of *V* are equal. If $\underset{\sim}{\mathbf{v}} \in V$ and $\underset{\sim}{\mathbf{w}} \in V$, then we will say they are equal if and only their respective components are equal.

A6. The f and g functions defined above are readily seen to satisfy the rules

$$g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}}) = g(\underset{\sim}{\mathbf{w}}, \underset{\sim}{\mathbf{v}}) \text{ and } f(\alpha, \underset{\sim}{\mathbf{v}}) = f^*(\underset{\sim}{\mathbf{v}}, \alpha)$$

$$g(\underset{\sim}{\mathbf{u}}, g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\mathbf{w}})) = g(g(\underset{\sim}{\mathbf{u}}, \underset{\sim}{\mathbf{v}}), \underset{\sim}{\mathbf{w}})$$

The "special" $\underset{\sim}{\boldsymbol{Z}} \in V$ will be declared to be $\{0, 0, 0\}$, from which it readily follows that

$$g(\underset{\sim}{\mathbf{v}}, \underset{\sim}{\boldsymbol{Z}}) = \underset{\sim}{\mathbf{v}}.$$

This concludes the outline of proving that ordinary engineering vectors really are vectors in the sense that mathematicians use the word. Now that we know that, we can immediately make use of any theorems that mathematicians have already derived for general vector spaces. For example, using the primitive axioms, you can prove that the zero vector is unique, and you can define subtraction of vectors. The vector space definitions make no reference to anything like a dot product, so we can't have access to mathematicians' theorems about angles between vectors or lengths of vectors. To do that, we need to add two more axioms as described below.

[*] and they are failing to also address the other axioms!

Many applied mechanics textbooks focus to distraction on proving that every new operation satisfies transformation rules. However, by recognizing that engineering vectors are special instances of abstract (mathematician's) vectors, you don't usually have to do that. Instead, you can assert that various operations used in engineering are special cases of operations already studied by mathematicians for general vector spaces. The mathematicians have *already* proved that these operations produce vectors, so why should we bother doing the same?

Inner product spaces

An inner product (here denoted $(\underset{\sim}{\mathbf{a}}, \underset{\sim}{\mathbf{b}})$) is a scalar-valued binary* operation between two vectors, $\underset{\sim}{\mathbf{a}}$ and $\underset{\sim}{\mathbf{b}}$, that must satisfy the following rules:

A7. $(\underset{\sim}{\mathbf{a}}, \underset{\sim}{\mathbf{b}}) = (\underset{\sim}{\mathbf{b}}, \underset{\sim}{\mathbf{a}})$

A8. $(\underset{\sim}{\mathbf{a}}, \underset{\sim}{\mathbf{a}}) > 0$ if $\underset{\sim}{\mathbf{a}} \neq \underset{\sim}{\mathbf{0}}$ and $(\underset{\sim}{\mathbf{a}}, \underset{\sim}{\mathbf{a}}) = \underset{\sim}{\mathbf{0}}$ only if $\underset{\sim}{\mathbf{a}} = \underset{\sim}{\mathbf{0}}$.

An **inner product space** is just a vector space that has an inner product.

Alternative inner product structures. Once you have verified that these axioms are satisfied, then you are free to replace the notation structure $(\underset{\sim}{\mathbf{a}}, \underset{\sim}{\mathbf{b}})$ with some other structure of your own invention (or one that is commonly employed by your colleagues) and you will be justified in stating that the structure represents an *inner product.* In this document, we adopted the structure $\underset{\sim}{\mathbf{a}} \bullet \underset{\sim}{\mathbf{b}}$ to denote the inner product between two vectors of class V_3^1 and we used the structure $\underset{\approx}{\boldsymbol{A}} : \underset{\approx}{\boldsymbol{B}}$ to denote the inner product between second order tensors of class V_3^2. Recognizing the limitations of these structure conventions for higher-order tensors (e.g., it would be ridiculous to stack up six dots to denote the inner product between sixth-order tensors), we also described the alternative notation $\boldsymbol{A}\textcircled{\scriptsize N}\boldsymbol{B}$ for the inner product between tensors of class V_3^N. You might also see the structure $\boldsymbol{A}\odot\boldsymbol{B}$ to denote the same thing.† Some people use $\boldsymbol{A} \cdot \boldsymbol{B}$ to denote the inner product and $\boldsymbol{AB}$ to denote tensor contraction; we can't use that notation because it would violate self-consistency rules that we have introduced in this book (namely, we use $\boldsymbol{A} \cdot \boldsymbol{B}$ for tensor contraction and $\boldsymbol{AB}$ for dyadic tensor multiplication). When people choose different structures to denote the same operation, don't be annoyed. They probably have good reasons for their choices.

* The term "binary" is just an obnoxiously fancy way of saying that the function has *two* arguments.

† Some people use $\boldsymbol{A} \cdot \boldsymbol{B}$ to denote the inner product (regardless of the order of the operands) and $\boldsymbol{AB}$ to denote tensor contraction; we can't use that notation because it would violate self-consistency rules that we have introduced in this book (namely, we use $\boldsymbol{A} \cdot \boldsymbol{B}$ for tensor contraction so we can't permit this structure to be an alternative representation for the totally different inner product operation. Likewise this book's tensor nomenclature has already said that $\boldsymbol{AB}$ stands for dyadic tensor multiplication so we can't let it also stand for tensor contraction). Issues like these make it essential for you to learn the *local* tensor structures (notation conventions) used by each individual author in each individual publication. Sorry, but that's how it goes in this field.

By the way, that we also introduced in this document the structure $\underset{\approx}{\boldsymbol{A}} \cdot\cdot \underset{\approx}{\boldsymbol{B}}$ whose value is computed in Rectangular Cartesian Systems (RCS) by $A_{ij}B_{ji}$. We did *not* however refer to this scalar valued operation as an "inner product" because the operation fails the positivity axiom A8. Good citizenship demands that the phrase "inner product" should be considered hallowed ground for the mathematicians — don't use this term unless your operation satisfies the required axioms.

Some examples of inner product spaces. Ordinary engineering vectors form an inner product space. First of all, engineering vectors are proved to be vectors in the first place by postulating the existence of some mutually agreed upon reference set of directions (usually orthonormal, such as the edges forming between the floor and walls of the laboratory), and all directional quantities are described in terms of multiples of these postulated "lab" directions (i.e., vector components). To formally verify the axioms for vectors, the vector transformation rule governing how components must change upon a change in basis) is used as the "discerning definition" and ordinary array addition and scalar multiplication are used for the f and g functions. With these awkward rules, it can be proved that engineering vectors are indeed vectors.

For engineering vectors, the inner product is computed by

$$(\underset{\sim}{\mathbf{a}}, \underset{\sim}{\mathbf{b}}) \equiv \underset{\sim}{\mathbf{a}} \bullet \underset{\sim}{\mathbf{b}} = a_1 b_1 + a_2 b_2 + a_3 b_3 = a_k b_k \tag{19.14}$$

The vector transformation rule ensures that you will get the same result from this operation no matter what basis you use. Even though the individual components of vectors will change upon a change of basis, the above *combination* of components will come out the same.

For second-order engineering tensors, the inner product is

$$(\underset{\approx}{\boldsymbol{A}}, \underset{\approx}{\boldsymbol{B}}) \equiv \underset{\approx}{\boldsymbol{A}} : \underset{\approx}{\boldsymbol{B}} = A_{ij} B_{ij} \tag{19.15}$$

The choice of inner product is not unique. For many engineering applications, it is often convenient to define an alternative *energy-based* inner product that uses a symmetric positive definite weighting tensor $\underset{\approx}{\boldsymbol{W}}$:

$$(\underset{\sim}{\mathbf{a}}, \underset{\sim}{\mathbf{b}}) \equiv \tfrac{1}{2} \underset{\sim}{\mathbf{a}} \bullet \underset{\approx}{\boldsymbol{W}} \bullet \underset{\sim}{\boldsymbol{b}} \tag{19.16}$$

Note that it is essential that $\underset{\approx}{\boldsymbol{W}}$ be symmetric in order to satisfy A7 and positive definite to satisfy A8.

The word "energy" is an apt descriptor for this type of inner product. For continuum elasticity problems, a material having a strain $\underset{\approx}{\varepsilon}$ and a stress $\underset{\approx}{\sigma}$ has an internal energy given by

$$u = \tfrac{1}{2} \underset{\approx}{\varepsilon} : \underset{\approx\approx}{\boldsymbol{E}} : \underset{\approx}{\varepsilon}, \tag{19.17}$$

where $\underset{\approx\approx}{\boldsymbol{E}}$ is the fourth-order elastic stiffness tensor. Thus, $\underset{\approx\approx}{\boldsymbol{E}}$ plays a weighting role much like the tensor $\underset{\approx}{\boldsymbol{W}}$ in Eq. (19.16).

Continuous functions are vectors!

Many seemingly unrelated things can be shown to be vectors by the above abstract, axiom-based, definition. Consider, for example, the following "discerning definition" for a set: V is the set of real continuous functions. Applying the above axioms shows that this is a vector space. Furthermore, if f and g are two continuous functions, then their inner product may be defined as

$$f \bullet g = \int_{-\infty}^{\infty} f(x)g(x)dx \qquad (19.18)$$

Note the similar structures: In (19.14), there is a summation over the dummy subscript, k. In (19.18), there is an integration over the dummy argument, x.

Once something is discovered to be an inner product vector space, a whole wealth of already-proved theorems becomes available. For example, the angle between two ordinary engineering vectors is defined by

$$\cos\theta = \frac{\underset{\sim}{\mathbf{a}} \bullet \underset{\sim}{\mathbf{b}}}{\sqrt{\underset{\sim}{\mathbf{a}} \bullet \underset{\sim}{\mathbf{a}}}\sqrt{\underset{\sim}{\mathbf{b}} \bullet \underset{\sim}{\mathbf{b}}}} \qquad (19.19)$$

We could likewise use Eq. (19.18) to define the "angle" between two continuous functions. Recall that the dyad between two ordinary vectors $\underset{\sim}{a}$ and $\underset{\sim}{b}$ is defined to be a new object such that $(\underset{\sim}{a} \otimes \underset{\sim}{b}) \bullet \underset{\sim}{v} = \underset{\sim}{a}(\underset{\sim}{b} \bullet \underset{\sim}{v})$ for all vectors $\underset{\sim}{v}$. Likewise, you can define a "dyad" between two real continuous functions f and g to be a new object such that

$$(f \otimes g) \bullet h = f(g \bullet h) \qquad (19.20)$$

Recall that a dyad $\underset{\sim}{a} \otimes \underset{\sim}{b}$ is a rudimentary tensor — i.e., a new object that needs two indices to describe it so that its indicial expression is $a_i b_j$. Likewise, the dyad between continuous functions is interpreted as a special kind of new object: a function of *two* variables (AKA **binary** function). Equation (19.20) tells us that $(f \otimes g)(x, y) = f(x)g(y)$. Recall that dyads were just special kinds of second-order tensors. Likewise $f \otimes g$ is a special kind of function of two variables — *separable*. Recall that the vector equation $\underset{\sim}{z} = \underset{\approx}{W} \bullet \underset{\sim}{v}$ is written in indicial form as

$$z_i = \sum_{j=1}^{3} W_{ij} v_j \qquad (19.21)$$

Analogously, given continuous functions $z(x)$, $v(y)$, and a binary function $W(x, y)$ the operation $z = W \bullet v$ would be interpreted as

$$z(x) = \int_{-\infty}^{\infty} W(x, y)v(y)dy \qquad (19.22)$$

Once you recognize this parallel structure between continuous functions and ordinary engineering vectors, you will find that reading advanced calculus textbooks becomes a lot easier. You will see that you *already know* many of the things that those books teach. The major distinction is that ordinary engineering vectors belong to finite-dimensional spaces (they only need 3 base vectors) whereas the space of continuous functions is *infinite* dimensional. As with ordinary vectors, you can define a basis for continuous functions, but the basis contains an infinite number of members. For example, a Taylor series is an expansion of a continuous function in terms of the power basis $\{1, x, x^2, \ldots\}$. A Fourier series is an expansion in terms a different (trigonometric) basis.

Tensors are vectors!

For applications in advanced continuum mechanics, it is extremely useful to recognize that second-order tensors form a *nine-dimensional* inner-product vector space. Using the rules for scalar multiplication and tensor addition, one can readily verify that the axioms A1 through A6 are satisfied by tensors. The inner product between two tensors is defined in Eq. (12.1) as

$$\underset{\approx}{\boldsymbol{A}}\boldsymbol{:}\underset{\approx}{\boldsymbol{B}} \equiv A_{ij}B_{ij} \tag{19.23}$$

which satisfies the inner product axioms, A7 and A8. This definition of the tensor inner product sums *pairwise* over all components just like Eq. (19.14). By pairwise, we mean that, for example, A_{12} is multiplied by B_{12}, not by B_{21}.

There are an infinite number of ways to generate a scalar from a binary operation between two tensors. For example, some authors wrongly refer to $\underset{\approx}{\mathbf{A}}\bullet\bullet\underset{\approx}{\mathbf{B}} = \text{tr}(\underset{\approx}{\boldsymbol{A}} \bullet \underset{\approx}{\boldsymbol{B}}) = A_{ij}B_{ji}$ as an inner product. However, this operation is *not* an inner product because it fails the positivity axiom A8. Specifically, consider

$$[\underset{\approx}{\boldsymbol{A}}] = \begin{bmatrix} 1 & -3 & 0 \\ 4 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19.24}$$

Then

$$\underset{\approx}{\mathbf{A}}\mathbf{:}\underset{\approx}{\mathbf{A}} = A_{11}A_{11} + A_{12}A_{12} + \ldots + A_{33}A_{33} = 1 + 9 + 16 + 4 + 1 = 31 > 0 \tag{19.25}$$

but

$$\underset{\approx}{\mathbf{A}}\bullet\bullet\underset{\approx}{\mathbf{A}} = A_{11}A_{11} + A_{12}A_{21} + \ldots + A_{33}A_{33} = 1 - 12 - 12 + 4 + 1 = -18 < 0, \tag{19.26}$$

which proves $\underset{\approx}{\mathbf{A}}\bullet\bullet\underset{\approx}{\mathbf{B}}$ is not acceptable as an inner product.

Both $\underset{\approx}{\mathbf{A}}:\underset{\approx}{\mathbf{A}}$ and $\underset{\approx}{\mathbf{A}}{\bullet\bullet}\underset{\approx}{\mathbf{A}}$ are scalar invariants of the tensor $\underset{\approx}{\mathbf{A}}$. Note that $\underset{\approx}{\mathbf{A}}{\bullet\bullet}\underset{\approx}{\mathbf{A}} = \mathrm{tr}(\underset{\approx}{\mathbf{A}}^2)$, which may be written in terms of the characteristic invariants as $I_1^2 - 2I_2$. For symmetric tensors, it turns out that $\underset{\approx}{\mathbf{A}}{\bullet\bullet}\underset{\approx}{\mathbf{A}} = \underset{\approx}{\mathbf{A}}:\underset{\approx}{\mathbf{A}}$. However, for nonsymmetric tensors, the inner product invariant $\underset{\approx}{\mathbf{A}}:\underset{\approx}{\mathbf{A}}$ is not generally expressible as a function of the characteristic invariants, and it is therefore a fourth *independent* invariant. To prove this last statement, consider

$$[\underset{\approx}{\mathbf{A}}] = \begin{bmatrix} 0 & \alpha & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad (19.27)$$

where α is an arbitrary scalar. The characteristic invariants of this tensor are all equal to zero. However, the inner product invariant is $\underset{\approx}{\mathbf{A}}:\underset{\approx}{\mathbf{A}} = \alpha^2$, which can take on different values, depending on the value of α. It therefore must not be expressible as a function $f(I_1, I_2, I_3)$, of the characteristic invariants. It is a fourth, wholly independent, invariant.

Vector subspaces

Consider a set S that is a proper subset of a vector set V. The set S is called a "subspace" if any linear combination of members of S is itself in S. A subspace S is itself a vector space because all of the vector space axioms that were proved for the larger vector set V continue to hold for set S. Proving that something is a subspace really only requires proving axioms A3 and A4 with axiom A2 replaced by the more restrictive discerning definition for membership in the subspace S.

Again, a subspace is itself a vector space. It is generally of lower dimension than V. In ordinary 3D space, a subspace is like a plane — any linear combination of vectors in a plane is itself in the plane. By contrast, consider the set of all unit vectors is *not* a subspace because a linear combination of two unit vectors is not itself a unit vector. Recalling that second-order tensors are themselves nine-dimensional vectors, we encounter *tensor* subspaces as well. The set of all symmetric second-order tensors is a tensor subspace because any linear combination of symmetric tensors is itself a symmetric tensor. The set of all orthogonal tensors is *not* a tensor subspace because a linear combination of orthogonal tensors is not itself orthogonal. The set of all transversely isotropic tensors is *not* a subspace, but the set (which we will denote $\mathbf{TI}_\mathrm{a}$) consisting of all tensors that are transversely isotropic about a *particular* (fixed) direction $\underset{\sim}{\boldsymbol{a}}$ *does* form a subspace. Specifically, referring to Eq. (18.59) we note that the tensors, $\underset{\approx}{\boldsymbol{I}}$ $\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{a}}$, and $-\underset{\approx}{\varepsilon} \bullet \underset{\sim}{\boldsymbol{a}}$ form a basis for $\mathbf{TI}_\mathrm{a}$. This basis, is not however, orthogonal because $\underset{\approx}{\boldsymbol{I}}:(\underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{a}}) \neq 0$. Referring instead to Eq. (18.58), we could alternatively define

$$\begin{aligned} \underset{\approx}{\boldsymbol{B}}_1 &= \underset{\approx}{\boldsymbol{I}} - \underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{a}} \\ \underset{\approx}{\boldsymbol{B}}_2 &= \underset{\sim}{\boldsymbol{a}}\underset{\sim}{\boldsymbol{a}} \\ \underset{\approx}{\boldsymbol{B}}_3 &= -\underset{\approx}{\varepsilon} \bullet \underset{\sim}{\boldsymbol{a}} \end{aligned} \qquad (19.28)$$

These tensors form an *orthogonal* basis for $\mathbf{TI}_\mathrm{a}$. The basis is orthogonal because $\underset{\approx}{\boldsymbol{B}}_i \boldsymbol{:} \underset{\approx}{\boldsymbol{B}}_j = 0$ if $i \neq j$. However,

$$\underset{\approx}{\boldsymbol{B}}_1 \boldsymbol{:} \underset{\approx}{\boldsymbol{B}}_1 = 2 \qquad \underset{\approx}{\boldsymbol{B}}_2 \boldsymbol{:} \underset{\approx}{\boldsymbol{B}}_2 = 1 \qquad \underset{\approx}{\boldsymbol{B}}_3 \boldsymbol{:} \underset{\approx}{\boldsymbol{B}}_3 = 2 \tag{19.29}$$

Hence, this basis is orthogonal, but not normalized. We *could* naturally divide $\underset{\approx}{\boldsymbol{B}}_1$ by $\sqrt{2}$ to normalize it, but doing so would make it no longer a projection. Normalizing the basis would be wise if our applications frequently required the *tensor inner product* of transversely isotropic tensors. Most engineering applications require just the *single dot* (composition) product between two transversely isotropic tensors, in which case leaving the basis as projectors is more convenient to utilize the fact that $\underset{\approx}{\boldsymbol{B}}_1 \bullet \underset{\approx}{\boldsymbol{B}}_1 = \underset{\approx}{\boldsymbol{B}}_1$. Incidentally, $\underset{\approx}{\boldsymbol{B}}_2$ is also a projector, but $\underset{\approx}{\boldsymbol{B}}_3$ is not.

Example: Let $\underset{\sim}{\boldsymbol{c}}$ be a fixed vector. Let S denote the set of all vectors $\underset{\sim}{\boldsymbol{x}}$ for which $\underset{\sim}{\boldsymbol{c}} \bullet \underset{\sim}{\boldsymbol{x}} = 0$. Is S a linear subspace? The answer is "yes" because any linear combination of members of S will itself belong to S. To see why, let $\underset{\sim}{\boldsymbol{x}}_1$ and $\underset{\sim}{\boldsymbol{x}}_2$ belong to S then, by definition of membership, we know

$$\underset{\sim}{\boldsymbol{c}} \bullet \underset{\sim}{\boldsymbol{x}}_1 = 0 \quad \text{and} \quad \underset{\sim}{\boldsymbol{c}} \bullet \underset{\sim}{\boldsymbol{x}}_2 = 0 \tag{19.30}$$

Now consider a linear combination of these two vectors, $\alpha_1 \underset{\sim}{\boldsymbol{x}}_1 + \alpha_2 \underset{\sim}{\boldsymbol{x}}_2$. Does this linear combination belong to S? The answer is "yes" if and only if

$$\underset{\sim}{\boldsymbol{c}} \bullet (\alpha_1 \underset{\sim}{\boldsymbol{x}}_1 + \alpha_2 \underset{\sim}{\boldsymbol{x}}_2) = 0 \tag{19.31}$$

Expanding this out using Eq. 19.30 shows that equality does hold. Therefore S is a subspace.

Example: commuting space. Let $\underset{\approx}{\boldsymbol{C}}$ denote a fixed symmetric tensor. Let S denote the set of all tensors $\underset{\approx}{\boldsymbol{X}}$, both symmetric and non-symmetric, that commute with $\underset{\approx}{\boldsymbol{C}}$. By this we mean

$$\underset{\approx}{\boldsymbol{X}} \text{ belongs to } S \text{ if and only if } \underset{\approx}{\boldsymbol{C}} \bullet \underset{\approx}{\boldsymbol{X}} = \underset{\approx}{\boldsymbol{X}} \bullet \underset{\approx}{\boldsymbol{C}} \tag{19.32}$$

To prove that this is a linear subspace, you must prove that

$$\underset{\approx}{\boldsymbol{C}} \bullet (\alpha_1 \underset{\approx}{\boldsymbol{X}}_1 + \alpha_2 \underset{\approx}{\boldsymbol{X}}) = (\alpha_1 \underset{\approx}{\boldsymbol{X}}_1 + \alpha_2 \underset{\approx}{\boldsymbol{X}}) \bullet \underset{\approx}{\boldsymbol{C}}$$
$$\text{for all tensors } \underset{\approx}{\boldsymbol{X}}_1 \text{ and } \underset{\approx}{\boldsymbol{X}}_2 \text{ belonging to } S \tag{19.33}$$

Multiplying this out, using the fact that Eq. 19.32 applies for $\underset{\approx}{\boldsymbol{X}}_1$ and $\underset{\approx}{\boldsymbol{X}}_2$ shows that this equation is true for the linear combination as well. Hence, S is a linear subspace.

The far more interesting questions are: How do you characterize members of S more explicitly and what is the dimension of S? To answer, let's write out the membership criterion in terms of the principal basis for $\underset{\approx}{\boldsymbol{C}}$ in which case, the matrix form of Eq. (19.32) becomes

$$\begin{bmatrix} c_1X_{11} & c_1X_{12} & c_1X_{13} \\ c_2X_{21} & c_2X_{22} & c_2X_{23} \\ c_3X_{31} & c_3X_{32} & c_3X_{33} \end{bmatrix} = \begin{bmatrix} c_1X_{11} & c_2X_{12} & c_3X_{13} \\ c_1X_{21} & c_2X_{22} & c_3X_{23} \\ c_1X_{31} & c_2X_{32} & c_3X_{33} \end{bmatrix} \tag{19.34}$$

The diagonal components are automatically satisfied for any tensor $\underset{\approx}{X}$. The off-diagonal components require

$$(c_1 - c_2)X_{12} = 0 \tag{19.35}$$

$$(c_2 - c_1)X_{21} = 0 \tag{19.36}$$

$$(c_2 - c_3)X_{23} = 0\text{, etc.} \tag{19.37}$$

If all of the eigenvalues of $\underset{\approx}{C}$ are distinct, then these off-diagonal constraints require that the off-diagonals of $\underset{\approx}{X}$ with respect to the principal basic for $\underset{\approx}{C}$ must all be zero — in other words, the set S is the set of all tensors with the same principal basis as $\underset{\approx}{C}$ and the dimension of this space is three (you only need to specify three eigenvalues to identify any member of S). When $\underset{\approx}{C}$ has distinct eigenvalues, the basis for its commuting space consists of simply the three eigenprojectors for $\underset{\approx}{C}$; namely,

$$\underset{\sim}{p}_1\underset{\sim}{p}_1,\ \underset{\sim}{p}_2\underset{\sim}{p}_2,\ \underset{\sim}{p}_3\underset{\sim}{p}_3 \quad \text{when } c_1 \neq c_2, \quad c_2 \neq c_3, \quad \text{and} \quad c_3 \neq c_1 \tag{19.38}$$

If, on the other hand, $\underset{\approx}{C}$ has only two distinct eigenvalues, then exactly two of the off-diagonals of $\underset{\approx}{X}$ become unconstrained. If, for example, $c_1 = c_2 \neq c_3$, then Eqs. (19.35) and (19.36) are automatically satisfied and the most general form for a member of the commuting space will have components with respect to the principal basis of $\underset{\approx}{C}$ of the form

$$\begin{bmatrix} X_{11} & X_{12} & 0 \\ X_{21} & X_{22} & 0 \\ 0 & 0 & X_{33} \end{bmatrix} \tag{19.39}$$

If $\underset{\approx}{X}$ is symmetric, note that the upper 2×2 matrix can be symmetrized, and the resulting principal basis for $\underset{\approx}{X}$ will also be a principal basis for $\underset{\approx}{C}$ because any two vectors in the 1-2 plane will be eigenvectors of $\underset{\approx}{C}$. Thus, whenever a symmetric tensor $\underset{\approx}{X}$ commutes with $\underset{\approx}{C}$, a commonly shared principal basis can always be found for the two tensors. This does *not* mean that the commuting space S is again reduced to three dimensions. To the contrary, the shared basis will be different for a *different* $\underset{\approx}{X}^*$ in S. When $\underset{\approx}{C}$ has $c_1 = c_2 \neq c_3$, the basis for the commuting space is

$$\underset{\sim}{p}_1\underset{\sim}{p}_1,\ \underset{\sim}{p}_2\underset{\sim}{p}_2,\ \underset{\sim}{p}_3\underset{\sim}{p}_3,\ \underset{\sim}{p}_1\underset{\sim}{p}_2,\ \underset{\sim}{p}_2\underset{\sim}{p}_1 \quad \text{when } c_1 = c_2 \neq c_3 \tag{19.40}$$

Finally, when $\underset{\approx}{C}$ has a *triple* eigenvalue, then it must be proportional to the identity, and therefore *any* tensor will commute with $\underset{\approx}{C}$. In this case, the commuting space is the entirety of tensor space itself, which is nine-dimensional.

