Geometry for Mathematical Physics

Volume I: Operational Toolbox

Linear Algebra, Multivariable Calculus, Tensors, and Forms

A working note for building a usable foundation toward modern geometry (and eventually higher geometry) in mathematical physics.

December 23, 2025

Preface

This volume is the first part of a long-term note project aimed at learning *geometry for mathematical physics* in a way that is both rigorous and *operational*. The guiding principle is simple:

If you cannot compute with it, you do not yet own it.

Many foundational results in geometry and analysis have deep proofs that belong to later volumes (or specialized texts). In this volume, whenever a proof is too heavy for the current stage, we adopt a *black-box but operational* approach: we state the theorem precisely (including its hypotheses), learn how to apply it in concrete computations, and record typical failure modes. Full proofs and conceptual expansions will appear in later volumes.

Goals of Volume I. After finishing this volume, you should be able to:

1. **Differentiate and compute reliably in $\mathbb{R}^n$:** compute Fréchet derivatives, Jacobians, Hessians, and use the multivariable chain rule in practice.

2. **Use the inverse/implicit function theorems as tools:** check hypotheses quickly and compute the resulting local formulae (e.g. the derivative of an implicitly defined function).

3. **Work fluently with tensors and indices:** manipulate multilinear maps, tensor products, contractions, and change-of-basis transformation rules (including the Einstein summation convention).

4. **Compute with alternating tensors and basic differential forms:** perform wedge-product computations (with correct signs) and connect classical vector calculus identities to the language of forms at an operational level.

5. **Solve and interpret basic linear ODE systems:** compute e^{tA} in standard cases and understand linear flows as the prototype for vector-field flows on manifolds.

6. **Recognize variational structures in simple settings:** derive Euler–Lagrange equations in basic examples and understand the "energy functional $\to$ PDE/ODE" pipeline.

What you will be able to do next. This volume is designed to make the transition to smooth manifolds and differential geometry (tangent bundles, differential forms, integration on manifolds, connections and curvature) *non-mysterious*. In particular, the computational habits built here will be reused constantly in: Riemannian geometry, gauge theory, symplectic/contact geometry, Hodge theory, and later, in the language of descent, stacks, and derived/higher geometry.

Exercises as training. Each chapter will eventually come with exercises of four types:

- **D (Definition/Notation)**: definitions, hypotheses, and symbol control;
- **C (