Commuting tensors can be considered in a more direct notation context. The **characteristic space** associated with an eigenvalue is the span of all eigenvectors associated with that eigenvalue. The "commutation theorem" [24] states that, if $\underset{\approx}{\boldsymbol{C}} \bullet \underset{\approx}{\boldsymbol{X}} = \underset{\approx}{\boldsymbol{X}} \bullet \underset{\approx}{\boldsymbol{C}}$, then if $\underset{\sim}{\boldsymbol{v}}$ belongs to a characteristic space of $\underset{\approx}{\boldsymbol{C}}$, then $\underset{\approx}{\boldsymbol{X}} \bullet \underset{\sim}{\boldsymbol{v}}$ also belongs to that same characteristic space. This does not imply that $\underset{\sim}{\boldsymbol{v}}$ is necessarily an eigenvector of $\underset{\approx}{\boldsymbol{X}}$, but it can be used to prove that there does exist at least on principal triad that is shared by $\underset{\approx}{\boldsymbol{C}}$ and $\underset{\approx}{\boldsymbol{X}}$.

Subspaces and the projection theorem

Whenever a subspace S is discovered, it is almost always wise to identify the operation that will project vectors in V into parts that are in S and orthogonal to S. Many seemingly unrelated theorems are all in fact applications of the projection theorem. Consider, for example, the set of all *even* continuous functions — i.e., they have the property $f(-x) = f(x)$. This set is subspace because any linear combination of even functions will also be even. A famous theorem states that any function can be written as the sum of an even plus an odd function. This theorem is an application of the projection theorem. Note that odd functions are orthogonal to even functions because the integral in Eq. (19.18) will be zero if f is even and g is odd.

The set of all matrices that have a zero trace is a subspace (because any linear combination of "traceless" tensors will itself be traceless). An oft-used theorem states that any matrix can be written as the sum of its isotropic and deviatoric (traceless) parts. This is the projection theorem again. Likewise any matrix can be written as the sum of its symmetric plus antisymmetric parts. This is the projection theorem.

Abstract contraction and swap (exchange) operators*

Virtually all operations in tensor algebra can be written using a rather arcane — but nevertheless conceptually useful — generalized notation in which the arguments are first multiplied dyadically and then operated on by either contraction or swap operators, as defined below. These operators are almost never mentioned explicitly in practice, but it is often useful to visualize simpler operations in terms of contractions and swaps in order to determine properties of the operations.

Any vector $\underset{\sim}{\boldsymbol{v}}$ is expressible as components v_i times base vectors $\underset{\sim}{\boldsymbol{e}}_j$. Any second-order tensor $\underset{\approx}{\boldsymbol{T}}$ is expressible as components T_{ij} times base dyads $\underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j$. In general, any N^{th}-order tensor may be expressed as a sum of components (having a total of N indices) times N dyadically multiplied base vectors.

* This section is rather mathematical and it may be skipped without loss in continuity.

The contraction operator $C_{\mathtt{I}}^{\mathtt{J}}$ is defined to dot the $\mathtt{I}^{\text{th}}$ base vector with the $\mathtt{J}^{\text{th}}$ base vector. Here, $\mathtt{I}$ and $\mathtt{J}$ are not tensor indices — they merely indicate the positional location of a dyadically multiplied base vectors. The contraction operation is well-defined only when operating on second- and higher-order tensors. For example

$$C_1^2(\underset{\sim}{a}\underset{\sim}{b}) = C_1^2(a_i b_j \underbrace{\underset{\sim}{e}_i\underset{\sim}{e}_j}_{\text{dot}}) = a_i b_j \underset{\sim}{e}_i \bullet \underset{\sim}{e}_j = a_i b_j \delta_{ij} = a_i b_i = \underset{\sim}{a} \bullet \underset{\sim}{b}$$

$$C_1^2(\underset{\approx}{T}) = C_1^2(T_{ij} \underbrace{\underset{\sim}{e}_i\underset{\sim}{e}_j}_{\text{dot}}) = T_{ij}\underset{\sim}{e}_i \bullet \underset{\sim}{e}_j = T_{ij}\delta_{ij} = T_{ii} = \text{tr}\underset{\approx}{T}$$

$$C_1^2\left(\underset{\sim}{\underset{\approx}{\beta}}\right) = C_1^2(\beta_{ijk} \underbrace{\underset{\sim}{e}_i\underset{\sim}{e}_j}_{\text{dot}}\underset{\sim}{e}_k) = \beta_{ijk}\underset{\sim}{e}_i \bullet \underset{\sim}{e}_j\underset{\sim}{e}_k = \beta_{iik}\underset{\sim}{e}_k$$

$$C_1^3\left(\underset{\sim}{\underset{\approx}{\beta}}\right) = C_1^3(\beta_{ijk} \underbrace{\underset{\sim}{e}_i\underset{\sim}{e}_j\underset{\sim}{e}_k}_{\text{dot}}) = \beta_{ijk}(\underset{\sim}{e}_i \bullet \underset{\sim}{e}_k)\underset{\sim}{e}_j = \beta_{iji}\underset{\sim}{e}_j$$

$$C_2^4\left(\underset{\approx}{\underset{\approx}{H}}\right) = C_2^4(H_{ijkl}\underset{\sim}{e}_i \underbrace{\underset{\sim}{e}_j\underset{\sim}{e}_k\underset{\sim}{e}_l}_{\text{dot}}) = H_{ijkl}(\underset{\sim}{e}_j \bullet \underset{\sim}{e}_l)\underset{\sim}{e}_i\underset{\sim}{e}_k = H_{ijkj}\underset{\sim}{e}_i\underset{\sim}{e}_k \tag{19.41}$$

The contraction operator reduces the order of the operand by two. If, for example, the operand is a second-order tensor, then the result is a zero-order tensor, which is a scalar. As seen above, the contraction of a third-order tensor gives a first-order tensor, which is a vector. The contraction of a fourth-order tensor is a second-order tensor.

The generalized operator $C_{\mathtt{IK}}^{\mathtt{JL}}$ dots the $\mathtt{I}^{\text{th}}$ base vector with the $\mathtt{J}^{\text{th}}$ vector and the $\mathtt{K}^{\text{th}}$ base vector with the $\mathtt{L}^{\text{th}}$ vector. This operation reduces the order by four. To be well defined, all four of the indices, $\mathtt{I}$, $\mathtt{J}$, $\mathtt{K}$, and $\mathtt{L}$, must have distinct values. The operator is well-defined for tensors of fourth-and higher order. For example,

$$C_{13}^{42}(\underset{\approx}{\underset{\approx}{H}}) = C_{13}^{42}(H_{ijkl}\underset{\sim}{e}_i\underset{\sim}{e}_j\underset{\sim}{e}_k\underset{\sim}{e}_l) = H_{ijkl}(\underset{\sim}{e}_i \bullet \underset{\sim}{e}_l)(\underset{\sim}{e}_j \bullet \underset{\sim}{e}_k) = H_{ikki} \tag{19.42}$$

Incidentally, note that $C_{13}^{42}(\ )$ is not the same thing as $C_1^4(C_1^2(\ ))$. In particular,

$$C_{13}^{42}(\underset{\approx}{\underset{\approx}{H}}) = C_1^2(C_1^4(\underset{\approx}{\underset{\approx}{H}})) = C_1^2(C_3^2(\underset{\approx}{\underset{\approx}{H}})). \tag{19.43}$$

Higher order contraction operators such as $C_{\mathtt{IKM}}^{\mathtt{JLN}}$ are similarly defined.

The **swap** (or “**exchange**”) operator $X_{\mathtt{I}}^{\mathtt{J}}$ is defined to swap the $\mathtt{I}^{\text{th}}$ base vector with the $\mathtt{J}^{\text{th}}$ vector. This operation does not affect the order of the operand. The swap operator is well-defined only when operating on second- and higher-order tensors. For example

$$X_1^2(\underset{\approx}{T}) = X_1^2(T_{ij}\underbrace{\underset{\sim}{e}_i\underset{\sim}{e}_j}_{\text{swap}}) = T_{ij}\underset{\sim}{e}_j\underset{\sim}{e}_i = \underset{\approx}{T}^T$$

$$X_1^2\left(\underset{\sim}{\underset{\approx}{\beta}}\right) = X_1^2(\beta_{ijk}\underbrace{\underset{\sim}{e}_i\underset{\sim}{e}_j}_{\text{swap}}\underset{\sim}{e}_k) = \beta_{ijk}\underset{\sim}{e}_j\underset{\sim}{e}_i\underset{\sim}{e}_k$$

$$X_1^3\left(\underset{\sim}{\underset{\approx}{\beta}}\right) = X_1^3(\beta_{ijk}\underbrace{\underset{\sim}{e}_i\underset{\sim}{e}_j\underset{\sim}{e}_k}_{\text{swap}}) = \beta_{ijk}\underset{\sim}{e}_k\underset{\sim}{e}_j\underset{\sim}{e}_i \tag{19.44}$$

The generalized swap operator $X_{\mathtt{IK}}^{\mathtt{JL}}$ swaps the $\mathtt{I}^{\text{th}}$ base vector with the $\mathtt{J}^{\text{th}}$ vector and the $\mathtt{K}^{\text{th}}$ base vector with the $\mathtt{L}^{\text{th}}$ vector. To be well defined, all four of the indices, $\mathtt{I}$, $\mathtt{J}$, $\mathtt{K}$, and $\mathtt{L}$, must have distinct values. The operator is well-defined for tensors of fourth-and higher order. For example,

$$X_{13}^{42}\left(\underset{\approx}{\underset{\approx}{H}}\right) = X_{13}^{42}(H_{ijkl}\underbrace{\underset{\sim}{e}_i\overbrace{\underset{\sim}{e}_j\underset{\sim}{e}_k}^{\text{swap}}\underset{\sim}{e}_l}_{\text{swap}}) = H_{ijkl}\underset{\sim}{e}_l\underset{\sim}{e}_k\underset{\sim}{e}_j\underset{\sim}{e}_i \tag{19.45}$$

Higher order swap operators such as $X_{\mathtt{IKM}}^{\mathtt{JLN}}$ are similarly defined.

The contraction operator and the swap operator are almost never mentioned explicitly in publications but they are nonetheless very useful in generating theorems because they are both linear operations. Namely,

$$C_{\mathtt{I}}^{\mathtt{J}}(\alpha A + B) = \alpha C_{\mathtt{I}}^{\mathtt{J}}(A) + C_{\mathtt{I}}^{\mathtt{J}}(B) \text{ for all scalars } \alpha$$
$$\text{and for all tensors } A \text{ and } B \text{ of arbitrary (but equal) order (} \geq 2\text{)}. \tag{19.46}$$

$$X_{\mathtt{I}}^{\mathtt{J}}(\alpha A + B) = \alpha X_{\mathtt{I}}^{\mathtt{J}}(A) + X_{\mathtt{I}}^{\mathtt{J}}(B) \text{ for all scalars } \alpha$$
$$\text{and for all tensors } A \text{ and } B \text{ of arbitrary (but equal) order.} \tag{19.47}$$

The last property is particularly useful for performing integrals of quantities that differ from each other only in the ordering of their indices. Noting that any permutation of indices may be obtained through a sequence of swap operations, we recognize that only one of our integrals need be computed. Linearity of the swap operation allows us to immediately write down the solution for all of the integrals once we have found the solution for the first one.

At the beginning of this section, we mentioned that most tensor operations may be written as a sequence of either swap or contraction operations acting on dyadically multiplied operands. For example

$$\underset{\approx}{A} \bullet \underset{\sim}{x} = C_2^3(\underset{\approx}{A} \otimes \underset{\sim}{x}) = C_1^3(\underset{\sim}{x} \otimes \underset{\approx}{A}) = C_1^2(\underset{\sim}{x} \otimes X_1^2(\underset{\approx}{A})) \tag{19.48}$$

and

$$\underset{\sim}{\boldsymbol{u}} \times \underset{\sim}{\boldsymbol{v}} = C^{45}_{23}\left(\underset{\approx}{\varepsilon}\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}}\right) \tag{19.49}$$

In the above equation, $\underset{\approx}{\varepsilon}$ is the permutation tensor given by $\varepsilon_{ijk}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\underset{\sim}{\boldsymbol{e}}_k$, where ε_{ijk} is the permutation symbol defined in Eq. (3.76). The operand $\underset{\approx}{\varepsilon}\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}}$ is a fifth order tensor having $ijklm$ components given by $\varepsilon_{ijk}u_l v_m$. The contraction operator C^{45}_{23} reduces the order from five down to one by contracting the 2nd index with the 4th index and the 3rd index with the 5th index to obtain $\varepsilon_{ijk}u_j v_k$, which is indeed the indicial expression for the i^{th} component of $\underset{\sim}{\boldsymbol{u}} \times \underset{\sim}{\boldsymbol{v}}$. It may appear that we have taken a fairly simple operation (the cross product) and written it in an insanely complicated manner.

Keep in mind: we are *not* advocating the use of the contraction and swap operations in published analyses. We merely want to drive home the point that virtually all tensor operations can be expressed using contraction and swap operators. Contractions and swaps are particularly useful when applied to tensor-integral calculus. Specifically, suppose that $f(\underset{\sim}{\boldsymbol{x}})$ denotes some n^{th}-order tensor valued function of a vector $\underset{\sim}{\boldsymbol{x}}$. If C and X denote any contraction and swap operation applicable to $f(\underset{\sim}{\boldsymbol{x}})$, then

$$C\iiint f(\underset{\sim}{\boldsymbol{x}})dV = \iiint C[f(\underset{\sim}{\boldsymbol{x}})]dV \tag{19.50}$$

$$X\iiint f(\underset{\sim}{\boldsymbol{x}})dV = \iiint X[f(\underset{\sim}{\boldsymbol{x}})]dV \tag{19.51}$$

In other words, contraction and swap operations commute with integration. This means that the best context in which to derive new theorems is to do it for dyadically multiplied operands whenever possible. From this generalized context, a host of new theorems “fall out” as corollaries. For example, a very general form of the Gauss-Ostrogradski theorem states that the integral over a closed surface of the outward normal $\underset{\sim}{\boldsymbol{n}}$ multiplied dyadically by a tensor of arbitrary order can be transformed to the volume integral of the gradient of that tensor.

$$\iint \underset{\sim}{\boldsymbol{n}}(\quad)dA = \iiint \nabla(\quad)dV \tag{19.52}$$

Here, the notation $\underset{\sim}{\boldsymbol{n}}(\quad)$ denotes the outward unit normal $\underset{\sim}{\boldsymbol{n}}$ to the integration surface multiplied some quantity $(\quad)$ of arbitrary tensor order, and $\nabla(\quad)$ denotes the gradient of that tensor.

Because the contraction operation commutes with integration, one can immediately write down the classic “divergence” theorem by taking the C^2_1 contraction of both sides of Eq. (19.52) to obtain

$$\iint \underset{\sim}{\boldsymbol{n}} \bullet (\quad)dA = \iiint \nabla \bullet (\quad)dV \tag{19.53}$$

Another corollary is obtained by inserting the cross product into Eq. (19.52), which we know we can do because we know that a cross product is expressible in terms of contractions (which in turn commute with integration). Thus,

$$\iint \underset{\sim}{\boldsymbol{n}} \times (\quad) dA = \iiint \nabla \times (\quad) dV \tag{19.54}$$

Incidentally, despite the presence of the cross product, this theorem is *not* equivalent to Stokes theorem.

Consider a particular indicial form of Eq. (19.52) in which the operand is a second-order tensor:

$$\iint n_i T_{jk} dA = \iiint \frac{\partial}{\partial x_i} T_{jk} dV \tag{19.55}$$

The swap operator X_I^J is linear and therefore commutes with integration. Thus, in addition to the above equation, we also have an immediate set of corollaries in which the indices in the above equation may be arbitrarily permuted. Likewise, the contraction operator C_I^J is linear and may be commuted with integration. Thus, any of the indices in the above equation may be contracted.

The contraction tensor. Since the contraction operator $C_1^2(\underset{\approx}{\boldsymbol{T}})$ is linear, we know from Eq. (9.7) that there must exist a second-order tensor $\underset{\approx}{\boldsymbol{C}}$ such that

$$C_1^2(\underset{\approx}{\boldsymbol{T}}) = \underset{\approx}{\boldsymbol{C}}\boldsymbol{:}\underset{\approx}{\boldsymbol{T}} = C_{ij} T_{ij} \tag{19.56}$$

We know from Eq. (19.41) that

$$C_1^2(\underset{\approx}{\boldsymbol{G}}) = \mathrm{tr}\underset{\approx}{\boldsymbol{T}} = T_{kk} = \delta_{ij} T_{ij} \tag{19.57}$$

Comparing the above two equations, we note that $\underset{\approx}{\boldsymbol{C}}$ is simply the identity tensor. Hence, the contraction operation can be viewed as the identity tensor double dotted into two specified base vectors. This point of view is useful when considering differentiation. The identity is a constant tensor and its derivative is therefore zero. It is often useful conceptually utilize the contraction in gradient identities. For example,

$$\nabla \bullet \underset{\sim}{\boldsymbol{w}} = \underset{\approx}{\boldsymbol{I}}\boldsymbol{:}\nabla \underset{\sim}{\boldsymbol{w}} \tag{19.58}$$

The contraction is extremely useful in integration identities. For example

$$\int\limits_{\Omega} \nabla \bullet \underset{\sim}{\boldsymbol{w}} dV = \int\limits_{\Omega} \underset{\approx}{\boldsymbol{I}}\boldsymbol{:}(\nabla \underset{\sim}{\boldsymbol{w}}) dV = \underset{\approx}{\boldsymbol{I}}\boldsymbol{:}\int\limits_{\Omega} \nabla \underset{\sim}{\boldsymbol{w}} dV \tag{19.59}$$

These formulas show that knowing the complete gradient tensor $\nabla \underset{\sim}{\boldsymbol{w}}$ is really the most useful. You can just take the trace of this total gradient tensor to immediately obtain the divergence. Likewise, you can take the (negative) axial part of the total gradient tensor to obtain the curl:

$$-\mathrm{axial}\nabla \underset{\sim}{\boldsymbol{w}} = \underset{\approx}{\varepsilon}\boldsymbol{:}\nabla \underset{\sim}{\boldsymbol{w}} = \nabla \times \underset{\sim}{\boldsymbol{w}} \tag{19.60}$$

The swap tensor. Recall from Eq. (19.44) that the effect of the swap operation on a second-order tensor $\underset{\approx}{\boldsymbol{T}}$ is to simply produce the transpose of that tensor:

$$X_1^2(\underset{\approx}{T}) = \underset{\approx}{\boldsymbol{T}}^T \qquad (19.61)$$

This is a linear operation transforming a tensor $\underset{\approx}{\boldsymbol{T}}$ into a new tensor $\underset{\approx}{\boldsymbol{T}}^T$. Therefore, this simple operation must be expressible in terms of a fourth-order swap tensor $\underset{\substack{\approx\\\approx}}{\boldsymbol{X}}$ such that

$$\underset{\approx}{\boldsymbol{T}}^T = \underset{\substack{\approx\\\approx}}{\boldsymbol{X}}:\underset{\approx}{\boldsymbol{T}} \qquad (19.62)$$

Writing this out in indicial notation reveals that

$$\underset{\substack{\approx\\\approx}}{\boldsymbol{X}} = \delta_{in}\delta_{jm}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\underset{\sim}{\boldsymbol{e}}_m\underset{\sim}{\boldsymbol{e}}_n \qquad (19.63)$$

This tensor can be regarded as the fourth-order transpose operator. Note that it is the same as the fourth-order identity tensor except the last two base vectors are swapped (or, equivalently, the last two indices in the indicial expression are swapped).

20. Vector and Tensor Visualization

Suppose you are given a 2D scalar field (i.e., scalars that vary in space in a known manner). For example, the altitude of a mountain could be regarded as a scalar that varies over a 2D space. Then some conventional ways to visually display this altitude is with a contour map, a 3D elevation, and a color density map. If, for example, the altitude of the mountain, z, is given by*

$$z = 1 + \sin(xy) \tag{20.1}$$

then these three strategies would yield the plots shown in Fig. 20.1. Scalar fields in 3D are typically rendered using these basic techniques on 2D slice planes of the body (more generally, these methods can be used on any 2D surface — for example, a color density plot could be shown on the surface of a machine part)

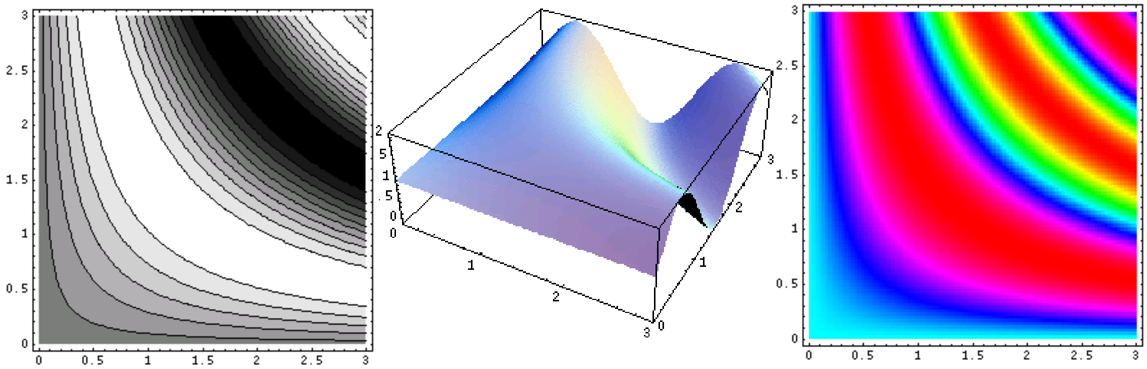
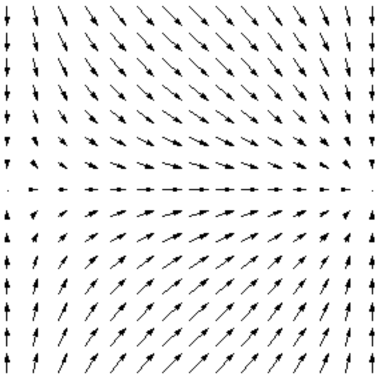


Figure 20.1. Three types of visualization for scalar fields. (a) contour plot, (b) elevation plot, and (c) Color density plot [ranges through red-yellow-green-cyan-blue-magenta-red as the elevation ranges from minimum to maximum].



Now suppose you are given a *vector* field in 2D. The standard approach is to plot the vectors as a family of arrows in 2D space, as shown in the inset.† You can alternatively generate streamlines that “flow” with the vector field. In three dimensions, you can generate vector plots across any 2D surface, as was suggested in the case of a scalar field. However, in 3D, the vectors will not necessarily lie in the plot plane — they might have components pointing out of the plane. Consequently, if these sorts of plots are generated on, say, the surface of a

* Incidentally, these three plots can be generated by the following mathematica commands: `ContourPlot`, `Density Plot`, and `Plot3D`.

† This plot was generated using the mathematica command, `<<Graphics‘PlotField‘;PlotVectorField[{Sin[x],Cos[y]},{x,0,Pi},{y,0,Pi}]`

machine part, they can have the appearance of a hedgehog (in fact many people call these plots by that name.). If desired, the arrows can be colored in proportion to the magnitude of the vector. Another, less frequently seen, method for visualizing vectors is to assign the colors Red, Green, and Blue to vary in proportion to the three components of the vector v_1, v_2, and v_3. Then an ordinary scalar density plot could be generated using the mixture of colors. Then blue regions would indicate regions where v_3 is large in comparison to the other components. Yellow (which is a mixture of equal parts of red and green) would indicate regions where $v_1 = v_2$ and $v_3 = 0$).

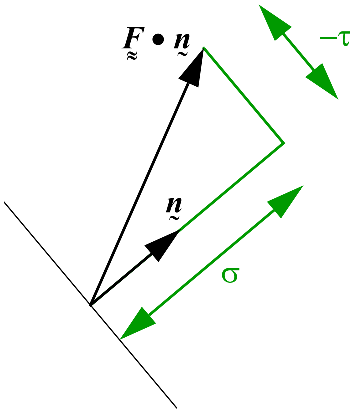
Clearly, visualizing vector fields is considerably more complicated than visualizing scalar fields. Vectors only have three components. How in the world can we visualize *tensors*, which have NINE components?!? We will describe here two standard methods for visualizing tensors.

Mohr's circle for 2D tensors

GOAL: Describe how to generate Mohr's circle for a 2x2 matrix that is not necessarily symmetric.

Consider a 2D second order tensor whose components are of the form,

$$\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{20.2}$$


$\underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{n}}$
$-\tau$
$\underset{\sim}{\boldsymbol{n}}$
σ

. A tensor takes on meaning only when it acts on a vector. For visualizing a tensor, the most crucial question is how that tensor transforms a *unit* vector $\underset{\sim}{\boldsymbol{n}}$. In the inset, we show a unit vector $\underset{\sim}{\boldsymbol{n}}$ and its transformation $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{n}}$. We define σ to be the normal component of the transformed vector and τ to be the shear component (with a sign given by the *left-hand-rule*, meaning that it is positive if the transformed vector tends to shear the plane in a *clockwise* direction).

To construct Mohr's circle, first define two points, an "H-point" and a "V-point" by

$$\text{H: } (F_{11}, -F_{21}) \tag{20.3}$$

$$\text{V: } (F_{22}, F_{12}) \tag{20.4}$$

Note that these two points are the (σ, τ) pairs associated with $\underset{\sim}{\boldsymbol{n}} = \underset{\sim}{\boldsymbol{E}}_1$ and $\underset{\sim}{\boldsymbol{n}} = \underset{\sim}{\boldsymbol{E}}_2$, respectively. In general, differently oriented normals will have different values of the (σ, τ) pairs. As the plane's orientation angle θ varies, then so will σ and τ. Specifically, an arbitrary normal may be expressed in the form

$$\{\underset{\sim}{\boldsymbol{n}}\} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{20.5}$$

and the transformed vector therefore has components

$$\{\underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{n}}\} = [F]\{n\} = \begin{bmatrix} F_{11}\cos\theta + F_{12}\sin\theta \\ F_{21}\cos\theta + F_{22}\sin\theta \end{bmatrix} \tag{20.6}$$

from which it follows that

Need to finish this section

$$\tau = \tag{20.7}$$

and

$$\sigma = \tag{20.8}$$

A phase space curve of τ vs. σ can be constructed by plotting (σ, τ) pairs for various orientation angles. This plot is called a Mohr diagram. Naturally, because θ is cyclic, the curve would have to be some sort of a closed loop. It turns out that this closed curve is in fact a *circle* and that planes separated by a distance $\Delta\theta$ in physical space will be separated by an angle $2\Delta\theta$ in the Mohr diagram. Recalling that the H and V points correspond to normals that differ by $90°$, this means that on the Mohr plane, the H and V points will be separated from each other by $180°$. In other words, the H and V points will lie on *diametrically opposite* locations of the Mohr's circle. Consequently, the center of Mohr's circle must be exactly between the H and V points. This is enough information to draw the Mohr's circle.

Moving an angular distance α, measured from the H-point, corresponds to σ and τ values on Mohr's circle which are the normal and shear components of the transformed vector $\underset{\approx}{\boldsymbol{F}} \bullet \underset{\sim}{\boldsymbol{n}}$ for which $\underset{\sim}{\boldsymbol{n}}$ is at an angle $\alpha/2$ measured from the horizontal. This halving or doubling angles when moving back and forth between the Mohr diagram and the physical plane can be confusing, and there is a really kool way to get around this problem by introducing a third point — called the pole point — on the Mohr diagram. This point is found by passing a horizontal line through the H point and a vertical line through the V point. These lines will intersect at the pole point P. You can measure angles in the Mohr plane from the pole point, and those angles will precisely equal the angle in the physical plane.

"You shouldn't put down a loser, Cindy... you might be one yourself some day."
— Carol Brady, The Brady Bunch

21. Vector/tensor differential calculus

Stilted definitions of grad, div, and curl

Up to now, this book has focused on vector and tensor *algebra*, not calculus. A standard rigorous (and therefore stilted) development of tensor calculus, as found in any good tensor analysis textbook, often begins by defining

$$\text{grad}\phi \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{n}} \phi \, dS \tag{21.1}$$

$$\text{grad}\underset{\sim}{\boldsymbol{u}} \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{n}} \underset{\sim}{\boldsymbol{u}} \, dS \tag{21.2}$$

$$\text{div}\underset{\sim}{\boldsymbol{u}} \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{n}} \bullet \underset{\sim}{\boldsymbol{u}} \, dS \tag{21.3}$$

$$\text{curl}\underset{\sim}{\boldsymbol{u}} \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{n}} \times \underset{\sim}{\boldsymbol{u}} \, dS \tag{21.4}$$

Here, V is the volume of an arbitrary region Ω in space, and $\partial\Omega$ is its boundary with an outward unit normal $\underset{\sim}{\boldsymbol{n}}$. Furthermore, dS is the area element, so the above integrals are really *double* integrals. Other ways to define these derivative operations (discussed later) do not require calling upon geometrical concepts such as volume or area. When these alternative definitions are adopted, the above equations come out as *results*. Alternative definitions of differentiation (discussed later) are superior if you are interested in evaluating variations with respect to something *other* than the spatial location $\underset{\sim}{\boldsymbol{x}}$.

In terms of rectangular Cartesian coordinate systems (RCS), these derivative operations can be computed by

$$\text{grad}\phi = \underset{\sim}{\boldsymbol{e}}_k \frac{\partial\phi}{\partial x_k} \tag{21.5}$$

$$\text{grad}\underset{\sim}{\boldsymbol{u}} = \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \frac{\partial u_j}{\partial x_i} \tag{21.6}$$

$$\operatorname{div}\underset{\sim}{\boldsymbol{u}} = \frac{\partial u_k}{\partial x_k} \tag{21.7}$$

$$\operatorname{curl}\underset{\sim}{\boldsymbol{u}} = \underset{\sim}{\boldsymbol{e}}_i \varepsilon_{ijk} \frac{\partial u_k}{\partial x_j} \tag{21.8}$$

Most modern books go on to acknowledge the following common alternative notations (structures):

$$\nabla\phi \text{ means the same thing as } \operatorname{grad}\phi \tag{21.9}$$

$$\nabla\underset{\sim}{\boldsymbol{u}} \text{ means the same thing as } \operatorname{grad}\underset{\sim}{\boldsymbol{u}} \tag{21.10}$$

$$\nabla \bullet \underset{\sim}{\boldsymbol{u}} \text{ means the same thing as } \operatorname{div}\underset{\sim}{\boldsymbol{u}} \tag{21.11}$$

$$\nabla \times \underset{\sim}{\boldsymbol{u}} \text{ means the same thing as } \operatorname{curl}\underset{\sim}{\boldsymbol{u}} \tag{21.12}$$

Note that $\nabla\underset{\sim}{\boldsymbol{u}}$ is a second-order tensor (class V_3^2) whose ij component is given by $\partial u_j / \partial x_i$. Later on, we will discuss an alternative definition of the gradient for which the ij component is $\partial u_i / \partial x_j$, the only difference being the ordering of the indices.

Also note that

$$\nabla \bullet \underset{\sim}{\boldsymbol{u}} = \operatorname{tr}(\nabla\underset{\sim}{\boldsymbol{u}}) \tag{21.13}$$

$$\nabla \times \underset{\sim}{\boldsymbol{u}} = \underset{\approx}{\varepsilon} : (\nabla\underset{\sim}{\boldsymbol{u}}) \tag{21.14}$$

Consequently, if the tensor $\nabla\underset{\sim}{\boldsymbol{u}}$ is available, then the divergence and curl can be readily computed without having to compute any spatial derivatives again (an efficiency advantage for computational analysis).

Another common operation seen frequently in tensor calculus is the *Laplacian*, which is simply the divergence of the gradient and is commonly denoted by the operator ∇^2:

$$\nabla^2\phi \equiv \operatorname{div}(\operatorname{grad}\phi) \tag{21.15}$$

For rectangular Cartesian systems (RCS), this can be computed by

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x_k \partial x_k} \tag{21.16}$$

Gradients in curvilinear coordinates. The component formulas for computing gradients (or divergences, or Laplacians, or any other operation that involves differentiating with respect to the position vector) become more complicated in curvilinear (non-RCS) systems such as cylindrical or spherical coordinates. That's because the base vectors in curvilinear systems are themselves functions of position and, consequently, changing the position vector will change both the position vector's components *and* its basis. Component formulas for the gradient operations in other coordinate systems can be found in virtually any mechanics textbook (the tables in Ref. [4] are particularly well done).

Before pulling a formula from a table, however, you should look at the table's entries for RCS to ensure that the authors have defined the gradient in the same way that we have. If not, you'll have to do a quick translation. Many authors, for example, define $\nabla \underset{\sim}{\boldsymbol{u}}$ to have ij components given by $\partial u_i / \partial x_j$, which is the transpose of how we have defined the gradient in this section. If you have a table that uses this index ordering, all you have to do is swap around the indices in the table accordingly. For self-consistency of *our* tensor structures (i.e., our notation), we will always define the indicial forms in a manner analogous to similar looking vector operations. For example, if $\underset{\sim}{\nabla}$ *were* a vector (it's not), then $\underset{\sim}{\nabla}\underset{\sim}{\boldsymbol{u}}$ would be a dyad with components $\nabla_i u_j$. The ∇ is *not* a vector — it's an operator. Nonetheless, the arrangement of indices in *any* expression involving ∇ will be the same as an expression of the same form with ∇ the operator replaced by $\underset{\sim}{\nabla}$ the vector. In indicial expressions, every place that the vector version has ∇_i, the gradient operation will have $\frac{\partial}{\partial x_i}$, which acts on operands to its right. Again, keep in mind that these statements hold only for RCS coordinates, and must be generalized if curvilinear coordinates are used. Later on, we will discuss a left operating gradient as well.

The general theory of curvilinear coordinates is covered in Ref. [6] (or in numerous other standard textbooks on tensor analysis), where it is shown that gradient operations will always involve extra terms resulting from spatial variation of the base vectors. In general curvilinear theory, this effect is accounted for through the appearance of Christoffel symbols. There are two kinds of Christoffel symbols, the first represents the change in the base vectors with respect to changes in position coordinates and the other represents the gradient of the position coordinates themselves. Thankfully, the necessary mathematics for computing gradients in the most common curvilinear systems (e.g, cylindrical, spherical, elliptic, parabolic, etc.) have been worked out long ago, and practicing engineers only need to look up the appropriate formulas in tables. The modern approach is to work out general physical theories using RCS coordinates, making certain that every step in any derivation is using only proper tensor operations (addition, scalar multiplication, contractions, swaps, gradients, etc.). Once a final result is obtained, it should be cast in symbolic (structured) form so that, if desired, the formula for that same operation in a non-RCS system can be simply looked up in a table. If, for example, your RCS analysis resulted in the equation

$$\dot{\rho} + \rho \frac{\partial v_k}{\partial x_k} = 0 \qquad (21.17)$$

then you should not stop there! Your final result should be cast in structured form as

$$\dot{\rho} + \rho(\nabla \bullet \underset{\sim}{\boldsymbol{v}}) = 0 \qquad (21.18)$$

With this result, you can go to a table for the divergence in, say, cylindrical coordinates and immediately write down the version of the result that applies for that curvilinear system:

$$\dot{\rho} + \rho\left[\frac{\partial v_r}{\partial r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{\partial v_z}{\partial z}\right] = 0 \tag{21.19}$$

This formula is not too vastly different from the RCS formula, but you will find that differences become increasingly pronounced for multiple gradient operations or for gradients of higher order tensors. Incidentally, note that the above formula has the divisor of "*r*" in the middle term — without it, the equation would not be dimensionally correct.

This "look it up in a table" mentality should not be treated as a licence to be sloppy, nor should it be paralyzing if you don't happen to have a table with you at the time you need a curvilinear formula. You can always go back to basics. Suppose, for example, that you need the formula cited in Eq. (21.19), but you don't have a gradient look-up table handy to find out how to express $\nabla \bullet \underset{\sim}{\boldsymbol{v}}$ in cylindrical coordinates. To figure it out on your own, first write the vector $\underset{\sim}{\boldsymbol{v}}$ in terms of cylindrical coordinates:

$$\underset{\sim}{\boldsymbol{v}} = v_r \underset{\sim}{\boldsymbol{e}}_r + v_\theta \underset{\sim}{\boldsymbol{e}}_\theta + v_z \underset{\sim}{\boldsymbol{e}}_z \tag{21.20}$$

From here, you can use linearity of the divergence operation in combination with formulas for the divergence of a scalar (in this case each coefficient) times a vector (in this case, the spatially-varying base vectors). This process will be described soon. The point here is that you don't have to just give up when you don't have a gradient table, nor do you need to invest time in learning general curvilinear theory unless you are *routinely* needing numerous curvilinear gradient formulas for obscure curvilinear systems.

When do you NOT have to worry about curvilinear formulas? So far, we have defined the meanings of various gradient operations, all of which involved derivatives with respect to the position vector. We have explained that you need to look up special formulas for gradient operations when you use a curvilinear coordinate system. These special formulas reflect that the curvilinear basis *itself* changes when you change the position vector.

Many physical operations involve derivatives with respect to the position vector, but equally many involve derivatives with respect to *other* vectors as well. For example, the kinetic energy per unit mass of a material is given by

$$\kappa = \frac{1}{2}\underset{\sim}{\boldsymbol{v}} \bullet \underset{\sim}{\boldsymbol{v}} \tag{21.21}$$

You can compute the derivative of κ with respect to the velocity in much the same way as you would compute the spatial gradient. To indicate that the derivative is with respect to velocity $\underset{\sim}{\boldsymbol{v}}$, many people put a subscript "$\underset{\sim}{\boldsymbol{v}}$" on their nabla so that

$$\nabla_{\underset{\sim}{\boldsymbol{v}}}\kappa = \frac{\partial}{\partial v_i}\left[\frac{1}{2}v_k v_k\right]\underset{\sim}{\boldsymbol{e}}_i = \left[\frac{1}{2}(\delta_{ik}v_k + v_k\delta_{ik})\right]\underset{\sim}{\boldsymbol{e}}_i = v_i\underset{\sim}{\boldsymbol{e}}_i = \underset{\sim}{\boldsymbol{v}} \tag{21.22}$$

We can derive this in a more explicit manner by writing Eq. (21.21) as

$$\kappa = \frac{1}{2}(v_1^2 + v_2^2 + v_3^2) \tag{21.23}$$

The gradient with respect to $\underset{\sim}{\boldsymbol{v}}$ is given by

$$\nabla_{\underset{\sim}{\boldsymbol{v}}}\kappa = \frac{\partial\kappa}{\partial v_1}\underset{\sim}{\boldsymbol{e}}_1 + \frac{\partial\kappa}{\partial v_2}\underset{\sim}{\boldsymbol{e}}_2 + \frac{\partial\kappa}{\partial v_3}\underset{\sim}{\boldsymbol{e}}_3 \qquad (21.24)$$

or

$$\nabla_{\underset{\sim}{\boldsymbol{v}}}\kappa = v_1\underset{\sim}{\boldsymbol{e}}_1 + v_2\underset{\sim}{\boldsymbol{e}}_2 + v_3\underset{\sim}{\boldsymbol{e}}_3 \qquad (21.25)$$

Now suppose that a curvilinear basis is used. Do we need to revise this formula? No! Here we are differentiating with respect to velocity, not the position vector. The basis changes when the position vector changes, but the basis is *not* a function of the velocity vector. In terms of a cylindrical coordinate system, the velocity vector is

$$\underset{\sim}{\boldsymbol{v}} = v_r\underset{\sim}{\boldsymbol{e}}_r + v_\theta\underset{\sim}{\boldsymbol{e}}_\theta + v_z\underset{\sim}{\boldsymbol{e}}_z \qquad (21.26)$$

and therefore

$$\kappa = \frac{1}{2}(v_r^2 + v_\theta^2 + v_z^2) \qquad (21.27)$$

The derivative of κ with respect to $\underset{\sim}{\boldsymbol{v}}$ is given by

$$\nabla_{\underset{\sim}{\boldsymbol{v}}}\kappa = \frac{\partial\kappa}{\partial v_r}\underset{\sim}{\boldsymbol{e}}_r + \frac{\partial\kappa}{\partial v_\theta}\underset{\sim}{\boldsymbol{e}}_r + \frac{\partial\kappa}{\partial v_z}\underset{\sim}{\boldsymbol{e}}_z = v_r\underset{\sim}{\boldsymbol{e}}_r + v_\theta\underset{\sim}{\boldsymbol{e}}_\theta + v_z\underset{\sim}{\boldsymbol{e}}_z \qquad (21.28)$$

Contrast this with the *spatial* gradient of κ, which is given by

$$\nabla\kappa = \frac{\partial\kappa}{\partial r}\underset{\sim}{\boldsymbol{e}}_r + \frac{1}{r}\frac{\partial\kappa}{\partial\theta}\underset{\sim}{\boldsymbol{e}}_r + \frac{\partial\kappa}{\partial z}\underset{\sim}{\boldsymbol{e}}_z \qquad (21.29)$$

The *spatial* gradient has the divisor $1/r$ in the middle term, but the gradient with respect to *velocity* doesn't (note that putting it there would foul up the physical dimensions of that term). The derivative of κ with respect to $\underset{\sim}{\boldsymbol{v}}$ looks essentially the same as the Cartesian formula except that the indices 1,2, and 3 are replaced by r, θ, and z.

In most engineering problems, a curvilinear basis is used only for the position vector. Hence, only formulas involving *spatial* derivatives will need special forms. As another example, consider the fourth-order elastic stiffness tensor from elasticity:

$$E_{ijkl} = \frac{\partial\sigma_{ij}}{\partial\varepsilon_{kl}} \qquad (21.30)$$

This tensor does not involve any spatial derivatives. Consequently, it takes an identical form in a curvilinear system. For example,

$$E_{r\theta r\theta} = \frac{\partial\sigma_{r\theta}}{\partial\varepsilon_{r\theta}} \qquad (21.31)$$

Keep in mind that an unadorned ∇ is reserved for *spatial* gradients. In other words, ∇ means the same thing as $\nabla_{\underset{\sim}{\boldsymbol{x}}}$.

Spatial gradients of higher-order tensors. Let $\underset{\approx}{\boldsymbol{T}}$ be a spatially varying second-order tensor. Its gradient is a third-order tensor defined

$$\mathrm{grad}\underset{\approx}{\boldsymbol{T}} \equiv \lim_{V\to 0}\frac{1}{V}\int_{\partial\Omega} \underset{\sim}{\boldsymbol{n}}\underset{\approx}{\boldsymbol{T}}dS, \tag{21.32}$$

which is also denoted $\nabla\underset{\approx}{\boldsymbol{T}}$.

In terms of a Rectangular Cartesian System (RCS), the indicial formula for the tensor gradient is

$$(\nabla\underset{\approx}{\boldsymbol{T}})_{pqr} = \frac{\partial(T_{qr})}{\partial x_p} \tag{21.33}$$

In RCS basis notation,

$$\nabla\underset{\approx}{\boldsymbol{T}} = \frac{\partial(T_{ij})}{\partial x_n}\underset{\sim}{\boldsymbol{e}}_n\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \tag{21.34}$$

The ordering of the indices is important. For example,

$$(\nabla\underset{\approx}{\boldsymbol{T}})_{pqr} \neq (\nabla\underset{\approx}{\boldsymbol{T}})_{qpr} \tag{21.35}$$

For non-RCS systems, the component formulas become far more complicated and will involve Christoffel symbols (bleach!). Whenever possible, just go look up the result in a table for your system of interest. Yes, you should be *capable* of deriving it on your own, but the analysis is so complicated that the probability of a mistake far outweighs the likelihood you will happen to pick a reference book that has the formula wrong. It is a simple matter of risk-management.

The divergence of a second-order tensor (which plays a pivotal role in the continuum equations of motion) is a first-order tensor (i.e., a vector), defined

$$\mathrm{div}\underset{\approx}{\boldsymbol{T}} \equiv \lim_{V\to 0}\frac{1}{V}\int_{\partial\Omega} \underset{\sim}{\boldsymbol{n}} \bullet \underset{\approx}{\boldsymbol{T}}dS, \tag{21.36}$$

Also denoted by $\nabla \bullet \underset{\approx}{\boldsymbol{T}}$, this divergence operation is given in RCS component and basis notation as

$$(\nabla \bullet \underset{\approx}{\boldsymbol{T}})_i = \frac{\partial(T_{ki})}{\partial x_k} \tag{21.37}$$

$$\nabla \bullet \underset{\approx}{\boldsymbol{T}} = \frac{\partial(T_{ki})}{\partial x_k}\underset{\sim}{\boldsymbol{e}}_i \tag{21.38}$$

Cylindrical and spherical coordinate formulas for this operation can be found in virtually any book on continuum mechanics.

Note that

$$\nabla \bullet \underset{\approx}{\boldsymbol{T}} = C_1^2[\nabla\underset{\approx}{\boldsymbol{T}}] = \underset{\approx}{\boldsymbol{I}}\boldsymbol{:}[\nabla\underset{\approx}{\boldsymbol{T}}], \tag{21.39}$$

where C_1^2 denotes contraction of the first two base vectors [See page 241].

The curl of a second-order tensor (which plays a role in non-local and polar theories of material modeling) is a second-order tensor, defined

$$\text{curl}\,\underset{\approx}{\boldsymbol{T}} \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{n}} \times \underset{\approx}{\boldsymbol{T}}\, dS\,, \tag{21.40}$$

Also denoted by $\nabla \times \underset{\approx}{\boldsymbol{T}}$, this operation is given in RCS indicial and basis notation as

$$(\nabla \times \underset{\approx}{\boldsymbol{T}})_{pq} = \varepsilon_{pmn} \frac{\partial (T_{nq})}{\partial x_m} \tag{21.41}$$

$$\nabla \times \underset{\approx}{\boldsymbol{T}} = \varepsilon_{imn} \frac{\partial (T_{nj})}{\partial x_m} \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \tag{21.42}$$

Note that

$$\nabla \times \underset{\approx}{\boldsymbol{T}} = \underset{\sim}{\underset{\approx}{\varepsilon}} \boldsymbol{:} [\nabla \underset{\approx}{\boldsymbol{T}}]\,, \tag{21.43}$$

Note that both the divergence and the curl can be obtained from the gradient $\nabla \underset{\approx}{\boldsymbol{T}}$ by applying a second-order inner product (i.e., a double dot, or double contraction) between a constant isotropic tensor (either the identity or the permutation tensor) and $\nabla \underset{\approx}{\boldsymbol{T}}$. This observation can prove useful when (later) evaluating integrals; specifically, both the constant isotropic tensor and the contraction operation (which is a linear operation) can be pulled "outside" the integral in many instances.

Spatial gradient operations for higher order tensors are defined similarly (consistent structure).

Product rule for gradient operations. For scalars, the product rule of differentiation says

$$\frac{d(f(x)g(x))}{dx} = f(x)g'(x) + g(x)f'(x)\,, \tag{21.44}$$

where the prime denotes differentiation. Similar rules exist for gradients (or divergences or curls) of products of higher-order tensors, but the issue of basis ordering becomes paramount. When seeking to derive such a formula, your best strategy is to apply the product rule to the RCS components, and then reconstruct the direct notation (structured) formula once you're done.

The direct notation formulas for scalar and vector arguments are straightforward and relatively intuitive. However, gradient formulas for second and higher-order tensors must frequently use of the exchange (swap) operator X_i^j discussed on page 242 if they are to be expressed in direct (structured) notation. The key concept is

Dyadic ordering must be preserved in gradient operations (21.45)

For example, $\nabla(\underset{\sim}{u}\underset{\sim}{v})$ is a special case of the gradient of a second-order tensor (because a dyad is a tensor). The result must be a third order tensor, expressed in RCS basis notation as

$$\nabla(\underset{\sim}{u}\underset{\sim}{v}) = \frac{\partial(u_i v_j)}{\partial x_n}\underset{\sim}{e}_n\underset{\sim}{e}_i\underset{\sim}{e}_j \tag{21.46}$$

Imagine for a moment that the ∇ is a vector. Then self-consistency of this book's tensor structures says that it must be multiplied dyadically with $\underset{\sim}{u}\underset{\sim}{v}$. The del ∇ comes first, then $\underset{\sim}{u}$, then $\underset{\sim}{v}$. Expanding $\nabla(\underset{\sim}{u}\underset{\sim}{v})$ with a product rule must result in an expression that preserves this exact same ordering. The base vector that goes along with ∇ has to come first. The base vector that goes with $\underset{\sim}{u}$ comes second and the one with $\underset{\sim}{v}$ comes last. Using the product rule, the derivative may be written in RCS basis notation as

$$\nabla(\underset{\sim}{u}\underset{\sim}{v}) = \left(\frac{\partial u_i}{\partial x_n}v_j\right)\underset{\sim}{e}_n\underset{\sim}{e}_i\underset{\sim}{e}_j + \left(u_i\frac{\partial v_j}{\partial x_n}\right)\underset{\sim}{e}_n\underset{\sim}{e}_i\underset{\sim}{e}_j \tag{21.47}$$

Consider the first term on the right-hand side of this equation. Equation (7.60) permits us to move over the v_j factor so that it is adjacent to the $\underset{\sim}{e}_j$ base vector. Doing this allows us to write that first term in direct (structured) notation:

$$\left(\frac{\partial u_i}{\partial x_n}v_j\right)\underset{\sim}{e}_n\underset{\sim}{e}_i\underset{\sim}{e}_j = \left[\left(\frac{\partial u_i}{\partial x_n}\right)\underset{\sim}{e}_n\underset{\sim}{e}_i\right]\{v_j\underset{\sim}{e}_j\} = [\nabla\underset{\sim}{u}]\underset{\sim}{v}, \tag{21.48}$$

The last term right-hand side of Eq. (21.47) is not so easy. Equation (7.60) tells us that scalar multiplication of dyads is a commutative operation — the scalar may "move around" freely, attaching itself to any of the vectors that form the dyad or triad. Unfortunately, however, dyadic multiplication itself is not commutative (it's order dependent). While it is permissible to move the u_i over to be adjacent to $\underset{\sim}{e}_i$, it is not permissible to move the $\underset{\sim}{e}_i$ over to become adjacent to u_i. Doing that would change the basis triad ordering from $\underset{\sim}{e}_n\underset{\sim}{e}_i\underset{\sim}{e}_j$ to $\underset{\sim}{e}_i\underset{\sim}{e}_n\underset{\sim}{e}_j$, which is not the same triad, so the move is not permissible. There is no way to express the last term right-hand side of Eq. (21.47) in structured notation *without using a swap operation.* Intuitively, you might anticipate the second term to come out to be $\underset{\sim}{u}(\nabla\underset{\sim}{v})$, however this is wrong — we've shown that this result would entail invalid rearrangement of the basis triad. Stated differently, it fails to preserve the required "∇-then-$\underset{\sim}{u}$-then-$\underset{\sim}{v}$" ordering. However, you can apply a swap operation, X_1^2 to the expression $\underset{\sim}{u}(\nabla\underset{\sim}{v})$ to put the basis ordering back to what it's supposed to be. Therefore,

$$\left(u_i\frac{\partial v_j}{\partial x_n}\right)\underset{\sim}{e}_n\underset{\sim}{e}_i\underset{\sim}{e}_j = X_1^2[\underset{\sim}{u}(\nabla\underset{\sim}{v})], \tag{21.49}$$

and we have our product rule identity in structured notation:

$$\nabla(\underset{\sim}{\mathbf{u}}\underset{\sim}{\mathbf{v}}) = (\nabla\underset{\sim}{\mathbf{u}})\underset{\sim}{\mathbf{v}} + X_1^2[\underset{\sim}{\mathbf{u}}(\nabla\underset{\sim}{\mathbf{v}})] \tag{21.50}$$

The rule about preserving dyadic ordering applies only to the dyadically multiplied vectors (or vector-like quantities such as the "nabla" or "del" ∇) — not to the ones that are "annihilated" by a dot product or other contraction (but you do need to make sure that the same vectors (or vector-like quantities) are annihilating each other in each term. For example, in our identity list below, we cite the formula

$$\nabla \bullet (\phi\underset{\sim}{\mathbf{v}}) = \underset{\sim}{\mathbf{v}} \bullet (\nabla\phi) + \phi(\nabla \bullet \underset{\sim}{\mathbf{v}}) \tag{21.51}$$

At first glance, the operand ordering might appear to be "∇-then-ϕ-then-$\underset{\sim}{\mathbf{v}}$". However, ϕ is a scalar, so it is not required to be part of this ordering. Furthermore, even though the dyadic product of two vectors is order dependent ($\underset{\sim}{\mathbf{a}}\underset{\sim}{\mathbf{b}} \neq \underset{\sim}{\mathbf{b}}\underset{\sim}{\mathbf{a}}$), the *dot* product is *not* order dependent $\underset{\sim}{\mathbf{a}} \bullet \underset{\sim}{\mathbf{b}} = \underset{\sim}{\mathbf{b}} \bullet \underset{\sim}{\mathbf{a}}$. Are the same "vectors" annihilating each other in every term? Yes. On the left hand side, the nabla ∇ "annihilates" the vector $\underset{\sim}{\mathbf{v}}$ (so their ordering doesn't matter, but they must still annihilate each other in each term of the identity). On the first term of the right hand side, the vector $\underset{\sim}{\mathbf{v}}$ is still annihilating the nabla, and it is doing the same on the last term. Order dependence is not required on the del or the $\underset{\sim}{\mathbf{v}}$ in Eq. (21.51) because, after the dot product, there exist no base vectors to keep ordered — the result is a scalar. Only dyadically multiplied vectors (or vector-like quantities) must preserve order. After a while, these subtleties will become second nature to you. Until then, your best bet is to work out identities like this using RCS indicial notation, and then cast the final result back to symbolic structured notation (you need to cast back to structured notation if you wish to apply the identity in curvilinear coordinates, where the RCS formulas don't apply; the fact that you temporarily used RCS in your derivations is inconsequential — once you have the structured formula, you can write it down in any system you prefer by using appropriate tabulated formulas for each operation).

Identities involving the "nabla".

Identities that involve the "nabla" ∇ gradient operator are listed below. For clarity, the equivalent identity is also expressed indicially for a rectangular Cartesian system (RCS). Though easier to interpret, the indicial expressions are valid only for RCS. Different formulas must be used for curvilinear systems [see page 252]. We highly recommend that you do *not* try to memorize this list or carry it around with you. Chances are that whatever identity you need for a particular application won't be listed here. Instead, you should use this list as a "practice-ground" for you to learn how to prove the formulas yourself.

$$\nabla(\phi\psi) = \phi(\nabla\psi) + \psi(\nabla\phi) \qquad \frac{\partial(\phi\psi)}{\partial x_n} = \psi\left(\frac{\partial\phi}{\partial x_n}\right) + \phi\left(\frac{\partial\psi}{\partial x_n}\right) \tag{21.52}$$

$$\nabla(\underset{\sim}{\mathbf{u}} \bullet \underset{\sim}{\mathbf{v}}) = (\nabla\underset{\sim}{\mathbf{u}}) \bullet \underset{\sim}{\mathbf{v}} + (\nabla\underset{\sim}{\mathbf{v}}) \bullet \underset{\sim}{\mathbf{u}} \qquad \frac{\partial(u_k v_k)}{\partial x_i} = \left(\frac{\partial u_k}{\partial x_i}\right)v_k + \left(\frac{\partial v_k}{\partial x_i}\right)u_k$$

also see Eq. (21.70)

(21.53)

$$\nabla(\phi+\psi) = \nabla\phi+\nabla\psi \qquad \frac{\partial(\phi+\psi)}{\partial x_n} = \frac{\partial\phi}{\partial x_n}+\frac{\partial\psi}{\partial x_n} \tag{21.54}$$

$$\nabla(\underset{\sim}{\boldsymbol{u}}+\underset{\sim}{\boldsymbol{v}}) = \nabla\underset{\sim}{\boldsymbol{u}}+\nabla\underset{\sim}{\boldsymbol{v}} \qquad \frac{\partial(u_i+v_i)}{\partial x_n} = \frac{\partial u_i}{\partial x_n}+\frac{\partial v_i}{\partial x_n} \tag{21.55}$$

$$\nabla(\phi\underset{\sim}{\boldsymbol{u}}) = (\nabla\phi)\underset{\sim}{\boldsymbol{u}}+\phi(\nabla\underset{\sim}{\boldsymbol{u}}) \qquad \frac{\partial(\phi u_i)}{\partial x_n} = \left(\frac{\partial\phi}{\partial x_n}\right)u_i+\phi\left(\frac{\partial u_i}{\partial x_n}\right) \tag{21.56}$$

$$\nabla(\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}}) = (\nabla\underset{\sim}{\boldsymbol{u}})\underset{\sim}{\boldsymbol{v}}+X_1^2[\underset{\sim}{\boldsymbol{u}}(\nabla\underset{\sim}{\boldsymbol{v}})] \qquad \frac{\partial(u_iv_j)}{\partial x_n} = \frac{\partial u_i}{\partial x_n}v_j+u_i\frac{\partial v_j}{\partial x_n} \tag{21.57}$$

$$\nabla\bullet(\nabla\times\underset{\sim}{\boldsymbol{v}}) = 0 \quad \textit{\textbf{(see below)}} \qquad \frac{\partial}{\partial x_i}\varepsilon_{ijk}\frac{\partial u_k}{\partial x_j} = \varepsilon_{ijk}\frac{\partial^2 u_k}{\partial x_i\partial x_j} = 0 \tag{21.58}$$

$$\nabla\times(\nabla\phi) = \underset{\sim}{\boldsymbol{0}} \quad \textit{\textbf{(see below)}} \qquad \varepsilon_{ijk}\frac{\partial(\partial\phi/\partial x_k)}{\partial x_j} = \varepsilon_{ijk}\frac{\partial^2\phi}{\partial x_j\partial x_k} = 0 \tag{21.59}$$

$$\nabla\bullet(\phi\underset{\sim}{\boldsymbol{v}}) = \underset{\sim}{\boldsymbol{v}}\bullet\nabla\phi+\phi\nabla\bullet\underset{\sim}{\boldsymbol{v}} \qquad \frac{\partial(\phi v_i)}{\partial x_i} = \left(\frac{\partial\phi}{\partial x_i}\right)v_i+\phi\left(\frac{\partial v_i}{\partial x_i}\right) \tag{21.60}$$

$$\nabla\times(\phi\underset{\sim}{\boldsymbol{v}}) = \phi\nabla\times\underset{\sim}{\boldsymbol{v}}+(\nabla\phi)\times\underset{\sim}{\boldsymbol{v}} \tag{21.61}$$

$$\nabla\bullet(\underset{\sim}{\boldsymbol{u}}\times\underset{\sim}{\boldsymbol{v}}) = \underset{\sim}{\boldsymbol{v}}\bullet(\nabla\times\underset{\sim}{\boldsymbol{u}})-\underset{\sim}{\boldsymbol{u}}\bullet(\nabla\times\underset{\sim}{\boldsymbol{v}}) \tag{21.62}$$

$$\nabla\times(\underset{\sim}{\boldsymbol{u}}\times\underset{\sim}{\boldsymbol{v}}) = \underset{\sim}{\boldsymbol{u}}(\nabla\bullet\underset{\sim}{v})-\underset{\sim}{\boldsymbol{u}}\bullet(\nabla\underset{\sim}{\boldsymbol{v}})+\underset{\sim}{v}\bullet(\nabla\underset{\sim}{\boldsymbol{u}})-\underset{\sim}{\boldsymbol{v}}(\nabla\bullet\underset{\sim}{\boldsymbol{u}}) \tag{21.63}$$

$$\nabla\bullet(\nabla\phi\times\nabla\psi) = 0 \quad \textit{\textbf{(see below)}} \tag{21.64}$$

$$\nabla\times(\nabla\times\underset{\sim}{\boldsymbol{v}}) = \nabla(\nabla\bullet\underset{\sim}{\boldsymbol{v}})-\nabla^2\underset{\sim}{\boldsymbol{v}} \tag{21.65}$$

$$\underset{\sim}{\boldsymbol{v}}\times(\nabla\times\underset{\sim}{\boldsymbol{u}}) = 2[\operatorname{skw}(\nabla\underset{\sim}{\boldsymbol{u}})]\bullet\underset{\sim}{\boldsymbol{v}} \tag{21.66}$$

The identity (21.58) follows from noting that, for any sufficiently smooth function $f(x, y)$,

$$\frac{\partial f}{\partial x\partial y} = \frac{\partial f}{\partial y\partial x} \tag{21.67}$$

When applied to a double gradient, this calculus identity may be written

$$\frac{\partial(\quad)}{\partial x_i\partial x_j} = \frac{\partial(\quad)}{\partial x_j\partial x_i} \tag{21.68}$$

In other words, there is symmetry with respect to the indices i and j. Consequently, the second partial derivative in Eq. (21.58) is symmetric with respect to the i and j indices. When contracted with ε_{ijk}, which is *skew-symmetric* in its i and j indices, the result must be zero. Equations (21.59) and (21.64) are zero for similar reasons.

Compound differential operator notation (and unfortunate pampering). You will frequently see authors use a compound gradient operator such as "$(\underset{\sim}{u} \bullet \nabla)$", the meaning of which depends on the operand. For example, the notation $(\underset{\sim}{u} \bullet \nabla)\phi$ means the same thing as $\underset{\sim}{u} \bullet (\nabla\phi)$ and the notation $(\underset{\sim}{u} \bullet \nabla)\underset{\sim}{v}$ means the same thing as $\underset{\sim}{u} \bullet (\nabla\underset{\sim}{v})$. Note that this equivalence is similar to saying that $(\underset{\sim}{a} \bullet \underset{\sim}{b})\underset{\sim}{c}$ means the same thing as $\underset{\sim}{a} \bullet (\underset{\sim}{b}\underset{\sim}{c})$.

As a general rule, if you are ever faced with "deciphering" a compound gradient operator (i.e., one that involves a gradient and other things), then the indicial form for the operation should be similar to the indicial forms of algebraic equations where the nabla/del is replaced with an ordinary vector. Just keep in mind that the nabla will be operating on *everything* to its right in a given term (or up through a closed parenthesis if the operator is intended to act on multiple terms).

Upon occasion, you will need to work with another person who doesn't have a background in tensor analysis. Such a person will not be able to decipher an expression like $(\nabla\underset{\sim}{u}) \bullet \underset{\sim}{v}$ because this expression involves a tensor $\nabla\underset{\sim}{u}$ which is foreign to them. To help these people out, you can replace $(\nabla\underset{\sim}{u}) \bullet \underset{\sim}{v}$ by something more familiar to non-tensor-savvy researchers through the use of the following identity:

$$(\nabla\underset{\sim}{u}) \bullet \underset{\sim}{v} = (\underset{\sim}{v} \bullet \nabla)\underset{\sim}{u} + \underset{\sim}{v} \times (\nabla \times \underset{\sim}{u}) \quad \textit{\textbf{(proof below)}} \qquad (21.69)$$

Using this identity, for example, you can write the identity of Eq. (21.53) as

$$\nabla(\underset{\sim}{u} \bullet \underset{\sim}{v}) = (\underset{\sim}{v} \bullet \nabla)\underset{\sim}{u} + (\underset{\sim}{u} \bullet \nabla)\underset{\sim}{v} + \underset{\sim}{v} \times (\nabla \times \underset{\sim}{u}) + \underset{\sim}{u} \times (\nabla \times \underset{\sim}{v}) \qquad (21.70)$$

which is what you would typically find in a standard handbook such as the CRC [3]. There is absolutely no reason to use this horrendous identity unless you are dealing with a "tensor-challenged" audience. The identity of Eq. (21.53) is far more elegant and less prone to computational error.

Recall that the compound operator $(\underset{\sim}{v} \bullet \nabla)$ in an expression of the form $(\underset{\sim}{v} \bullet \nabla)\underset{\sim}{u}$ is merely an alternative notation for $\underset{\sim}{v} \bullet (\nabla\underset{\sim}{u})$. The compound operator is introduced to make the presence of a tensor transparent to people who aren't familiar with them. Note that $\underset{\sim}{v} \bullet (\nabla\underset{\sim}{u}) \neq (\nabla\underset{\sim}{u}) \bullet \underset{\sim}{v}$. Instead, $\underset{\sim}{v} \bullet (\nabla\underset{\sim}{u}) = (\nabla\underset{\sim}{u})^T \bullet \underset{\sim}{v}$. A simple — but of course crucial — difference of arrangement of indices that distinguishes these expressions.

To prove Eq. (21.69), consider the expression $\underset{\sim}{v} \times (\nabla \times \underset{\sim}{u})$. In indicial form, this is

$$\{\underset{\sim}{v} \times (\nabla \times \underset{\sim}{u})\}_i = \varepsilon_{imn} v_m \left(\varepsilon_{npq} \frac{\partial}{\partial x_p} u_q \right) \qquad (21.71)$$

This expression has two permutation symbols, and the index "*n*" appears on both of them, so you can use the e-delta identity to write

$$\{\underset{\sim}{v} \times (\nabla \times \underset{\sim}{u})\}_i = (\delta_{ip}\delta_{mq} - \delta_{iq}\delta_{mp}) v_m \frac{\partial u_q}{\partial x_p} \qquad (21.72)$$

Multiplying this out gives

$$\{\underset{\sim}{v} \times (\nabla \times \underset{\sim}{u})\}_i = v_q \frac{\partial u_q}{\partial x_i} - v_p \frac{\partial u_i}{\partial x_p} \qquad (21.73)$$

In structured notation, this result may be written

$$\underset{\sim}{\boldsymbol{v}} \times (\nabla \times \underset{\sim}{\boldsymbol{u}}) = (\nabla \underset{\sim}{\boldsymbol{u}}) \bullet \underset{\sim}{\boldsymbol{v}} - \underset{\sim}{\boldsymbol{v}} \bullet (\nabla \underset{\sim}{\boldsymbol{u}}) \qquad (21.74)$$

Solving for $(\nabla \underset{\sim}{\boldsymbol{u}}) \bullet \underset{\sim}{\boldsymbol{v}}$ gives the result of Eq. (21.69). This result, by the way, is also the proof of Eq. (21.66).

Right and left gradient operations (we love them both!)

You can define the gradient, divergence, and Laplacian formulas differently from the way they were defined in the previous section. Specifically, you can define

$$\mathrm{GRAD}\phi \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \phi \underset{\sim}{\boldsymbol{n}} dS \qquad (21.75)$$

$$\mathrm{GRAD}\underset{\sim}{\boldsymbol{u}} \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{n}} dS \qquad (21.76)$$

$$\mathrm{DIV}\underset{\sim}{\boldsymbol{u}} \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{u}} \bullet \underset{\sim}{\boldsymbol{n}} dS \qquad (21.77)$$

$$\mathrm{CURL}\underset{\sim}{\boldsymbol{u}} \equiv \lim_{V \to 0} \frac{1}{V} \int_{\partial\Omega} \underset{\sim}{\boldsymbol{u}} \times \underset{\sim}{\boldsymbol{n}} dS \qquad (21.78)$$

Note that the only difference between these formulas and the ones in the previous section is the placement of the unit normal $\underset{\sim}{\boldsymbol{n}}$. In terms of rectangular Cartesian coordinate systems (RCS), these operations can be computed by

$$\mathrm{GRAD}\phi = \underset{\sim}{\boldsymbol{e}}_k \frac{\partial \phi}{\partial x_k} \qquad (21.79)$$

$$\mathrm{GRAD}\underset{\sim}{\boldsymbol{u}} = \frac{\partial u_i}{\partial x_j} \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \qquad (21.80)$$

$$\mathrm{DIV}\underset{\sim}{\boldsymbol{u}} = \frac{\partial u_k}{\partial x_k} \qquad (21.81)$$

$$\mathrm{CURL}\underset{\sim}{\boldsymbol{u}} = \varepsilon_{ijk} \frac{\partial u_j}{\partial x_k} \underset{\sim}{\boldsymbol{e}}_i \qquad (21.82)$$

Writing these operations using a "del" ∇ operation requires the introduction of a new del symbol, $\overleftarrow{\nabla}$. This is a backward operating gradient defined such that

$\phi \overleftarrow{\nabla}$ means the same thing as $\mathrm{GRAD}\phi$ (21.83)

$\underset{\sim}{\boldsymbol{u}} \overleftarrow{\nabla}$ means the same thing as $\mathrm{GRAD}\underset{\sim}{\boldsymbol{u}}$ (21.84)

$\underset{\sim}{\boldsymbol{u}} \bullet \overleftarrow{\nabla}$ means the same thing as $\mathrm{DIV}\underset{\sim}{\boldsymbol{u}}$ (21.85)

$$\underset{\sim}{\boldsymbol{u}} \times \overleftarrow{\nabla} \text{ means the same thing as } \mathrm{CURL}\underset{\sim}{\boldsymbol{u}} \tag{21.86}$$

To avoid any confusion when working with both types of gradient operators, the "del" symbol used in the previous section can be alternatively written as a forward operating del as $\overrightarrow{\nabla}$.

Note that

$$\phi\overleftarrow{\nabla} = \overrightarrow{\nabla}\phi \tag{21.87}$$

$$\underset{\sim}{\boldsymbol{u}}\overleftarrow{\nabla} = (\overrightarrow{\nabla}\underset{\sim}{\boldsymbol{u}})^T \tag{21.88}$$

$$\underset{\sim}{\boldsymbol{u}} \bullet \overleftarrow{\nabla} = \overrightarrow{\nabla} \bullet \underset{\sim}{\boldsymbol{u}} \tag{21.89}$$

$$\underset{\sim}{\boldsymbol{u}} \times \overleftarrow{\nabla} = -(\overrightarrow{\nabla} \times \underset{\sim}{\boldsymbol{u}}) \tag{21.90}$$

The last formula is similar to the ordinary vector identity, $\underset{\sim}{\boldsymbol{u}} \times \underset{\sim}{\boldsymbol{v}} = -\underset{\sim}{\boldsymbol{v}} \times \underset{\sim}{\boldsymbol{u}}$. Note that $\overrightarrow{\nabla}\underset{\sim}{\boldsymbol{u}}$ is a second-order tensor (of class V_3^2). Consequently,

$$\underset{\sim}{\boldsymbol{u}} \bullet \overleftarrow{\nabla} = \mathrm{tr}(\underset{\sim}{\boldsymbol{u}}\overleftarrow{\nabla}) \tag{21.91}$$

$$\underset{\sim}{\boldsymbol{u}} \times \overleftarrow{\nabla} = \underset{\approx}{\varepsilon}\boldsymbol{:}(\underset{\sim}{\boldsymbol{u}}\overleftarrow{\nabla}) \tag{21.92}$$

The *Laplacian*, which is simply the DIVergence of the GRADient and is commonly denoted by the operator $\overleftarrow{\nabla}^2$:

$$\phi\overleftarrow{\nabla}^2 \equiv \mathrm{DIV}(\mathrm{GRAD}\phi) = \overrightarrow{\nabla}^2\phi \tag{21.93}$$

Sadly, the backward operating gradient is rarely seen in publications. We won't deny that it is certainly difficult to typeset. Nonetheless, it is very useful to recognize that these alternative definitions of differentiation exist. For both the forward and backward operating dels, note that the index on the $\partial(\quad)/\partial x_k$ is the same as the index that would be used if the del were just a vector. For example, if $\underset{\sim}{\boldsymbol{d}}$ is a vector, then dyad $\underset{\sim}{\boldsymbol{d}}\underset{\sim}{\boldsymbol{u}}$ would expand as $d_i u_j \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j$, which is similar in structure to

$$\overrightarrow{\nabla}\underset{\sim}{\boldsymbol{u}} = \frac{\partial}{\partial x_i} u_j \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \tag{21.94}$$

Conversely, the dyad $\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{d}}$ would expand as $u_i d_j \underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j$, which is similar in structure to

$$\underset{\sim}{\boldsymbol{u}}\overleftarrow{\nabla} = (\partial u_j/\partial x_i)\underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j \tag{21.95}$$

Despite this heuristic notation, keep in mind that the del is *not* a vector. The issue of right and left operating dels comes up implicitly in the mechanics literature, where it is not uncommon to see authors (e.g., Ref. [18]) to define $\nabla\underset{\sim}{\boldsymbol{u}}$ to mean $\frac{\partial u_j}{\partial x_i}\underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j$ in their nomenclature section, but later state that the velocity gradient tensor is $\nabla\underset{\sim}{\boldsymbol{v}} = \frac{\partial v_i}{\partial x_j}\underset{\sim}{\boldsymbol{e}}_i \underset{\sim}{\boldsymbol{e}}_j$, which is actually a *backward* gradient. This is an infuriatingly sloppy structure consistency violation. It undermines the goal of having heuristically self-explanatory notation.

Other authors [e.g. 24] define the gradient GRAD$\underset{\sim}{\boldsymbol{u}}$ with respect to $\underset{\sim}{\boldsymbol{x}}$ by being more explicit in the presumption that $\underset{\sim}{\boldsymbol{u}}$ is expressible as a function of $\underset{\sim}{\boldsymbol{x}}$ so that there is a function g such that $\underset{\sim}{\boldsymbol{u}} = g(\underset{\sim}{\boldsymbol{x}})$. Then GRAD$\underset{\sim}{\boldsymbol{u}}$ is defined to be the linear transformation (if it exists) for which

$$g(\underset{\sim}{\boldsymbol{x}}+\underset{\sim}{\boldsymbol{s}}) = g(\underset{\sim}{\boldsymbol{x}}) + (\mathrm{GRAD}\underset{\sim}{\boldsymbol{u}}) \bullet \underset{\sim}{\boldsymbol{s}} + o(s^2) \tag{21.96}$$

Here, the symbol "o" represents "order" and $o(s^2)$ represents a function $f(\underset{\sim}{\boldsymbol{s}})$ for which

$$\lim_{\underset{\sim}{\boldsymbol{s}}\to\underset{\sim}{0}} \frac{||f(\underset{\sim}{\boldsymbol{s}})||}{||\underset{\sim}{\boldsymbol{s}}||} = 0 \tag{21.97}$$

Note that Eq. (21.96) looks a lot like a first-order Taylor series expansion. When this approach is used to define the gradient, then Eq. (21.2) comes out as a *result*. If you think about it, this definition is pretty close to the definition of ordinary differentiation that you might see in an undergraduate calculus book. If you think of $\underset{\sim}{\boldsymbol{s}}$ as being an infinitesimal increment $d\underset{\sim}{\boldsymbol{x}}$, then $g(\underset{\sim}{\boldsymbol{x}}+\underset{\sim}{\boldsymbol{s}})-g(\underset{\sim}{\boldsymbol{x}})$ can be regarded as an increment $d\underset{\sim}{\boldsymbol{u}}$ and therefore Eq. (21.96) can be written somewhat less cryptically as

$$d\underset{\sim}{\boldsymbol{u}} = (\mathrm{GRAD}\underset{\sim}{\boldsymbol{u}}) \bullet d\underset{\sim}{\boldsymbol{x}} \tag{21.98}$$

Using our nabla notation, this would be written

$$d\underset{\sim}{\boldsymbol{u}} = \underset{\sim}{\boldsymbol{u}}\overleftarrow{\nabla} \bullet d\underset{\sim}{\boldsymbol{x}} \tag{21.99}$$

The other derivative operations can be defined similarly using this approach.

Unfortunately, many authors [e.g., 24] define the gradient as shown in Eq. (21.98) — which is a backward or "right" gradient — but they still use a nabla on the left so that they write Eq. (21.99) as $d\underset{\sim}{\boldsymbol{u}} = (\nabla\underset{\sim}{\boldsymbol{u}}) \bullet d\underset{\sim}{\boldsymbol{x}}$. We find this horribly confusing. With our distinction between left and right gradients, the forward (left) gradient would be defined in analog to Eq. (21.98) as

$$d\underset{\sim}{\boldsymbol{u}} = d\underset{\sim}{\boldsymbol{x}} \bullet (\mathrm{grad}\underset{\sim}{\boldsymbol{u}}) \tag{21.100}$$

which would be written using the nabla as

$$d\underset{\sim}{\boldsymbol{u}} = d\underset{\sim}{\boldsymbol{x}} \bullet \overrightarrow{\nabla}\underset{\sim}{\boldsymbol{u}} \tag{21.101}$$

Note that, regardless of whether a forward or backward nabla is used, the spatial increment $d\underset{\sim}{\boldsymbol{x}}$ is always positioned so that it dots directly into the nabla. Again, we contend that *both* backward and forward gradients are useful, but we feel adamant that the notation should carefully distinguish between them. Incidentally, as you move along to gradients of higher-order tensors, even more types of gradients can be defined according to the index placements. For example, here are three possible definitions of the gradient of a second-order tensor:

$$(\mathrm{forward\_grad}\underset{\approx}{\boldsymbol{U}})_{ijk} = \frac{\partial U_{jk}}{\partial x_i} \tag{21.102}$$

$$(\mathrm{middle\_grad}\underset{\approx}{\boldsymbol{U}})_{ijk} = \frac{\partial U_{ik}}{\partial x_j} \tag{21.103}$$

$$(\text{backward\_grad}\underset{\approx}{\boldsymbol{U}})_{ijk} = \frac{\partial U_{ij}}{\partial x_k} \tag{21.104}$$

We would be hard-pressed to come up with a simple direct notation for the middle gradient, and the situation would only get worse for higher-order tensors. Note, however, that the only distinction between these definitions is the index ordering. We have defined the exchange operator as a mechanism for re-ordering indices. Hence, you can always compute a gradient using any index placement you prefer, and then easily obtain the other gradients by exchange operations. Specifically,

$$\underset{\approx}{\boldsymbol{U}}\overleftarrow{\nabla} = \text{backward\_grad}\underset{\approx}{\boldsymbol{U}} = \frac{\partial U_{ij}}{\partial x_k}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\underset{\sim}{\boldsymbol{e}}_k \tag{21.105}$$

$$(\text{middle\_grad}\underset{\approx}{\boldsymbol{U}})_{ijk} = X_2^3(\underset{\approx}{\boldsymbol{U}}\overleftarrow{\nabla}) \tag{21.106}$$

$$(\text{forward\_grad}\underset{\approx}{\boldsymbol{U}})_{ijk} = X_1^2X_2^3(\underset{\approx}{\boldsymbol{U}}\overleftarrow{\nabla}) \tag{21.107}$$

What a mess! Our point here is that you can get to any other gradient definition if you have computed it using another definition — all that's needed is exchange operations. This fact can be very convenient in product rule situations.

Casual (non-rigorous) tensor calculus

The gradient operations that we have defined so far don't appeal to the intuition of people who are new to this subject. Furthermore, definitions like these immediately trap a person into a mindset that tensor calculus always has something to do with how fields (scalar, vector, or tensor) vary in *space* where the independent variable is the position vector $\underset{\sim}{\boldsymbol{x}}$. However, whenever you have a quantity (tensor or scalar) whose value can be expressed as a function of some other quantity (tensor or scalar), then straightforward extensions of scalar calculus exist to permit you to quantify how the first quantity changes in response to a change in the second quantity. For example, in elastic materials modeling, the stress tensor often depends in some measurable way on the strain tensor, and the derivative of the stress tensor with respect to the strain tensor turns out to be a fourth-order tensor — called the elastic tangent stiffness — that is completely analogous to Young's modulus E in the uniaxial stress equation $\sigma = E\varepsilon$.

To approach tensor calculus in a more casual way, we will define our calculus notation through a series of examples with the understanding that you should force yourself to plod through more the formal expositions on this subject available in any good tensor analysis textbook.

Whenever we write

$$\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}}, \tag{21.108}$$

we are implicitly telling you that the vector $\underset{\sim}{\boldsymbol{y}}$ can be expressed as a function of a *single* vector $\underset{\sim}{\boldsymbol{x}}$. Consequently each y_i component must be a function of the *three* $\{x_1, x_2, x_3\}$ components. By the chain rule,*

$$d\underset{\sim}{\boldsymbol{y}} = dy_i\underset{\sim}{\boldsymbol{e}}_i = \frac{\partial y_i}{\partial x_j}dx_j\underset{\sim}{\boldsymbol{e}}_i \qquad (21.109)$$

Noting that $dx_j = d\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{e}}_j$, this may be written

$$d\underset{\sim}{\boldsymbol{y}} = \frac{\partial y_i}{\partial x_j}(d\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{e}}_j)\underset{\sim}{\boldsymbol{e}}_i = \left(\frac{\partial y_i}{\partial x_j}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\right) \bullet d\underset{\sim}{\boldsymbol{x}} \qquad (21.110)$$

Even if $\underset{\sim}{\boldsymbol{y}}$ is a nonlinear function of $\underset{\sim}{\boldsymbol{x}}$, note that the infinitesimal *increment* $d\underset{\sim}{\boldsymbol{y}}$ is linear with respect to $d\underset{\sim}{\boldsymbol{x}}$. Thus, recalling Eq. (9.7), we have now demonstrated that there exists a second-order tensor, which we will denote $d\underset{\sim}{\boldsymbol{y}}/d\underset{\sim}{\boldsymbol{x}}$, such that

$$d\underset{\sim}{\boldsymbol{y}} = \left(\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}}\right) \bullet d\underset{\sim}{\boldsymbol{x}} \qquad (21.111)$$

Comparing with Eq. (21.110) shows that

$$\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial y_i}{\partial x_j}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \qquad (21.112)$$

If $\underset{\sim}{\boldsymbol{x}}$ is the position vector, note that $d\underset{\sim}{\boldsymbol{y}}/d\underset{\sim}{\boldsymbol{x}}$ is actually a *backward*-operating gradient.

SIDEBAR: "total" and "partial" derivative notation. Note that we have quietly introduced *our* notational preference to use a "d" when writing $d\underset{\sim}{\boldsymbol{y}}/d\underset{\sim}{\boldsymbol{x}}$ even though we must, by necessity, use a "∂" in the partial derivative on the right-hand-side indicial expansion. This choice has been made quite deliberately in order to maximize the analogies that tensor calculus has with ordinary scalar calculus. To get a better appreciation of why we have made this choice, consider the scalar calculus situation in which some variable y is expressible *solely* as a function of an independent variable x. Then, of course, you would naturally write dy/dx, not $\partial y/\partial x$. The very *act* of writing dy/dx tells you that y *really can* be written solely in terms of x. If it turns out that y can be *alternatively* written as a function of two other variables, *u* and *v*, then we could use the chain rule to write

[*] In Eq. (21.109) we have written $\partial y_i/\partial x_j$ on the reasonable assumption that you will understand this to mean the partial derivative with respect to x_j, *holding all other components of* $\underset{\sim}{\boldsymbol{x}}$ *constant.* Except in this very special case, we strongly recommend explicitly showing what variables are held constant in partial derivatives whenever there is even the slightest chance of misinterpretation or confusion by your readers. This issue pops up continually in thermodynamics. The partial derivative of pressure P with respect to volume V takes on different values depending on the conditions at which the measurement is made. If, for example, the variation of pressure with volume is measured at constant temperature T, then the derivative should be written $(\partial P/\partial V)_T$. However, if the measurement is made under reversible insulated (no heat flow) conditions, then the derivative should be written $(\partial P/\partial V)_s$, where s is entropy. The two derivatives aren't equal!

$$\frac{dy}{dx} = \left(\frac{\partial y}{\partial u}\right)_v \frac{du}{dx} + \left(\frac{\partial y}{\partial v}\right)_u \frac{dv}{dx} \qquad (21.113)$$

Note that we use “d” on the left-hand-side, and “∂’ on the right hand side where partial derivatives exist, and “d” again for the total derivatives of u and v. The situation with vector derivatives is very similar. We have a vector $\underset{\sim}{\boldsymbol{y}}$ that can be computed if we know the *single* vector $\underset{\sim}{\boldsymbol{x}}$. Just as an automobile is a single entity composed of hundreds of parts, we say $\underset{\sim}{\boldsymbol{x}}$ is a *single* vector because it really is a single entity even though it consists of multiple *parts* (components and base vectors). A velocity vector has three components, but you wouldn’t say “*the velocity ARE fast”* any more than you would say *“the automobile ARE fast.”* A vector IS the sum (not “are the sum”) of components times base vectors. A vector IS a single object. When $\underset{\sim}{\boldsymbol{y}} = f(\underset{\sim}{\boldsymbol{x}})$, we can alternatively regard $\underset{\sim}{\boldsymbol{y}}$ to be a function of the three RCS components x_1, x_2, and x_3. Then, in analog with Eq. (21.113) we would write

$$\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial \underset{\sim}{\boldsymbol{y}}}{\partial x_1}\frac{dx_1}{d\underset{\sim}{\boldsymbol{x}}} + \frac{\partial \underset{\sim}{\boldsymbol{y}}}{\partial x_2}\frac{dx_2}{d\underset{\sim}{\boldsymbol{x}}} + \frac{\partial \underset{\sim}{\boldsymbol{y}}}{\partial x_3}\frac{dx_3}{d\underset{\sim}{\boldsymbol{x}}} \qquad (21.114)$$

Note the distinctions in the use of “*d*” and “∂” in this equation. It’s not hard to show that, when the x_j are RCS components

$$\frac{dx_j}{d\underset{\sim}{\boldsymbol{x}}} = \underset{\sim}{\boldsymbol{e}}_j \qquad (21.115)$$

Likewise, since the base vectors themselves are fixed when using RCS coordinates,

$$\frac{\partial \underset{\sim}{\boldsymbol{y}}}{\partial x_j} = \frac{\partial (y_i \underset{\sim}{\boldsymbol{e}}_i)}{\partial x_j} = \frac{\partial y_i}{\partial x_j}\underset{\sim}{\boldsymbol{e}}_i \qquad (21.116)$$

Putting Eqs. (21.115) and (21.116) into (21.114), taking care to preserve the order of dyadic multiplication of the base vectors, gives us back our result of Eq. (21.112).

There are other analogs with conventional scalar calculus that motivate our use of “*d*” on the left hand side of Eq. (21.112). In scalar calculus, everyone knows that

$$\frac{dx}{dy} = \frac{1}{dy/dx} \qquad (21.117)$$

This identity holds if $dy/dx \neq 0$ (otherwise, we would be dividing by zero). The analog for tensors is

$$\frac{d\underset{\sim}{\boldsymbol{x}}}{d\underset{\sim}{\boldsymbol{y}}} = \left[\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}}\right]^{-1}, \qquad (21.118)$$

where the superscript “–1” is the *tensor inverse*. Of course, this property holds only if the inverse exists (i.e., only if $\det(d\underset{\sim}{\boldsymbol{y}}/d\underset{\sim}{\boldsymbol{x}}) \neq 0$), just as Eq. (21.117) holds only if $dy/dx \neq 0$.

Eq. (21.117) was applicable because y was expressible *solely* as a function of x. In general,

$$\frac{\partial x}{\partial y} \neq \frac{1}{\partial y/\partial x} \qquad (21.119)$$

Equality generally holds only if *both* partial derivatives hold the same variable constant in the derivative (in which case we are implicitly restricting our attention to a "world" in which that same variable is always constant; in such a world, y would be expressible *solely* as a function of x). To convince yourself that Eq. (21.119) is correct, consider polar coordinates in which x and y are expressible as functions of r and θ. Namely

$$x = r\cos\theta \qquad \text{and} \qquad y = r\sin\theta \tag{21.120}$$

from which it follows that

$$\begin{bmatrix} \left(\frac{\partial x}{\partial r}\right)_\theta & \left(\frac{\partial x}{\partial \theta}\right)_r \\ \left(\frac{\partial y}{\partial r}\right)_\theta & \left(\frac{\partial y}{\partial \theta}\right)_r \end{bmatrix} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \tag{21.121}$$

Equation (21.120) could, in principal, be inverted to obtain r and θ in terms of x and y to permit directly computing $(\partial r/\partial x)_y$. However, to compute $(\partial r/\partial x)_y$, you can alternatively use *implicit* differentiation to obtain

$$\begin{bmatrix} \left(\frac{\partial r}{\partial x}\right)_y & \left(\frac{\partial r}{\partial y}\right)_x \\ \left(\frac{\partial \theta}{\partial x}\right)_y & \left(\frac{\partial \theta}{\partial y}\right)_x \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial x}{\partial r}\right)_\theta & \left(\frac{\partial x}{\partial \theta}\right)_r \\ \left(\frac{\partial y}{\partial r}\right)_\theta & \left(\frac{\partial y}{\partial \theta}\right)_r \end{bmatrix}^{-1} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta \\ \frac{-\sin\theta}{r} & \frac{\cos\theta}{r} \end{bmatrix} \tag{21.122}$$

Note from Eq. (21.121) that

$$\left(\frac{\partial x}{\partial r}\right)_\theta = \cos\theta \tag{21.123}$$

and Eq. (21.122) tells us that

$$\left(\frac{\partial r}{\partial x}\right)_y = \cos\theta \tag{21.124}$$

Therefore,

$$\left(\frac{\partial r}{\partial x}\right)_y \neq \frac{1}{(\partial x/\partial r)_\theta} \tag{21.125}$$

If (for some strange reason) you wished to know the variation in r with respect to x holding θ constant, then you *would* be able to write

$$\left(\frac{\partial r}{\partial x}\right)_\theta = \frac{1}{(\partial x/\partial r)_\theta} \tag{21.126}$$

The only distinction between the above two equations is what variable is being held constant. In one case equality holds; in the other it doesn't — that's why it is so essential to explicitly show what's being held constant unless you are 100% sure there is no chance of confusion.* Don't be so sure. In classic multivariable, scalar, calculus, the very act of writing $\partial y/\partial x$ tells your reader that y can *not* be written solely in terms of x — it must be expressible in terms of x and whatever variable is being held constant in the partial derivative. That's why we do *not* use "∂" when we write $d\underset{\sim}{y}/d\underset{\sim}{x}$. We want you to know implicitly that $\underset{\sim}{y}$ can be written purely as a function of $\underset{\sim}{x}$ and nothing else. We would write $\partial\underset{\sim}{y}/\partial\underset{\sim}{x}$ only when $\underset{\sim}{y}$ depends additionally on other things, and we would be diligent to show what variable is held constant in the derivative if there is any chance for misinterpretation of what this second variable is.

The "nabla" or "del" gradient operator. Researchers who use $\partial y_j/\partial x_i$ typically do so because this convention permits them to use the alternative notation $\nabla\underset{\sim}{y}$, where the "nabla" or "del" operator is heuristically defined such that

$$\overset{\rightarrow}{\nabla}(\ ) = \frac{\partial(\ )}{\partial x_i}\underset{\sim}{e}_i, \tag{21.127}$$

With this "right-operating" del, the differentiation with respect to $\underset{\sim}{x}$ acts on quantities to the right. Thus, the ij component of $\overset{\rightarrow}{\nabla}\underset{\sim}{y}$ has indices that are heuristically similar to the ij component of an ordinary dyad $\underset{\sim}{a}\underset{\sim}{y}$, namely $a_i y_j$. In other words, to ensure structural self-consistency, $\overset{\rightarrow}{\nabla}\underset{\sim}{y}$ would have to be defined such that

$$d\underset{\sim}{y} = d\underset{\sim}{x} \bullet (\overset{\rightarrow}{\nabla}\underset{\sim}{y}), \text{ which implies that } (\overset{\rightrightarrows}{\nabla}\underset{\sim}{y})_{ij} = \frac{\partial y_j}{\partial x_i} \tag{21.128}$$

To use a heuristic backward-operating "del" operator with our definition, we would have to define

$$(\ )\overset{\leftarrow}{\nabla} = \partial(\ )/\partial x_j \underset{\sim}{e}_j, \tag{21.129}$$

which operates on arguments to its *left*. Then we could write

$$d\underset{\sim}{y}/d\underset{\sim}{x} = \underset{\sim}{y}\overset{\leftarrow}{\nabla} \tag{21.130}$$

Then the ij component of $\underset{\sim}{y}\overset{\leftarrow}{\nabla}$ would be heuristically similar to the ij components of $\underset{\sim}{y}\underset{\sim}{b}$. Thus, $\underset{\sim}{y}\overset{\leftarrow}{\nabla}$ would be defined such that

$$d\underset{\sim}{y} = (\underset{\sim}{y}\overset{\leftarrow}{\nabla}) \bullet d\underset{\sim}{x}, \text{ which implies that } (\underset{\sim}{y}\overset{\leftarrow}{\nabla})_{ij} = \frac{\partial y_i}{\partial x_j} \tag{21.131}$$

Both the backward and forward operating del are used by Malvern and we also find that they useful, depending on the application at hand. For the applications in this book, a "del" operator is never needed. We will always use the "fraction-like" notation of Eq. (21.112), which is technically a backward-operating derivative. With the fraction-like notation, the indices are placed on the dependent variable y_i and then on the independent variable x_j in the same order as the indices appear in the basis dyad $\underset{\sim}{e}_i\underset{\sim}{e}_j$. Because we will always follow this convention, we may write $d\underset{\sim}{y}/d\underset{\sim}{x}$ as

[*] An example where there is no chance of confusion is in expressions such as $\partial\phi/\partial x_i$. This partial derivative is with respect to x_i holding the other x_j constant. For example, $\partial\phi/\partial x_2$ is understood to mean $(\partial\phi/\partial x_2)_{x_1, x_3}$. Aside from this component-based exception, we recommend explicitly showing all other variables held constant. For example, $(\partial s/\partial x_2)_t$ is understood to mean $(\partial s/\partial x_2)_{x_1, x_3, t}$.

$$\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} \tag{21.132}$$

without ambiguity.

To ensure structural integrity, our convention of placing indices first on the dependent variable and then on the independent variable will extend to higher order derivatives as well. For example, the notation

$$\frac{d\underset{\approx}{\boldsymbol{U}}}{d\underset{\underset{\sim}{\approx}}{\boldsymbol{Y}}} \tag{21.133}$$

denotes a sixth-order tensor defined by

$$\frac{\partial U_{ij}}{\partial Y_{klmn}}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\underset{\sim}{\boldsymbol{e}}_k\underset{\sim}{\boldsymbol{e}}_l\underset{\sim}{\boldsymbol{e}}_m\underset{\sim}{\boldsymbol{e}}_n \tag{21.134}$$

In contrast to Eq. (21.132), when we write a derivative with the "∂" symbol instead of the "d" symbol, we are telling you that the dependent variable depends on more than one tensor or vector quantity. For example, by writing the notation

$$\left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} \tag{21.135}$$

we are implicitly telling you that $\underset{\sim}{\boldsymbol{y}}$ can be written as a proper function of $\underset{\sim}{\boldsymbol{x}}$ and $\underset{\sim}{\boldsymbol{z}}$. The subscript also indicates that the vector $\underset{\sim}{\boldsymbol{z}}$ is being held constant in the derivative. Thus, the above derivative quantifies how the vector $\underset{\sim}{\boldsymbol{y}}$ would change if $\underset{\sim}{\boldsymbol{x}}$ were to be varied without changing $\underset{\sim}{\boldsymbol{z}}$. For general variations in *both* independent variables, the *total* increment in $\underset{\sim}{\boldsymbol{y}}$ must include contributions from both $d\underset{\sim}{\boldsymbol{x}}$ and $d\underset{\sim}{\boldsymbol{z}}$. Specifically, the above partial derivative must be defined such that

$$d\underset{\sim}{\boldsymbol{y}} = \left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} \bullet d\underset{\sim}{\boldsymbol{x}} + \left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{z}}}\right)_{\underset{\sim}{\boldsymbol{x}}} \bullet d\underset{\sim}{\boldsymbol{z}} \tag{21.136}$$

As before, if $\underset{\sim}{\boldsymbol{y}}$ is a function of $\underset{\sim}{\boldsymbol{x}}$ and $\underset{\sim}{\boldsymbol{z}}$, then y_i depends on the x_j and z_k components. Hence,

$$\left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} = \left(\frac{\partial y_i}{\partial x_j}\right)_{\underset{\sim}{\boldsymbol{z}}}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j\,. \tag{21.137}$$

By the way, note that

$$\left(\frac{\partial\underset{\sim}{\boldsymbol{x}}}{\partial\underset{\sim}{\boldsymbol{y}}}\right)_{\underset{\sim}{\boldsymbol{z}}} = \left[\left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}}\right]^{-1} \tag{21.138}$$

This is true because the *same quantity* — namely, $\underset{\sim}{\boldsymbol{z}}$ — is being held constant on both sides of the equation. The property is not true if different things are held constant. Thus,

$$\left(\frac{\partial\underset{\sim}{\boldsymbol{x}}}{\partial\underset{\sim}{\boldsymbol{y}}}\right)_{\underset{\sim}{\boldsymbol{z}}} \neq \left[\left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{w}}}\right]^{-1} \text{ in general} \tag{21.139}$$

Okay, if the above relation does not hold, does anything LIKE IT hold?

If you really wish to relate $(\partial\underset{\sim}{\boldsymbol{x}}/\partial\underset{\sim}{\boldsymbol{y}})_{\underset{\sim}{\boldsymbol{z}}}$ to $(\partial\underset{\sim}{\boldsymbol{y}}/\partial\underset{\sim}{\boldsymbol{x}})_{\underset{\sim}{\boldsymbol{w}}}$, then you must be extra careful with your logic. The basic motivation for seeking such a relationship in the first place is that you must not have the actual function for $\underset{\sim}{\boldsymbol{x}}$ as a function of $\underset{\sim}{\boldsymbol{y}}$ and $\underset{\sim}{\boldsymbol{z}}$ available. Instead, you have $\underset{\sim}{\boldsymbol{y}}$ as a function of $\underset{\sim}{\boldsymbol{x}}$ and some other variable $\underset{\sim}{\boldsymbol{w}}$. Recall that the very *act* of writing $(\partial\underset{\sim}{\boldsymbol{x}}/\partial\underset{\sim}{\boldsymbol{y}})_{\underset{\sim}{\boldsymbol{z}}}$ tells your reader that there does *exist* some function for $\underset{\sim}{\boldsymbol{x}}$ as a function of $\underset{\sim}{\boldsymbol{y}}$ and $\underset{\sim}{\boldsymbol{z}}$ even if you don't happen to have that function available at hand. Now, if $\underset{\sim}{\boldsymbol{x}}$ is a function of $\underset{\sim}{\boldsymbol{y}}$ and $\underset{\sim}{\boldsymbol{z}}$, then it might be possible to solve this relationship to obtain $\underset{\sim}{\boldsymbol{y}}$ as a function of $\underset{\sim}{\boldsymbol{x}}$ and $\underset{\sim}{\boldsymbol{z}}$. The very act of writing the expression $(\partial\underset{\sim}{\boldsymbol{y}}/\partial\underset{\sim}{\boldsymbol{x}})_{\underset{\sim}{\boldsymbol{w}}}$, implies that $\underset{\sim}{\boldsymbol{y}}$ must be expressible as a function of $\underset{\sim}{\boldsymbol{x}}$ and some other variable $\underset{\sim}{\boldsymbol{w}}$. Thus, you have simultaneous *implied* functions

$$\underset{\sim}{\boldsymbol{y}} = \underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}}, \underset{\sim}{\boldsymbol{z}}) \text{ and } \underset{\sim}{\boldsymbol{y}} = \underset{\sim}{\boldsymbol{y}}(\underset{\sim}{\boldsymbol{x}}, \underset{\sim}{\boldsymbol{w}}) \tag{21.140}$$

The only way these can both be true is if

$$\underset{\sim}{\boldsymbol{w}} = \underset{\sim}{\boldsymbol{w}}(\underset{\sim}{\boldsymbol{x}}, \underset{\sim}{\boldsymbol{z}}) \tag{21.141}$$

Now you can apply the chain rule to write

$$\left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} = \left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{w}}} \bullet \left(\frac{\partial\underset{\sim}{\boldsymbol{x}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} + \left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{w}}}\right)_{\underset{\sim}{\boldsymbol{x}}} \bullet \left(\frac{\partial\underset{\sim}{\boldsymbol{w}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} \tag{21.142}$$

or, noting that $(\partial\underset{\sim}{\boldsymbol{x}}/\partial\underset{\sim}{\boldsymbol{x}})_{\underset{\sim}{\boldsymbol{z}}}$ is just the identity tensor,

$$\left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} = \left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{w}}} + \left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{w}}}\right)_{\underset{\sim}{\boldsymbol{x}}} \bullet \left(\frac{\partial\underset{\sim}{\boldsymbol{w}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}} \tag{21.143}$$

Putting this into Eq. (21.138) gives

$$\left(\frac{\partial\underset{\sim}{\boldsymbol{x}}}{\partial\underset{\sim}{\boldsymbol{y}}}\right)_{\underset{\sim}{\boldsymbol{z}}} = \left[\left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{w}}} + \left(\frac{\partial\underset{\sim}{\boldsymbol{y}}}{\partial\underset{\sim}{\boldsymbol{w}}}\right)_{\underset{\sim}{\boldsymbol{x}}} \bullet \left(\frac{\partial\underset{\sim}{\boldsymbol{w}}}{\partial\underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{z}}}\right]^{-1} \tag{21.144}$$

Another set of commonly-used identities from single-variable scalar calculus is

$$\frac{dy}{dx} = \frac{dy/dw}{dx/dw} \qquad \text{and} \qquad \frac{dy}{dx} = \frac{dw/dx}{dw/dy} \tag{21.145}$$

The analogs in tensor analysis are

$$\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} = \frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{w}}} \bullet \left[\frac{d\underset{\sim}{\boldsymbol{x}}}{d\underset{\sim}{\boldsymbol{w}}}\right]^{-1} \qquad \text{and} \qquad \frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} = \left[\frac{d\underset{\sim}{\boldsymbol{w}}}{d\underset{\sim}{\boldsymbol{y}}}\right]^{-1} \bullet \frac{d\underset{\sim}{\boldsymbol{w}}}{d\underset{\sim}{\boldsymbol{x}}} \tag{21.146}$$

Note that the inverted tensor is on the right side of the dot product in one case but on the left in the other. The way to determine which one is correct is to start with the chain rule:

$$\frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{w}}} = \frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} \bullet \frac{d\underset{\sim}{\boldsymbol{x}}}{d\underset{\sim}{\boldsymbol{w}}} \qquad \text{and} \qquad \frac{d\underset{\sim}{\boldsymbol{w}}}{d\underset{\sim}{\boldsymbol{x}}} = \frac{d\underset{\sim}{\boldsymbol{w}}}{d\underset{\sim}{\boldsymbol{y}}} \bullet \frac{d\underset{\sim}{\boldsymbol{y}}}{d\underset{\sim}{\boldsymbol{x}}} \tag{21.147}$$

Solving these expressions for $d\underset{\sim}{\boldsymbol{y}}/d\underset{\sim}{\boldsymbol{x}}$ gives Eq. (21.145).

For *multi-variable* scalar calculus, the identities equivalent to (21.145) are

$$\left(\frac{\partial y}{\partial x}\right)_s = \frac{(\partial y/\partial w)_s}{(\partial x/\partial w)_s} \qquad \text{and} \qquad \left(\frac{\partial y}{\partial x}\right)_s = \frac{(\partial w/\partial x)_s}{(\partial w/\partial y)_s} \tag{21.148}$$

These identities are truly equivalent to (21.145) because the same variable, s, is being held constant in all of the partial derivatives.

Another *fundamentally different* but similar identity states that

$$\boxed{\left(\frac{\partial y}{\partial x}\right)_s = -\frac{(\partial s/\partial x)_y}{(\partial s/\partial y)_x}} \tag{21.149}$$

Note that this identity is completely different from the second one in Eq. (21.148) because of the variables held constant in the derivative. To derive this result, start with the basics. We are interested in converting a derivative for which s is an independent variable held constant into an expression where s is a dependent variable. The act of writing $(\partial y/\partial x)_s$ tells us that $y = y(x, s)$. In principle, we could solve this function for $s = s(x, y)$ and it is this function that is being differentiated on the right-hand-side of Eq. (21.149). Writing out the chain rule for this function gives

$$ds = \frac{\partial s}{\partial x}dx + \frac{\partial s}{\partial y}dy \tag{21.150}$$

This expression holds for arbitrary variations of x and y. However, recalling that we are trying to obtain an expression for $(\partial y/\partial x)_s$, what we really want to consider is variations of x and y that will involve no changes in s — in other words, we are restricting attention to motions for which $ds = 0$. Note that

$$\left(\frac{\partial s}{\partial \xi}\right)_s = 0 \tag{21.151}$$

regardless of what variable ξ represents (other than s itself, of course). This equation is simply asserting that the variation of s holding s constant must be zero. We can therefore write the total increment chain rule of Eq. (21.150) in terms of partial derivatives holding s constant:

$$0 = \frac{\partial s}{\partial x}\left(\frac{\partial x}{\partial \xi}\right)_s + \frac{\partial s}{\partial y}\left(\frac{\partial y}{\partial \xi}\right)_s \tag{21.152}$$

Now let's make a choice for the variable ξ. Taking ξ to denote x, the above equation becomes

$$0 = \frac{\partial s}{\partial x}\left(\frac{\partial x}{\partial x}\right)_s + \frac{\partial s}{\partial y}\left(\frac{\partial y}{\partial x}\right)_s \tag{21.153}$$

or, since $\partial x/\partial x = 1$ no matter what is being held constant (except x itself of course)

$$0 = \frac{\partial s}{\partial x} + \frac{\partial s}{\partial y}\left(\frac{\partial y}{\partial x}\right)_s \tag{21.154}$$

Solving for $(\partial y/\partial x)_s$ completes the proof of Eq. (21.149).

For tensor calculus, the analog of Eq. (21.154) is

$$\underset{\approx}{0} = \frac{\partial \underset{\sim}{\boldsymbol{s}}}{\partial \underset{\sim}{\boldsymbol{x}}} + \frac{\partial \underset{\sim}{\boldsymbol{s}}}{\partial \underset{\sim}{\boldsymbol{y}}} \bullet \left(\frac{\partial \underset{\sim}{\boldsymbol{y}}}{\partial \underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{s}}} \tag{21.155}$$

or

$$\left(\frac{\partial \underset{\sim}{\boldsymbol{y}}}{\partial \underset{\sim}{\boldsymbol{x}}}\right)_{\underset{\sim}{\boldsymbol{s}}} = -\left[\frac{\partial \underset{\sim}{\boldsymbol{s}}}{\partial \underset{\sim}{\boldsymbol{y}}}\right]^{-1} \bullet \frac{\partial \underset{\sim}{\boldsymbol{s}}}{\partial \underset{\sim}{\boldsymbol{x}}} \tag{21.156}$$

Note the rule of thumb: to derived tensor generalizations of familiar calculus derivative identities, you will always want to start with the chain rule.

Directed derivative

Consider a function f that depends only on $\underset{\sim}{\boldsymbol{x}}$. The value of the function may be a scalar, vector, or tensor of any order. Recall that

$$\frac{df}{d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial f}{\partial x_i}\underset{\sim}{\boldsymbol{e}}_i \tag{21.157}$$

The gradient characterizes how the quantity $f(\underset{\sim}{\boldsymbol{x}})$ changes in response to an incremental but otherwise arbitrary change in $\underset{\sim}{\boldsymbol{x}}$. Let us now be more specific. Suppose that you are interested in simplifying the problem to a special case where $\underset{\sim}{\boldsymbol{x}}$ is known to change from some initial position $\underset{\sim}{\boldsymbol{x}}_0$ to a current position $\underset{\sim}{\boldsymbol{x}}$ such that it moves from the first to final point along a straight line segment. The vector connecting these two points can be defined

$$\underset{\sim}{\boldsymbol{s}} \equiv \underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_0 \tag{21.158}$$

The unit vector in the direction of the line segment $\underset{\sim}{\boldsymbol{s}}$ is defined simply

$$\underset{\sim}{\boldsymbol{n}} \equiv \frac{\underset{\sim}{\boldsymbol{s}}}{\sqrt{\underset{\sim}{\boldsymbol{s}} \bullet \underset{\sim}{\boldsymbol{s}}}}, \text{ where } s \equiv \sqrt{\underset{\sim}{\boldsymbol{s}} \bullet \underset{\sim}{\boldsymbol{s}}} \tag{21.159}$$

so that the position increment may be written

$$\underset{\sim}{\boldsymbol{s}} = s\underset{\sim}{\boldsymbol{n}} \tag{21.160}$$

When restricting attention to situations where $\underset{\sim}{\boldsymbol{x}}$ moves only along the line segment $\underset{\sim}{\boldsymbol{s}}$, then it makes sense to consider the scalar s to quantify the location. Using just s makes computations easier because it's always easier to work with scalars than with vectors. The **directed derivative** is the derivative of f with respect to a change in distance s along the line segment in the direction of $\underset{\sim}{\boldsymbol{n}}$.

$$\frac{\partial f}{\partial s} \equiv \frac{df}{d\underset{\sim}{\boldsymbol{x}}} \bullet \underset{\sim}{\boldsymbol{n}} \tag{21.161}$$

Our use of partial derivatives on the left hand side is a break with traditional notation, but consistent with our self-imposed notation structure rules. We use partial derivatives here to emphasize that we are examining the change in f with respect to $\underset{\sim}{\boldsymbol{x}}$, holding the allowable *direction* of the change constant.

EXAMPLE. Consider

$$f = \underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}} \qquad (21.162)$$

Then

$$\frac{df}{d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial f}{\partial x_i}\underset{\sim}{\boldsymbol{e}}_i = \frac{\partial(x_k x_k)}{\partial x_i}\underset{\sim}{\boldsymbol{e}}_i = (\delta_{ki}x_k + x_k\delta_{ki})\underset{\sim}{\boldsymbol{e}}_i = 2x_i\underset{\sim}{\boldsymbol{e}}_i = 2\underset{\sim}{\boldsymbol{x}} \qquad (21.163)$$

Suppose you wish to characterize how f changes when $\underset{\sim}{\boldsymbol{x}}$ moves in a direction that is parallel to $\underset{\sim}{\boldsymbol{x}}$ itself. In this case,

$$\underset{\sim}{\boldsymbol{n}} = \frac{\underset{\sim}{\boldsymbol{x}}}{\sqrt{\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}}}} \qquad (21.164)$$

and therefore the directed derivative is

$$\frac{df}{d\underset{\sim}{\boldsymbol{x}}} \bullet \underset{\sim}{\boldsymbol{n}} = (2\underset{\sim}{\boldsymbol{x}}) \bullet \frac{\underset{\sim}{\boldsymbol{x}}}{\sqrt{\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}}}} = 2\sqrt{\underset{\sim}{\boldsymbol{x}} \bullet \underset{\sim}{\boldsymbol{x}}} = 2\|\underset{\sim}{\boldsymbol{x}}\| \qquad (21.165)$$

If you let r denote the magnitude of $\underset{\sim}{\boldsymbol{x}}$, this means that

$$\frac{\partial f}{\partial r} = 2r. \qquad (21.166)$$

Given the way that the function is defined, you should have realized by now that a natural choice for the basis is *spherical* $\{\underset{\sim}{\boldsymbol{e}}_r, \underset{\sim}{\boldsymbol{e}}_\theta, \underset{\sim}{\boldsymbol{e}}_\psi\}$. As a matter of fact, the function f in Eq. (21.162) can be alternatively written in terms of spherical coordinates as

$$f = r^2 \qquad (21.167)$$

making the result of Eq. (21.166) far less mysterious, and further indicating that the quantities held constant in the partial derivative are simply the other two spherical coordinates θ and ψ. We're talking about variations in f resulting from *radial* motion. Furthermore, in terms of spherical coordinates, Eq. (21.164) is simply

$$\frac{df}{d\underset{\sim}{\boldsymbol{x}}} = 2r\underset{\sim}{\boldsymbol{e}}_r \qquad (21.168)$$

Now suppose that you wish to consider how the function f changes in response to a change in $\underset{\sim}{\boldsymbol{x}}$ that is *perpendicular* to $\underset{\sim}{\boldsymbol{x}}$. In Eq. (21.165) we showed that the unit vector in the direction of $\underset{\sim}{\boldsymbol{x}}$ happens to equal $\underset{\sim}{\boldsymbol{e}}_r$. Therefore, $\underset{\sim}{\boldsymbol{e}}_\theta$ is perpendicular to $\underset{\sim}{\boldsymbol{x}}$, and it turns out that the directed derivative corresponding to changes in $\underset{\sim}{\boldsymbol{x}}$ that move in the direction of $\underset{\sim}{\boldsymbol{e}}_\theta$ are zero:

$$\frac{\partial f}{\partial \theta} = \frac{df}{d\underset{\sim}{\boldsymbol{x}}} \bullet \underset{\sim}{\boldsymbol{e}}_\theta = (2r\underset{\sim}{\boldsymbol{e}}_r) \bullet \underset{\sim}{\boldsymbol{e}}_\theta = 0, \qquad (21.169)$$

meaning that the value of f does not change when $\underset{\sim}{\boldsymbol{x}}$ moves in the direction of $\underset{\sim}{\boldsymbol{e}}_\theta$, again not a surprising result in light of Eq. (21.167).

To summarize, the directed derivative is best regarded as simply a way for you to explore in a step-by-step manner how a function changes in response to changes in $\underset{\sim}{\boldsymbol{x}}$. All you have to do is simply consider different directions for $\underset{\sim}{\boldsymbol{x}}$ to move in order to gain insight into the field variability of f. Incidentally, note that $\partial f/\partial x_i$ is itself a directed derivative, given by

$$\frac{\partial f}{\partial x_i} = \frac{df}{d\underset{\sim}{\boldsymbol{x}}} \bullet \underset{\sim}{\boldsymbol{e}}_i \qquad (21.170)$$

Derivatives in reduced dimension spaces

Suppose that you know that some scalar s can be expressed as a function of a vector $\underset{\sim}{\boldsymbol{x}}$. Then, *nominally*, you may write $ds/d\underset{\sim}{\boldsymbol{x}} = (\partial s/\partial x_i)\underset{\sim}{\boldsymbol{e}}_i$. However, this is not really the correct result if the vector $\underset{\sim}{\boldsymbol{x}}$ is not allowed to take any possible variation. If, for example, the vector $\underset{\sim}{\boldsymbol{x}}$ is constrained such that it must always be a unit vector (i.e., if $\underset{\sim}{\boldsymbol{x}}$ must always point to a location on the unit sphere), then radial variations of the vector $\underset{\sim}{\boldsymbol{x}}$ are not permissible. For situations like these, the definition of the derivative $ds/d\underset{\sim}{\boldsymbol{x}}$ must be redefined such that

$$ds = \left(\frac{ds}{d\underset{\sim}{\boldsymbol{x}}}\right) \bullet d\underset{\sim}{\boldsymbol{x}} \text{ for all \textit{permissible} variations in } \underset{\sim}{\boldsymbol{x}}. \qquad (21.171)$$

The way to compute this revised definition of the gradient is to *first* compute a nominal gradient on the assumption that even inadmissible variations in $\underset{\sim}{\boldsymbol{x}}$ are allowed. Then this nominal result must be modified to remove the inadmissibilities. Without this modification, the gradient will not be unique and could even lead to strange counterintuitive results!

As an introduction to this important topic, consider a one-dimensional curve embedded in 3D space. Such a curve can be defined parametrically such that for each value of a scalar s there exists a unique point $\underset{\sim}{\boldsymbol{x}}$ on the curve. As a specific example, consider

$$\underset{\sim}{\boldsymbol{x}} = s\underset{\sim}{\boldsymbol{E}}_1 + e^s\underset{\sim}{\boldsymbol{E}}_2 \qquad (21.172)$$

The parameter s is here assumed to vary freely from $-\infty$ to ∞. The non-parametric equation for this space curve is simply $x_2 = e^{x_1}$ and $x_3 = 0$.

The derivative of $\underset{\sim}{\boldsymbol{x}}$ with respect to s is

$$\frac{d\underset{\sim}{\boldsymbol{x}}}{ds} = \underset{\sim}{\boldsymbol{E}}_1 + e^s\underset{\sim}{\boldsymbol{E}}_2 \qquad (21.173)$$

Not surprisingly, this result will always be tangent to the curve.

Because the curve is one-dimensional, we know that each position $\underset{\sim}{\boldsymbol{x}}$ corresponds to a unique value of the parameter s. Conversely, each value of s is associated with exactly one position vector $\underset{\sim}{\boldsymbol{x}}$ on the curve. Hence, s may be regarded as a proper function of $\underset{\sim}{\boldsymbol{x}}$. However, what exactly is this function? Considering the coefficients of $\underset{\sim}{\boldsymbol{E}}_1$ in Eq. (21.128), you can assert that

$$s = x_1 \tag{21.174}$$

but you can alternatively consider the coefficients of $\underset{\sim}{\boldsymbol{E}}_2$ to claim that

$$s = \ln x_2 \tag{21.175}$$

Both of these expressions will give the correct value for s as long as $\underset{\sim}{\boldsymbol{x}}$ lies on the space curve. However, directly differentiating these different expressions for s with respect to $\underset{\sim}{\boldsymbol{x}}$ gives different answers. We will call these derivatives the "nominal" derivatives, and we will denote them with a star:

$$\text{Eq. (21.174) gives } \left(\frac{ds}{d\underset{\sim}{\boldsymbol{x}}}\right)^{*} = \underset{\sim}{\boldsymbol{E}}_1 \tag{21.176}$$

$$\text{Eq. (21.175) gives } \left(\frac{ds}{d\underset{\sim}{\boldsymbol{x}}}\right)^{*} = \frac{\underset{\sim}{\boldsymbol{E}}_2}{x_2} \tag{21.177}$$

These two expressions are perpendicular to each other, so they cannot possibly be equal. The discrepancy must be resolved by using a projection of the increment $d\underset{\sim}{\boldsymbol{x}}$ onto the space of allowable increments. From Eq. (21.172), note that

$$d\underset{\sim}{\boldsymbol{x}} = (\underset{\sim}{\boldsymbol{E}}_1 + e^s \underset{\sim}{\boldsymbol{E}}_2)\ ds \tag{21.178}$$

Therefore, the increment in position must always be parallel to the vector

$$\underset{\sim}{\boldsymbol{b}} \equiv \underset{\sim}{\boldsymbol{E}}_1 + e^s \underset{\sim}{\boldsymbol{E}}_2 \tag{21.179}$$

You can define the projector onto this vector by using Eq. (10.14):

$$\underset{\approx}{\boldsymbol{P}} = \frac{\underset{\sim}{\boldsymbol{b}} \otimes \underset{\sim}{\boldsymbol{b}}}{\underset{\sim}{\boldsymbol{b}} \bullet \underset{\sim}{\boldsymbol{b}}} = \frac{\underset{\sim}{\boldsymbol{E}}_1 \underset{\sim}{\boldsymbol{E}}_1 + e^s \underset{\sim}{\boldsymbol{E}}_2 \underset{\sim}{\boldsymbol{E}}_1 + e^s \underset{\sim}{\boldsymbol{E}}_1 \underset{\sim}{\boldsymbol{E}}_2 + e^{2s} \underset{\sim}{\boldsymbol{E}}_2 \underset{\sim}{\boldsymbol{E}}_2}{1 + e^{2s}} \tag{21.180}$$

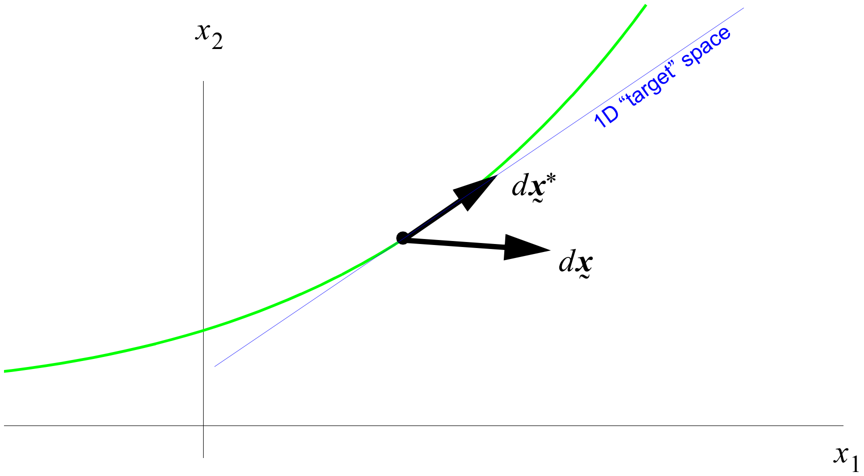

x_2
1D "target" space
$d\underset{\sim}{\mathbf{x}}^*$
$d\underset{\sim}{\mathbf{x}}$
x_1

Figure 21.1. Projecting an arbitrary position increment onto the space of allowable position increments.

If $d\underset{\sim}{\mathbf{x}}$ is an arbitrary increment in $\underset{\sim}{\mathbf{x}}$ that may or may not lie on the space curve (see Fig. 21.1), you can write the set of all *allowable* increments in the form

$$(d\underset{\sim}{\mathbf{x}})^* = \underset{\approx}{\mathbf{P}} \bullet d\underset{\sim}{\mathbf{x}} \tag{21.181}$$

To make use of the definition of a gradient, the increment in s must be expressed in terms of *truly arbitrary* increments in $\underset{\sim}{\mathbf{x}}$, not just constrained increments. This is done by replacing the constrained increment $(d\underset{\sim}{\mathbf{x}})^*$ by $\underset{\approx}{\mathbf{P}} \bullet d\underset{\sim}{\mathbf{x}}$, which involves truly arbitrary increments:

$$ds = \left(\frac{ds}{d\underset{\sim}{\mathbf{x}}}\right)^* \bullet (d\underset{\sim}{\mathbf{x}})^* = \left(\frac{ds}{d\underset{\sim}{\mathbf{x}}}\right)^* \bullet \underset{\approx}{\mathbf{P}} \bullet d\underset{\sim}{\mathbf{x}} \tag{21.182}$$

Applying the definition of gradients, it follows that the proper expression for the derivative of s with respect to allowable increments in position must be given by

$$\frac{ds}{d\underset{\sim}{\mathbf{x}}} = \left(\frac{ds}{d\underset{\sim}{\mathbf{x}}}\right)^* \bullet \underset{\approx}{\mathbf{P}} \tag{21.183}$$

When applied to Eq. (21.176) this formula gives

$$\frac{ds}{d\underset{\sim}{\mathbf{x}}} = \frac{\underset{\sim}{\mathbf{E}}_1 + e^s \underset{\sim}{\mathbf{E}}_2}{1 + e^{2s}} \tag{21.184}$$

and when applied to Eq. (21.177), it gives

$$\frac{ds}{d\underset{\sim}{\mathbf{x}}} = \frac{\underset{\sim}{\mathbf{E}}_1 + e^s \underset{\sim}{\mathbf{E}}_2}{1 + e^{2s}} \tag{21.185}$$

The above two results are identical — the paradox has been resolved! This unique result should be used whenever the derivative of s with respect to $\underset{\sim}{\boldsymbol{x}}$ is needed. Using this expression for the derivative also leads to intuitive results for chain rule applications. For example, the chain rule tells us that

$$\frac{d\underset{\sim}{\boldsymbol{x}}}{d\underset{\sim}{\boldsymbol{x}}} = \frac{d\underset{\sim}{\boldsymbol{x}}}{ds}\frac{ds}{d\underset{\sim}{\boldsymbol{x}}} \tag{21.186}$$

Substituting Eqs. (21.183) and (21.184) into the right hand side leads to

$$\frac{d\underset{\sim}{\boldsymbol{x}}}{d\underset{\sim}{\boldsymbol{x}}} = \underset{\approx}{\boldsymbol{P}} \tag{21.187}$$

Thus, for this case of *constrained* increments $d\underset{\sim}{\boldsymbol{x}}^*$, the derivative of $\underset{\sim}{\boldsymbol{x}}$ with respect to itself is *not* the identity tensor — it is the projection operator that takes you to the space of allowable increments in $\underset{\sim}{\boldsymbol{x}}$. For allowable increments $(d\underset{\sim}{\boldsymbol{x}})^*$, the projector has the property

$$\underset{\approx}{\boldsymbol{P}} \bullet (d\underset{\sim}{\boldsymbol{x}})^* = (d\underset{\sim}{\boldsymbol{x}})^* \tag{21.188}$$

Thus, the projection operator is the identity tensor along the space curve!

Another way of looking at this issue is to define a variable $f \equiv x_2 - e^{x_1}$ so that the actual curve of interest corresponds to $f = 0$. The advantage of this approach is outlined in general below.

Suppose you are given $s = s(\underset{\sim}{\boldsymbol{x}})$ where $\underset{\sim}{\boldsymbol{x}}$ is constrained such that it satisfies $f(\underset{\sim}{\boldsymbol{x}}) = 0$ for some constraint function f. Then you have two equations that must be satisfied simultaneously:

$$ds = \underset{\sim}{\boldsymbol{g}} \bullet d\underset{\sim}{\boldsymbol{x}}\text{, where } \underset{\sim}{\boldsymbol{g}} = \left(\frac{ds}{d\underset{\sim}{\boldsymbol{x}}}\right)^*\text{, and} \tag{21.189}$$

$$df = \underset{\sim}{\boldsymbol{h}} \bullet d\underset{\sim}{\boldsymbol{x}} = 0\text{, where } \underset{\sim}{\boldsymbol{h}} \equiv \frac{df}{dx_i}\underset{\sim}{\boldsymbol{e}}_i \tag{21.190}$$

The second equation tells us that $d\underset{\sim}{\boldsymbol{x}} = d\underset{\sim}{\boldsymbol{x}} - \hat{\underset{\sim}{\boldsymbol{h}}}\hat{\underset{\sim}{\boldsymbol{h}}} \bullet d\underset{\sim}{\boldsymbol{x}}$, where $\hat{\underset{\sim}{\boldsymbol{h}}} \equiv \underset{\sim}{\boldsymbol{h}} / \sqrt{\underset{\sim}{\boldsymbol{h}} \bullet \underset{\sim}{\boldsymbol{h}}}$. This leads to the general formula for a constrained gradient:

$$\frac{ds}{d\underset{\sim}{\boldsymbol{x}}} = \underset{\sim}{\boldsymbol{g}} \bullet \underset{\approx}{\boldsymbol{P}},$$

$$\text{where } \underset{\approx}{\boldsymbol{P}} = \underset{\approx}{\boldsymbol{I}} - \hat{\underset{\sim}{\boldsymbol{h}}}\hat{\underset{\sim}{\boldsymbol{h}}}\text{, } \underset{\sim}{\boldsymbol{g}} = \left(\frac{ds}{d\underset{\sim}{\boldsymbol{x}}}\right)^*\text{, } \underset{\sim}{\boldsymbol{h}} \equiv \frac{df}{dx_i}\underset{\sim}{\boldsymbol{e}}_i\text{, and } \hat{\underset{\sim}{\boldsymbol{h}}} \equiv \frac{\underset{\sim}{\boldsymbol{h}}}{\sqrt{\underset{\sim}{\boldsymbol{h}} \bullet \underset{\sim}{\boldsymbol{h}}}} \tag{21.191}$$

This formula extends similarly for dependent and independent variables of any tensor order. It also extends to the case of multiple constraints. One serious issue with this formula relates to the possibility that $\underset{\sim}{\boldsymbol{h}}$ might turn out to be zero. If the constraint is $f{=}0$, then a perfectly legitimate and equivalent alternative constraint would be $f^* \equiv f^2 = 0$. When using f^* instead of f, the vector $\underset{\sim}{\boldsymbol{h}}^* \equiv \frac{df^*}{dx_i}\underset{\sim}{\boldsymbol{e}}_i = 2f\underset{\sim}{\boldsymbol{h}}$ would turn out to be zero within the constrained space, and the above formula would break down because it would not be possible to compute a unit vector in the direction of $\underset{\sim}{h}^*$. For this reason, it's important to use a careful procedural means of determining the projector $\underset{\approx}{P}$ that converts an arbitrary increment into an admissible increment.

A more physically significant example. In materials modeling, tensors are frequently symmetric. The projection operator that transforms any tensor into its symmetric part is given by

$$P^{\text{sym}}_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{21.192}$$

For nonlinear elasticity, the stress tensor σ_{pq} is taken to be a function of the strain tensor ε_{mn} . In rate form, nonlinear elasticity is therefore of the form

$$\dot{\sigma}_{pq} = E_{pqmn}\dot{\varepsilon}_{mn} \tag{21.193}$$

where the tangent stiffness tensor is defined

$$E_{pqmn} = \frac{\partial\sigma_{pq}}{\partial\varepsilon_{mn}} \tag{21.194}$$

Even though $\varepsilon_{mn} = \varepsilon_{nm}$, it is not automatically true that $E_{pqmn} = E_{pqnm}$. You have to *force* this property to hold by imposing the constrained increment projector onto this nominal derivative.*

Because the strain tensor is symmetric, the tangent stiffness tensor can (*and should*) be replaced by

$$E_{pqmn} = \left(\frac{\partial\sigma_{pq}}{\partial\varepsilon_{ij}}\right)P^{\text{sym}}_{ijmn} \tag{21.195}$$

Doing this will ensure consistent results for derivatives, and you won't end up with pathological problems similar to those seen in the previous section. This is an extremely important (and poorly attended to) concept.

* To convince yourself of this assertion, consider the very simple stress-strain relationship $\sigma_{ij} = \alpha\varepsilon_{ij}$, where α is a constant. Direct differentiation would give the nominal result $E_{ijmn} = \alpha\delta_{im}\delta_{jn}$, which does not satisfy $E_{pqmn} = E_{pqnm}$. Minor symmetry must be imposed after the nominal differentiation to obtain $E_{ijmn} = \frac{\alpha}{2}(\delta_{im}\delta_{jn} + \delta_{in}\delta_{jm})$.

Series expansion of a nonlinear vector function

The concept of expanding a nonlinear vector function as a series expansion is so useful that we will now close this book with it. In undergraduate calculus, the expansion of a function of one variable $f(x)$ about a point x_o is given as

$$f(x) = f(x_o) + f'(x_o)(x - x_o) + \frac{1}{2!}f''(x_o)(x - x_o)^2 + \frac{1}{3!}f'''(x_o)(x - x_o)^3 + \dots \qquad (21.196)$$

Now consider a function of two variables, $f(x, y)$. The expansion of this function about the point (x_o, y_o) is given by

$$\begin{aligned} f(x, y) = {} & f^o + f^o_x(x - x_o) + \frac{1}{2!}f^o_{xx}(x - x_o)^2 \\ & + f^o_y(y - y_o) + \frac{1}{2!}f^o_{yy}(y - y_o)^2 \\ & + \frac{1}{2!}f^o_{xy}(x - x_o)(y - y_o) \\ & + \frac{1}{2!}f^o_{yx}(y - y_o)(x - x_o) + \dots \end{aligned} \qquad (21.197)$$

where

$$\begin{aligned} & f^o \equiv f(x_o, y_o) \\ & f^o_x = \frac{\partial f(x, y)}{\partial x} \text{ evaluated at } (x, y) = (x_o, y_o) \\ & f^o_y = \frac{\partial f(x, y)}{\partial y} \text{ evaluated at } (x, y) = (x_o, y_o) \\ & f^o_{xx} = \frac{\partial^2 f(x, y)}{\partial x^2} \text{ evaluated at } (x, y) = (x_o, y_o) \\ & f^o_{yy} = \frac{\partial^2 f(x, y)}{\partial y^2} \text{ evaluated at } (x, y) = (x_o, y_o) \\ & f^o_{xy} = \frac{\partial^2 f(x, y)}{\partial x \partial y} \text{ evaluated at } (x, y) = (x_o, y_o) \\ & f^o_{xy} = \frac{\partial^2 f(x, y)}{\partial x \partial y} \text{ evaluated at } (x, y) = (x_o, y_o) \quad = \quad f^o_{yx} \end{aligned} \qquad (21.198)$$

Note that the expansion is similar in structure to the expansion of a function of a single variable except that there is one derivative of every order for every possible number of ways to select the differentiation variables. Even though $f^o_{xy} = f^o_{yx}$, this mixed partial derivative still counts twice in the expansion.

Now consider a field variable $f(\underset{\sim}{\boldsymbol{x}})$. The vector $\underset{\sim}{\boldsymbol{x}}$ has three components, so this is a function of *three* variables. In direct notation, the expansion of this field variable about a point $\underset{\sim}{\boldsymbol{x}}_o$ is

$$f(\underset{\sim}{\boldsymbol{x}}) = f_o + \underset{\sim}{\boldsymbol{f}'}_o \bullet (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o) + \frac{1}{2!}(\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o) \bullet \underset{\approx}{\boldsymbol{f}''}_o \bullet (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o) + \dots \qquad (21.199)$$

where

$$
\begin{aligned}
&f_o \equiv f(\underset{\sim}{\boldsymbol{x}}_o)\\
&\underset{\sim}{\boldsymbol{f}}'_o = \frac{df(\underset{\sim}{\boldsymbol{x}})}{\partial \underset{\sim}{\boldsymbol{x}}} = \frac{\partial f}{\partial x_i}\underset{\sim}{\boldsymbol{e}}_i \text{ evaluated at } \underset{\sim}{\boldsymbol{x}} = \underset{\sim}{\boldsymbol{x}}_o\\
&\underset{\approx}{\boldsymbol{f}}''_o = \frac{d^2 f(\underset{\sim}{\boldsymbol{x}})}{d\underset{\sim}{\boldsymbol{x}}\,d\underset{\sim}{\boldsymbol{x}}} = \frac{\partial^2 f}{\partial x_i \partial x_j}\underset{\sim}{\boldsymbol{e}}_i\underset{\sim}{\boldsymbol{e}}_j \text{ evaluated at } \underset{\sim}{\boldsymbol{x}} = \underset{\sim}{\boldsymbol{x}}_o
\end{aligned}
\tag{21.200}
$$

In Eq. (21.199) the first order derivative $\underset{\sim}{\boldsymbol{f}}'_o$ is a vector and it appears in a vector inner product with $(\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o)$. The second-order derivative $\underset{\approx}{\boldsymbol{f}}''_o$ is a second-order tensor, and it is dotted from both sides by $(\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o)$, which is the same thing as taking the second-order tensor inner product

$$
\underset{\approx}{\boldsymbol{f}}'' \boldsymbol{:} [(\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o) \otimes (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o)]
\tag{21.201}
$$

Following along with this pattern, the next term in Eq. (21.199) would be

$$
\frac{1}{3!}\underset{\approx}{\boldsymbol{f}}'''_o \vdots\ [(\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o) \otimes (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o) \otimes (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o)],
\tag{21.202}
$$

where $\vdots$ would denote the third-order tensor inner product* so that the component form of the above expression would be

$$
\frac{1}{3!}\frac{\partial^3 f}{\partial x_i \partial x_j \partial x_k}\bigg|_{\underset{\sim}{\boldsymbol{x}}_o} [(x_i - x_i^o)(x_j - x_j^o)(x_k - x_k^o)]
\tag{21.203}
$$

In this case, the component form is clearer than the direct structural form, so it is a better choice in publications, especially given that the triple dot notation is highly nonstandard and might only confuse your readers. Alternatively, you can define a "*" operator to denote the inner product appropriate to the tensor order of the quantity to the right of the operator. Then the series expansion can be written

$$
f(\underset{\sim}{\boldsymbol{x}}) = \sum_{n=0}^{\infty} \frac{1}{n!}[f^n(\underset{\sim}{\boldsymbol{x}}_o)] * (\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o)^{\wedge n},
\tag{21.204}
$$

where $f^n(\underset{\sim}{\boldsymbol{x}}_o)$ is an n^{th} order tensor equal to the n^{th} derivative of f with respect to $\underset{\sim}{\boldsymbol{x}}$, evaluated at $\underset{\sim}{\boldsymbol{x}}_o$, and $(\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o)^{\wedge n}$ denotes $\underset{\sim}{\boldsymbol{x}} - \underset{\sim}{\boldsymbol{x}}_o$ multiplied dyadically by itself n times. This formula generalizes to f being an arbitrary order tensor if the derivatives are *backward* derivatives.

* Clearly, indicating the order of the inner product by the number of "stacked dots" could lead to some very weird notation for higher order tensors. You have seen the notation $\boldsymbol{::}$ used for the fourth-order inner product. A circled number can always be used for higher order inner products.

Exact differentials of one variable

Consider the expression typically given in undergraduate textbooks for the increment in work W required for a force F to act through a distance dx.

$$dW = Fdx \tag{21.205}$$

This is very unfortunate notation. Notationally, in order for this to be true, the force must be given by

$$F = \frac{dW}{dx} \tag{21.206}$$

which is not always true. The very act of writing this derivative implies that W really can be written as a proper single-valued function of x, but that's only true if the force is "conservative." For frictional forces, simply knowing where you are (i.e., knowing x) is not enough to determine the work — you have to know *how you got there*. If, for example, you know that $x=2$, then what is the work? Pushing a frictional block directly from $x=0$ to $x=2$ will require a different amount of work than pushing it from $x=0$ to $x=4$, and then finally to $x=2$. For friction, the force depends on the direction of motion, not the position of the block.

Even if you are dealing with a physical quantity that changes whenever x changes, that does not make it a *function* of x. If you want to define work to equal the integral of Fdx, then the only way you can actually evaluate the integral is if F is a function of x. In general, you will have to define work to be a path dependent quantity equal to the time integral of $F\dot{x}dt$. An example of a conservative force in one dimension is a simple spring. A linear spring is governed by $F = kx$ and therefore the work is $\frac{1}{2}kx^2$. When you stretch a spring, it takes positive work on the outbound leg (you must apply a force in the same direction as you are moving), but you gain that work back on the return trip (the force points in the opposite direction of motion — that is, the spring helps pull you back on the return trip).

The symbol dW really should be used only for *exact* differentials. The expression Fdx is not generally an exact differential. Consider any given time history, or "path" p, for x. Then the work associated with the path should be defined

$$W(t_2;p) - W(t_1;p) = \int_{t_1}^{t_2} F\dot{x}dt \tag{21.207}$$

where $\dot{x} \equiv dx/dt$. We've inserted reference to the path p to emphasize that, in general, the values you obtain for the work will depend on the nature of the path itself, not just on the endpoints. In particular, if the path for x begins and ends at the same position, the work required to move the position around this closed path might not be zero:

$$\oint Fdx \neq 0 \tag{21.208}$$

Nonzero work for a closed path will occur, for example, when you push a block a certain distance along your desk and then back again. Because your desk has friction, you will have to exert work both on the outbound path and on the return. In this case, rather than typesetting the work increment as dW, you should explicitly call out this path dependence by writing it as $đW$. The slash through the "d" tells your readers that information about the entire path (not just endpoints) would have to be provided to integrate this work increment.

For Fdx to be an exact differential, you must have

$$\oint F dx = 0 \text{ for all closed paths in } x \tag{21.209}$$

If the force satisfies this constraint, then it is called a "conservative" force field, and it is legitimate to write dW for the associated work increment. In this case we are permitted to write $dW = Fdx$. To summarize

$$F = \frac{dW}{dx} \text{ for conservative forces} \tag{21.210}$$

Exact differentials of two variables

Consider the expression

$$f(x, y)dx + g(x, y)dy \tag{21.211}$$

Is this an exact differential? In other words, does there exist a potential function $u(x, y)$ such that

$$du = f(x, y)dx + g(x, y)dy \tag{21.212}$$

If such a potential function exists, then the chain rule demands that we should be able to write

$$du = \frac{\partial u}{\partial x}dx + \frac{\partial u}{\partial y}dy \tag{21.213}$$

Comparing coefficients of dx and dy in Eqs. (21.211) and (21.213) shows that a necessary condition for existence of the potential is

$$\frac{\partial u}{\partial x} = f(x, y) \qquad \text{and} \qquad \frac{\partial u}{\partial y} = g(x, y) \tag{21.214}$$

The potential exists if this set of equations can be integrated. A necessary and sufficient condition for existence of u is

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 u}{\partial y \partial x} \tag{21.215}$$

or

$$\frac{\partial f}{\partial y} = \frac{\partial g}{\partial x} \tag{21.216}$$

If this is true, then the system in Eq. (21.214) has a solution. Otherwise, the expression $fdx + gdy$ is inexact.

The same result in a different notation. Instead of using the distinct symbols f and g, let's replace f by f_1 and g by f_2. Let's also replace x and y by x_1 and x_2, respectively. Then the above results can be written as follows:

$$f_1 dx_1 + f_2 dx_2 \tag{21.217}$$

is an exact differential if and only if

$$\frac{\partial f_1}{\partial x_2} = \frac{\partial f_2}{\partial x_1} \tag{21.218}$$

We can define a matrix $H_{ij} = \partial f_i / \partial x_j$, where the indices range from 1 to 2. Then the constraint of Eq. (21.218) requires that this matrix must be *symmetric*. If it is, then there exists a potential such that

$$du = f_1 dx_1 + f_2 dx_2 \tag{21.219}$$

The potential is found by integrating

$$f_i = \frac{\partial u}{\partial x_i} \tag{21.220}$$

This system of two equations has a solution if the matrix $H_{ij} = \partial f_i / \partial x_j$ is symmetric.

Exact differentials in three dimensions

Under what conditions does there exist a potential $u(x_1, x_2, x_3)$ such that

$$du = f_1 dx_1 + f_2 dx_2 + f_3 dx_3 \tag{21.221}$$

where the f_k are each functions of x_1, x_2, and x_3. A necessary condition for the potential to exist is

$$f_i = \frac{\partial u}{\partial x_i} \tag{21.222}$$

A sufficient condition for the existence of the potential is

$$\frac{\partial^2 u}{\partial x_i \partial x_j} = \frac{\partial^2 u}{\partial x_j \partial x_i} \tag{21.223}$$

or

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial f_j}{\partial x_i} \tag{21.224}$$

In other words, the matrix,

$$H_{ij} = \frac{\partial f_i}{\partial x_j} \tag{21.225}$$

where now the indices range from 1 to 3, must be symmetric.

Coupled inexact differentials

Consider a vector $\underset{\sim}{\boldsymbol{y}}$ that is a proper function of another vector $\underset{\sim}{\boldsymbol{x}}$. Then

$$dy_i = \frac{\partial y_i}{\partial x_j} dx_j \tag{21.226}$$

or

$$dy_i = C_{ij} dx_j \text{, where } C_{ij} = \frac{\partial y_i}{\partial x_j} \tag{21.227}$$

Note that

$$\frac{\partial^2 y_i}{\partial x_j \partial x_k} = \frac{\partial^2 y_i}{\partial x_k \partial x_j} \tag{21.228}$$

and therefore

$$\frac{\partial C_{ij}}{\partial x_k} = \frac{\partial C_{ik}}{\partial x_j} \tag{21.229}$$

Now consider the inverse question. Given a tensor C_{ij} that varies with $\underset{\sim}{\boldsymbol{x}}$, under what conditions will it be true that there exists a field $\underset{\sim}{\boldsymbol{y}}$ such that

$$C_{ij} = \frac{\partial y_i}{\partial x_j} \tag{21.230}$$

The field $\underset{\sim}{\boldsymbol{y}}$ will exist if and only if Eq. (21.229) is true.

This result has an elegant application in elasticity theory. For large deformations, the deformation gradient tensor is defined

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \tag{21.231}$$

Given only a candidate spatially varying field F_{ij}, this field corresponds to a physically realizable deformation if and only if

$$\frac{\partial F_{ij}}{\partial X_k} = \frac{\partial F_{ik}}{\partial X_j} \tag{21.232}$$

Consider, for example,

$$[F] = \begin{bmatrix} X_2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{21.233}$$

Then

$$\frac{\partial F_{11}}{\partial X_2} = k \tag{21.234}$$

but

$$\frac{\partial F_{12}}{\partial X_1} = 0 \tag{21.235}$$

Therefore, this is not a physically realizable deformation.

22. Vector/tensor Integral calculus

Gauss theorems

Stokes theorem

Divergence theorem

Integration by parts

In scalar calculus, the chain rule for differentiating a product is

$$\frac{d(uv)}{dx} = u\frac{dv}{dx} + v\frac{du}{dx} \tag{22.1}$$

Integrating both sides with respect to x varying from a to b gives

$$\int_a^b \frac{d(uv)}{dx}dx = \int_a^b u\frac{dv}{dx}dx + \int_a^b v\frac{du}{dx}dx \tag{22.2}$$

The left-hand-side is simply $uv|_a^b$. Thus, the above result can be written

$$\int_a^b u\frac{dv}{dx}dx = uv|_a^b - \int_a^b v\frac{du}{dx}dx, \tag{22.3}$$

which is the formula for integration by parts. Note that

$$uv\big|_a^b = uv\big|_b - uv\big|_a \tag{22.4}$$

We can regard the one dimensional domain to be a uniaxial bar with outward normal $n = 1$ at $x = b$ and $n = -1$ at $x = a$. Then the above expression may be written

$$uv\big|_a^b = nuv\big|_b + nuv\big|_a \tag{22.5}$$

and therefore integration by parts becomes

$$\int_a^b u\frac{dv}{dx}dx = nuv\big|_b + nuv\big|_a - \int_a^b v\frac{du}{dx}dx\,, \tag{22.6}$$

Now let’s generalize integration by parts for tensors. Suppose that, during your research, you encounter an integral of the form

$$\iiint_V \underset{\sim}{\boldsymbol{u}} \bullet \nabla \underset{\sim}{\boldsymbol{v}}\, dV \tag{22.7}$$

The integrand is a lot like the expression udv/dx just discussed in the 1D context, but now you have a *volume* integral containing a *gradient* operation. You might hope to apply one of the Gauss theorems to convert it to a surface integral. Unfortunately, however, Gauss’s theorems require that the gradient operator must act on the *entire* integrand, not just part of it. You can move partially in the desired direction by using the product rule to write the integrand differently:

$$[\underset{\sim}{\boldsymbol{u}} \bullet \nabla \underset{\sim}{\boldsymbol{v}}]_k = u_m\frac{\partial v_k}{\partial x_m} = \frac{\partial u_m v_k}{\partial x_m} - v_k\frac{\partial u_m}{\partial x_m} = [\nabla \bullet (\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}}) - \underset{\sim}{\boldsymbol{v}}(\nabla \bullet \underset{\sim}{\boldsymbol{u}})]_k \tag{22.8}$$

This equation, which is the analog of Eq. (22.1), permits the volume integral of interest to be written

$$\iiint_V \underset{\sim}{\boldsymbol{u}} \bullet \nabla \underset{\sim}{\boldsymbol{v}}\, dV = \iiint_V \nabla \bullet (\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}})dV - \iiint_V \underset{\sim}{\boldsymbol{v}}(\nabla \bullet \underset{\sim}{\boldsymbol{u}})dV \tag{22.9}$$

The first integrand on the right-hand-side is now of the form required for application of Gauss’s theorem, so it can be converted to a surface integral:

$$\iiint_V \underset{\sim}{\boldsymbol{u}} \bullet \nabla \underset{\sim}{\boldsymbol{v}}\, dV = \iint_S \underset{\sim}{\boldsymbol{n}} \bullet (\underset{\sim}{\boldsymbol{u}}\underset{\sim}{\boldsymbol{v}})dS - \iiint_V \underset{\sim}{\boldsymbol{v}}(\nabla \bullet \underset{\sim}{\boldsymbol{u}})dV \tag{22.10}$$

This is the generalization of integration by parts. Note the similarity of this result with the 1D formula in Eq. (22.6). This basic process may be applied to integrands of various structures. Specifically, if your integrand involves a gradient, you can apply the product rule to make the gradient act across the entire integrand, subtracting appropriate terms. Then a Gauss theorem applies to the gradient term, converting it to a surface integral.

Leibniz* theorem

The Leibniz formula,

$$\frac{d}{dt}\int_{a(t)}^{b(t)} f(x,t)dx = \int_{a(t)}^{b(t)} \frac{\partial f(x,t)}{\partial t}dx + f(b(t),t)\frac{db(t)}{dt} - f(a(t),t)\frac{da(t)}{dt} \tag{22.11}$$

is typically given lower-class status in applied math textbooks. It appears only as a footnote in Ref. [36] and as part of a homework question in Ref. [1]. Though elementary, this is a *profoundly* useful formula in its own right and especially in its generalization to higher dimensions.

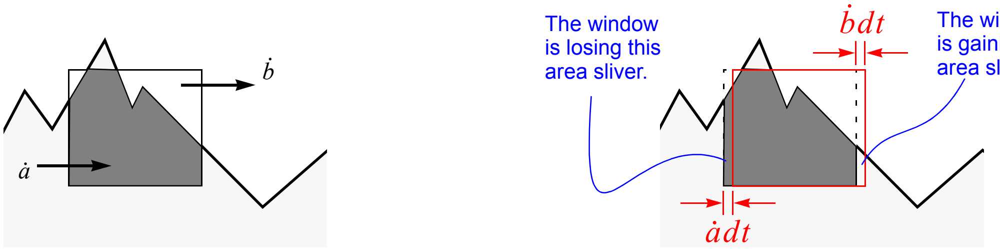
Let's first understand the Leibniz formula in the 1D form cited in Eq. (22.11). What is this equation trying to say? Why should we care? Looking at the left-hand-side of Eq. (22.11) shows that this is a formula for taking the time derivative of an integral where both the integrand *and the integration limits* vary with time. The variable t doesn't have to be *time* — it can be any scalar, but we will discuss it using the name "time" for t and the name "position" for x. When written with the arguments of the functions being *understood*, and letting total differentiation by time be denoted with a superimposed "dot", Eq. (22.11) takes a cleaner look:

$$\frac{d}{dt}\int_a^b f dx = \int_a^b \frac{\partial f}{\partial t}dx + f_b\dot{b} - f_a\dot{a} \tag{22.12}$$

Here, f_a and f_b denote the function evaluated at $x=a$ and $x=b$, respectively. The symbols $\dot{a}$ and $\dot{b}$ represent the velocities of the boundary. The integral on the left-hand side represents the area under the curve $f(x)$ at some particular moment in time. The right-hand-side tells us that this area can change in two possible ways: you can change the function itself (so $f(x)$ is really $f(x,t)$) or you can change the limits of integration. This equation tells us you can't simply bring the total time derivative inside the integral — movement of the boundary must be accounted for. Also note that a total derivative "d" is used on the left-hand side, but a partial derivative is used once the derivative is "passed inside" the inte-

* Note: this word is *not* spelled "Leibnitz". Even though there's no "T", it should be *pronounced* as if it were spelled with one. Reference [36], which is written for engineers, gets it wrong but Reference [1], for physicists, gets it right. It's another embarrassing day for us engineers.

gral. That's because the function f depends on both x and t, but the *area* depends only on t. Now, what about those boundary terms? Why is there a negative sign in front of the term for the left boundary, but no negative for right boundary? A picture tells it all. Suppose you are looking at the profile of a mountain through the window of an airplane:


$\dot b$
$\dot a$
The window is losing this area sliver.
$\dot b dt$
The window is gaining this area sliver.
$\dot a dt$

Only a portion of the mountain shows through your window, and you naturally observe that the visible area changes with time. The mountain itself isn't (we hope) moving, so $\partial f / \partial t$. The visible area is changing simply because the boundary is moving. In this example, the leading edge is moving at the same speed as the trailing edge. From the picture, it's clear that you are *gaining* area on the leading edge (new area is "flowing" into the window). The incremental change in area equals the height of the function there, f_b, times the width $\dot b dt$. You are *losing* area (negative area rate) on the trailing edge with an area increment $f_a \dot a dt$.* In the limit, the rate of area is determined by the *flux* of the function across the boundary. Of course, if the mountain were simultaneously sinking into the ocean, then you would see area changes from *interior* contributions to the area integral captured through the $\partial f / \partial t$ term.

Another way to deal with the sign of the flux (and a key to generalizing Leibniz's formula to higher dimensions) is to introduce the *outward-pointing* normal to the boundary:

$$n_a = -1 = \text{outward unit normal at } x{=}a \tag{22.13}$$

$$n_b = +1 = \text{outward unit normal at } x{=}b \tag{22.14}$$

Then the 1D Leibniz formula would read

$$\frac{d}{dt}\int_a^b f dx = \int_a^b \frac{\partial f}{\partial t} dx + f_b n_b \dot b + f_a n_a \dot a \tag{22.15}$$

In this form, there is no negative. Each boundary term is the value of the function at that boundary location times the product of the boundary velocity times the outward normal.

The generalization of the Leibniz formula to 3D is

$$\boxed{\frac{d}{dt}\int_{\Omega(t)} f(\underset{\sim}{x}, t) dV = \int_{\Omega(t)} \frac{\partial f(\underset{\sim}{x}, t)}{\partial t} dV + \int_{\partial\Omega(t)} f(\underset{\sim}{x}, t)(\underset{\sim}{n} \bullet \underset{\sim}{v}_b) dS} \tag{22.16}$$

* If the airplane were ascending or descending, you would have additional changes in area from the bottom and top edges of the window frame, but that would make this a two-dimensional problem, so we will assume the plane is cruising at a constant altitude.

Note the similarity between this equation and its 1D counterpart, Eq. (22.15). Here, Ω is a volume whose shape is varying in time. The integral over dV is a volume integral and the integral over dS is a surface integral over the time-varying boundary $\partial\Omega$. The vector $\underset{\sim}{v}_b$ is the velocity of the boundary and the integrand of the surface integral is the *net* flux of the function f into the integral resulting from motion of the boundary. Of course outflow occurs if the boundary velocity is encroaching on the interior of Ω, which would be accounted for automatically because the boundary velocity $\underset{\sim}{v}_b$ would have a *negative* dot product with the *outward* unit normal $\underset{\sim}{n}$. Tangential components of the boundary velocity have no influence on the rate of the integral.

For the generalized Leibniz formula to make sense, you must do a good job of specifying the velocity of the boundary everywhere. Continuum mechanics (and thermodynamics) theory typically deals with two types of "control volumes." An *Eulerian* control volume is fixed in space so that $\underset{\sim}{v}_b = \underset{\sim}{0}$ and

$$\frac{d}{dt}\int_{\Omega} f(\underset{\sim}{x}, t)\,dV = \int_{\Omega}\frac{\partial f(\underset{\sim}{x}, t)}{\partial t}\,dV \quad \text{for Eulerian control volumes} \tag{22.17}$$

Because an Eulerian control volume is fixed in space, material (and any properties it might carry such as density and polarization) will flux through the boundary (note that we changed $\Omega(t)$ to simply Ω). A *Lagrangean** control volume, on the other hand, is defined to always contain the same material particles, so its velocity equals the material velocity, $\underset{\sim}{v}_b=\underset{\sim}{v}$, and therefore the 3D Leibniz formula for this type of control volume is

$$\frac{d}{dt}\int_{\Omega(t)} f(\underset{\sim}{x}, t)\,dV = \int_{\Omega(t)}\frac{\partial f(\underset{\sim}{x}, t)}{\partial t}\,dV + \int_{\partial\Omega(t)} f(\underset{\sim}{x}, t)(\underset{\sim}{n} \bullet \underset{\sim}{v})\,dS \quad \text{for Lagrangean control volumes} \tag{22.18}$$

This form of the Leibniz formula is called Reynolds† transport theorem. The only difference between Eqs. (22.17) and (22.18) is the motion of the boundary, which shows just how crucial it is for you to be extremely diligent to be clear about which viewpoint you are using. Many theorists prefer to apply *all* of their integral equations to Lagrangean control volumes — this makes a lot of sense because the most fundamental statements of physical principals are often the most elegant when applied to a specific set of particles. For example, Lagrangean conservation of mass states that the total mass of a specific set of particles must not change‡ and Newtonian mechanics tells us that the net force on a specific set of particles must equal the rate of change of the momentum. People who tend to use Lagrangean control volumes can spare themselves the trouble of constantly reminding their readers that their control volumes move with the material by using the material deriv-

* The spelling of this word is yet another subject of debate among the cognoscenti. "Lagrangean" appears better justified from the historical record and from rules of orthography, but so many people use "Lagrangian" that this might be a losing battle.

† Here we go again: It's Reynolds, not Reynold's (each undergraduate's bane).

‡ Before the nuclear engineers out there go ballistic and remind us that $E = mc^2$ tells us that mass can be annihilated by converting it to energy, just keep in mind that conservation of mass just a principle, not a law.

ative symbol, $D(\ )/Dt$, to tell their readers that this is a time derivative in which the material particles are being followed. When using this notation, showing the explicit time dependence in $\Omega(t)$ is not necessary because using "D" instead of "d" implies the time dependence. Thus, you might see Reynolds transport written as

$$\frac{D}{Dt}\int_{\Omega} f(\underset{\sim}{x}, t)dV = \int_{\Omega}\frac{\partial f(\underset{\sim}{x}, t)}{\partial t}dV + \int_{\partial\Omega} f(\underset{\sim}{x}, t)(\underset{\sim}{n} \bullet \underset{\sim}{v})dS \tag{22.19}$$

The phrase "for Lagrangean control volumes" is no longer needed because it's implied by the use of "D."

LONG EXAMPLE: conservation of mass. In Eulerian form, the principle of conservation of mass says, in essence,

What goes in must come out — or stay there. (22.20)

The first part (what goes in must come out) is referring to flux of mass across the fixed Eulerian boundary. The second part is referring to the fact that mass can simply accumulate inside the control volume (neither coming in nor going out). In rate form, the Eulerian statement of conservation of mass is

The rate of accumulation must equal the rate of influx (22.21)

or

The rate of accumulation minus the rate of influx must equal zero (22.22)

Letting stuff going in be regarded as a negative contribution to stuff coming out, the statement could alternatively read

The rate of accumulation plus the rate of OUTflow must equal zero (22.23)

Since mass equals density ρ times volume, this Eulerian statement of mass conservation may be written mathematically as

$$\frac{d}{dt}\int_{\Omega}\rho dV + \int_{\partial\Omega}\rho(\underset{\sim}{n} \bullet \underset{\sim}{v})dS = 0 \quad \text{for Eulerian control volumes} \tag{22.24}$$

The first term measures the rate of mass accumulation and the second term measures the rate of outflow. Since this is an Eulerian control volume, you can use Eq. (22.17) "inside" the integral to write

$$\int_{\Omega}\frac{\partial\rho}{\partial t}dV + \int_{\partial\Omega}\rho(\underset{\sim}{n} \bullet \underset{\sim}{v})dS = 0 \tag{22.25}$$

Now consider how the analysis of conservation of mass would go from a Lagrangean perspective. If we are considering a fixed set of particles, conservation of mass merely says that the total mass of those particles must not change. In rate form, that means

$$\frac{D}{Dt}\int_{\Omega}\rho dV = 0 \tag{22.26}$$

or, using Eq. (22.19),

$$\int_{\Omega}\frac{\partial\rho}{\partial t}dV + \int_{\partial\Omega}\rho(\underset{\sim}{n} \bullet \underset{\sim}{v})dS = 0 \tag{22.27}$$

Note that this Lagrangean result is identical to Eq. (22.25) which was derived using an Eulerian perspective. Importantly, once a derivative has been brought inside of an integral, the nature of the boundary (Lagrangean, Eulerian, or otherwise) becomes inconsequential. Now that the derivative is inside the spatial integral, both equations give the same result if evaluated over the same region in space. The fact that those regions might no longer coincide at some time in the future has no bearing on what's going on now.

Incidentally, by using the divergence theorem, the surface integral may be converted to a volume integral. To do this, keep in mind that the unit normal becomes a "nabla" gradient operator that acts on *everything else* in the integrand. Thus, $\rho(\underset{\sim}{\boldsymbol{n}} \bullet \underset{\sim}{\boldsymbol{v}})$ will *not* become $\rho\vec{\nabla} \bullet \underset{\sim}{\boldsymbol{v}}$. Instead, it will become $\vec{\nabla} \bullet (\rho\underset{\sim}{\boldsymbol{v}})$. Of course, don't forget to change the $\partial\Omega$ to an Ω and the dS to a dV to obtain

$$\int_{\Omega} \frac{\partial\rho}{\partial t} dV + \int_{\Omega} \vec{\nabla} \bullet (\rho\underset{\sim}{\boldsymbol{v}}) dV = 0 \tag{22.28}$$

or

$$\int_{\Omega} \left(\frac{\partial\rho}{\partial t} + \vec{\nabla} \bullet (\rho\underset{\sim}{\boldsymbol{v}}) \right) dV = 0 \tag{22.29}$$

The differential form of conservation of mass (often called the continuity equation) is obtained by asserting that this integral equation must be true regardless of what region of space we choose to use as a control volume. The only way that can be true is if the integrand itself is zero:

$$\frac{\partial\rho}{\partial t} + \vec{\nabla} \bullet (\rho\underset{\sim}{\boldsymbol{v}}) = 0 \tag{22.30}$$

In this equation, the time derivative holds the spatial position $\underset{\sim}{\boldsymbol{x}}$ fixed and the gradient is a derivative with respect to $\underset{\sim}{\boldsymbol{x}}$ holding time constant. To make this point more explicit, you might see the partial derivative being written more carefully with a subscript that describes precisely what is being held fixed:

$$\boxed{\left(\frac{\partial\rho}{\partial t}\right)_{\underset{\sim}{\boldsymbol{x}}} + \vec{\nabla} \bullet (\rho\underset{\sim}{\boldsymbol{v}}) = 0} \tag{22.31}$$

The fact that the time derivative holds the spatial position fixed makes this form of conservation of mass particularly appealing to people with an Eulerian mindset. An Eulerian observer plants his you-know-what in a single location and watches how things change in front of him. He will be looking at the densities of various particles at the instant they pass by his fixed location. This perspective is unsavory to folks with a Lagrangean mind-set.

Lagrangean people prefer to talk about how properties of a *single-particle* of interest change over time. Of course, to do that, they will have to obligingly move themselves with the particle, keeping it always under their watchful eye. To describe the Lagrangean time derivative, we need to introduce a variable $\underset{\sim}{\boldsymbol{X}}$ whose value will never change for any given particle. Furthermore, distinct particles need to have distinct values of $\underset{\sim}{\boldsymbol{X}}$ so that $\underset{\sim}{\boldsymbol{X}}$ can, in a sense, be regarded as a unique identifier or "name" for any particle in much the same way that $\underset{\sim}{x}$ uniquely defines a particular location in space. A natural choice for $\underset{\sim}{X}$ is simply the *initial* location of a particle. The uniqueness requirement is satisfied because no two particles will have occupied the same initial location in space.* The requirement that $\underset{\sim}{X}$ not vary in time is satisfied because, naturally, the *initial* location of a particle is time-independent. Now we are in a position to define the **substantial derivative** or **material derivative** of any field variable Φ as

$$\frac{D\Phi}{Dt} \equiv \left(\frac{\partial\Phi}{\partial t}\right)_{\underset{\sim}{X}} \qquad (22.32)$$

Since $\underset{\sim}{X}$ is being held constant, and since $\underset{\sim}{X}$ uniquely identifies a *particular particle* this expression is the time derivative of Φ as seen by an observer who is moving with the particle — it is the Lagrangean time derivative!

What is the connection between what a Lagrangean and Eulerian observers see? The answer comes from the chain rule:

$$\left(\frac{\partial\Phi}{\partial t}\right)_{\underset{\sim}{X}} = \left(\frac{\partial\Phi}{\partial t}\right)_{\underset{\sim}{x}} + \left(\frac{\partial\Phi}{\partial \underset{\sim}{x}}\right)_{t} \bullet \left(\frac{\partial \underset{\sim}{x}}{\partial t}\right)_{\underset{\sim}{X}} \qquad (22.33)$$

In the last term, $(\partial\underset{\sim}{x}/\partial t)_{\underset{\sim}{X}}$ holds $\underset{\sim}{X}$ constant, so it is the rate at which the Lagrangian observer sees the spatial locations change. You might be wrongly tempted to think that, if the observer is moving to the right, then he sees the material beneath him moving to the left, but you have to interpret the derivative $(\partial\underset{\sim}{x}/\partial t)_{\underset{\sim}{X}}$ literally. If for example, the Lagrangean observer moves from $x = 0$ to $x = 4$ over some period of time, he sees that x has *increased*. Thus, even though the spatial grid appears to be moving to his left, the *values* on the grid are increasing. The rate of increase is exactly equal to the velocity of the observer. Since Lagrangean observers move with the material, this means that

$$\left(\frac{\partial \underset{\sim}{x}}{\partial t}\right)_{\underset{\sim}{X}} = \underset{\sim}{v}\text{, the material velocity!} \qquad (22.34)$$

Thus, the material derivative becomes

$$\left(\frac{\partial\Phi}{\partial t}\right)_{\underset{\sim}{X}} = \left(\frac{\partial\Phi}{\partial t}\right)_{\underset{\sim}{x}} + \left(\frac{\partial\Phi}{\partial \underset{\sim}{x}}\right)_{t} \bullet \underset{\sim}{v} \qquad (22.35)$$

You will frequently see this result presented more cryptically as

$$\frac{D\Phi}{Dt} = \frac{\partial\Phi}{\partial t} + (\nabla\Phi) \bullet \underset{\sim}{v} \qquad (22.36)$$

* Actually, this is not quite true from a mathematical perspective — if a material fractures, then a particle that started out at a particular location will "break" into separate locations. There are ways around this issues. For the discussion at hand, though, let's assume no fracture.

and it will be up to you to keep in mind that *every term* involves partial derivatives, which can make physical sense only if you vigilantly remind yourself what is being held constant in those derivatives. Using this result in Eq. (22.30) permits us to now write the continuity equation as

$$\frac{D\rho}{Dt} - (\nabla\rho) \bullet \underset{\sim}{\boldsymbol{v}} + \vec{\nabla} \bullet (\rho\underset{\sim}{\boldsymbol{v}}) = 0 \tag{22.37}$$

Using the product rule on the last term leads to a cancellation of the middle term, giving

$$\boxed{\frac{D\rho}{Dt} + \rho(\vec{\nabla} \bullet \underset{\sim}{\boldsymbol{v}}) = 0} \tag{22.38}$$

Contrast this with Eq. (22.31). They are both the *same* result, with one presented using the Eulerian time derivative and the other using the Lagrangean material derivative. Note also that one equation has the density field inside the divergence operator, while the other doesn't.

Yet another notational convenience is in common use: for any field variable Φ,

$$\dot{\Phi} \quad \text{means the same thing as} \quad \frac{D\Phi}{Dt} \tag{22.39}$$

Using this convention, Eq. (22.38) can be written

$$\frac{\dot{\rho}}{\rho} = \vec{\nabla} \bullet \underset{\sim}{\boldsymbol{v}} \tag{22.40}$$

For continuum mechanics researchers, this is a very appealing result because numerical material models typically have access to the **velocity gradient tensor**, typically denoted by $\underset{\approx}{\boldsymbol{L}}$, and defined

$$\underset{\approx}{\boldsymbol{L}} \equiv \underset{\sim}{\boldsymbol{v}}\overleftarrow{\nabla}\text{, or } L_{ij} = \frac{\partial v_i}{\partial x_j} \tag{22.41}$$

Note that

$$\mathrm{tr}\underset{\approx}{\boldsymbol{L}} = \underset{\sim}{\boldsymbol{v}} \bullet \overleftarrow{\nabla} = \vec{\nabla} \bullet \underset{\sim}{\boldsymbol{v}} \tag{22.42}$$

and therefore, the continuity equation can be written in yet *another* form

$$\frac{\dot{\rho}}{\rho} = \mathrm{tr}\underset{\approx}{\boldsymbol{L}} \tag{22.43}$$

Finally, since the trace operation depends only on the diagonal components, this result is often written

$$\frac{\dot{\rho}}{\rho} = \mathrm{tr}\underset{\approx}{\boldsymbol{D}}, \qquad \text{where} \qquad \underset{\approx}{\boldsymbol{D}} \equiv \frac{1}{2}(\underset{\approx}{\boldsymbol{L}} + \underset{\approx}{\boldsymbol{L}}^T) = \mathrm{sym}(\underset{\approx}{\boldsymbol{L}}) \tag{22.44}$$

The tensor $\underset{\approx}{\boldsymbol{D}}$ is often called the "rate" of deformation even though it is not a true rate.

Generalized integral formulas for discontinuous integrands

All of the integral formulas cited so far require the integrand to be differentiable. What can you do if the integrand is differentiable everywhere except across a singular “jump” surface, like what you have in shock physics? The answer is to first break up the surface integrals into two separate parts over each differentiable domain. However, doing this will break a formerly closed surface into the union of two open surfaces. Before you can apply the divergence theorem, you will need to close the surfaces, this time along the shared boundary defining the discontinuity. This is the basic flavor of dealing with discontinuous integrands. Now let’s get into the details.

"The ability to quote is a serviceable substitute for wit." — *W. Somerset Maugham*

23. Closing remarks

This book has provided an introduction to vector and tensor concepts, as well as some very advanced forays into more obscure topics such as projections (in ordinary engineering space, as well as higher dimensional spaces), the concept that a tensor is also a vector, the notion of right and left gradient operations, and generalizations of hallowed integral theorems permitting application to discontinuous integrands (as in shock physics).

A key goal of this book was to call out a distinction between the functional, engineering, and structural perspectives on tensor analysis. You can't have one without the other. The functional meaning of a tensor is a linear transformation from vectors to vectors. This statement does not make sense unless the term "vector" is defined and, for engineering applications, this necessarily requires discussing of how vector components transform upon a change in basis. Finally, the *structural* perspective of tensor analysis deals exclusively for the arrangement of symbols that we write down on a page to *represent* the functional meaning of a tensor. The structures themselves are often referred to as tensors, and there is nothing wrong with that. In this book, we outlined a structural self-consistent tensor nomenclature that is well-poised for moving smoothly into higher dimensions, but we are not intending to suggest that our structure is "the best." There is no best tensor structure. Granted there are some bad ones out there that aren't self-consistent, but lots of folks use alternative notations that are adequate and often superior for their applications.

This book is by no means a comprehensive discourse. Our goal here was to emphasize theorems and concepts that are useful in mechanics, especially in the higher dimensional applications that come up in materials modeling. We have omitted many classic discussions (such as formulas for gradient operations in curvilinear systems) that are very standardized and available in almost any textbook on vector and tensor analysis. We only mentioned classic operations and concepts when (a) they were needed for completeness of later, less classic, discussions that we will publish elsewhere or (b) we had non-traditional insights to offer.

This book should be regarded as a "zeroth" or "beta" edition. As such it is highly likely that it still contains errors and typos. Readers are encouraged to send feedback and corrections to ~~rmbrann@me.unm.edu~~.

24. Solved problems

This chapter is brand-new and totally disorganized. It will take a few more seasons to get it up to speed. Sorry.

REFERENCES

[1]Arfken, George B. and Weber, Hans, J., **Mathematical Methods for Physicists**, Academic Press, San Diego (1995).

[2]Aris, R., **Vectors, Tensors, and the Basic Equations of Fluid Mechanics**. Englewood Cliffs, N.J.: Prentice-Hall, 1962.

[3]Beyer, William H., **The CRC handbook of mathematical sciences, 6th Edition.**, CRC Press, Boca Raton, FL, 2000.

[4]Bird, R.B., Armstrong, R.C., and Hassager, O., **Dynamics of Polymeric Liquids, Fluid Mechanics**, 2nd Ed, Wiley-Interscience, 1987.

[5]Budiansky, Bernard, (1990) *Tensors*, in: **Handbook of Applied Mathematics 2nd Ed**, Edited by Carl E. Pearson. Van Nostrand Reinhold.

[6]Brannon, R. M., `http://me.unm.edu/~rmbrann/curvilinear.pdf`

[7]Brannon, R. M., `http://me.unm.edu/~rmbrann/Rotation.pdf`

[8]Brannon, R. M., `http://me.unm.edu/~rmbrann/RadialReturn.pdf`

[9]Brannon, R. M., `http://me.unm.edu/~rmbrann/MohrsCircle.pdf`

[10]Brannon, Rebecca M. and Drugan, Walter J., *Influence of nonclassical elastic-plastic features on shock wave existence and spectral solutions*. J. Mech. Phys. Solids, **41** (2), pp. 297-330.

[11]Brannon, Rebecca M. *Caveats concerning conjugate stress and strain measures for anisotropic elasticity.* Acta Mechanica, Accepted 1997.

[12]Stakgold, Ivar, **Green's Functions and Boundary Value Problems**, Wiley (1979).

[13]Malvern, Lawrence E., **Introduction to the Mechanics of a Continuous Medium**, Prentice-Hall (1969).
This is one of the best known references on elementary continuum mechanics.

[14]Smart, W.M., **Textbook on Spherical Astronomy**. 6th Ed. revised by R. M. Green., Cambridge University Press, 1977.
Contains theorems that are useful in generating the mesh of the sphere.

[15]Smith, G.F., Smith, M.M., and Rivlin, R.S., *Integrity bases for a symmetric tensor and a vector — The crystal classes.* Arch. Rational Mech. Anal., Vol 12 (1963).
The title says it all. This paper shows the most general quadratic form that can be constructed from the isotropic invariants for a crystal class.

[16]Spencer, A.J.M., *Theory of Invariants*, in **Continuum Physics I**, edited by A.C. Eringen Academic, New York (1971).
Derives the integrity bases for arbitrary numbers of vectors and tensors for (i) full orthogonal group (ii) proper orthogonal (rotation) group, (iii) transverse isotropy, and (iv) crystal classes.

[17]Varadan, Vasundara V.; Jeng, Jiann-Hwa; and Varadan, Vijay K., *Form invariant constitutive relations for transversely isotropic piezoelectric materials.* J Acoust. Soc. Am. **82** (1) pp. 337-342 (1987).
This paper shows how to derive transversely isotropic constitutive relations using integrity basis for the transverse orthogonal group of transformations.

[18]Lai, W. M., Krempl, E., and Rubin, D., **Introduction to Continuum Mechanics, 3rd Ed.** Butterworth-Heinemann, 1995.

This book seems well received by new students to the subject. However, the reader should be cautioned that this book is riddled with errors and it uses somewhat old-fashioned notation.

[19]Tai, Chen-To (1997) **Generalized Vector and Dyadic Analysis**. Applied Mathematics in Field Theory, 2nd Ed. IEEE Press, NY.

[20]Simmonds, James G. **A Brief on Tensor Analysis**, Springer-Verlag, NY (1994).

[21]Chadwick, Peter, **Continuum mechanics, Concise theory and applications, 2nd Ed.,** Dover, Reprinted 1999.

[22]Stewart, G.W., **Introduction to Matrix Computations**, in *Computer Science and Applied Mathematics series of monographs and textbooks.* Academic Press (1973).

[23]Greenberg, Michael, **Advanced Engineering Mathematics (2nd Edition)**, Prentice Hall, 1998.

[24]Gurtin, Morton E., **An introduction to Continuum Mechanics**, Academic Press, New York, 1981.

[25]Higham, N.J., *Computing the Polar Decomposition — with applications.* SIAM J. Sci. Stat. Comput. 7, p. 1160-1174 (1986).

[26]Horn, Roger A., and Johnson, Charles R., **Matrix Analysis**, Cambridge University Press, Melbourne Australia (1985).

[27]Wolfram, Stephen, **The Mathematica Book, 4th Ed**, Cambridge Univ. Press, 1999.

[28]Friedman, Avner, **Foundations of Modern Analysis**, Dover (1970).

[29]Rudin, Walter, **Real and Complex Analysis**, McGraw-Hill (1987).

[30]Ramkrishna, Doraiswami, and Amundson, Neal R., **Linear Operator Methods in Chemical Engineering**, Prentice-Hall (1985).

[31]Folland, Gerald B., **Real Analysis**, Wiley-Interscience (1984).

[32]Buck, R. Creighton, **Advanced Calculus**, McGraw-Hill (1978).

[33]Boccara, Nino, **Functional Analysis: An introduction for Physicists**, Academic Press (1990).

[34]Noble, Ben, and Daniel, James W., **Applied Linear Algebra, 3rd Ed.**, Prentice-Hall, 1988.

[35]Pearson, Carl (editor), **Handbook of Applied Mathematics, 2nd Ed.**, Van Nostrand Reinhold, 1990.
This book uses the same "no symbol" notation for dyadic multiplication that we espouse. This book also has good succinct chapters on matrix and vector analysis.

[36]Wylie, C. Ray, and Barrett, Louis C., **Advanced Engineering Mathematics**, McGraw-Hill (1982).

INDEX

This index is a work in progress. Please notify the author of any critical omissions or errors.

Interpreting this index

Whenever a topic has an index entry labeled "Example," you will locate usage or application of that topic on the cited page. ExampleE, ExampleM and ExampleT indicate examples that range in difficulty from **E**asy to **M**edium to **T**oilsome, respectively. Examples marked with an exclamation point are of particular interest to the mechanics and materials modeling community.

More examples and solved problems are to come in future editions.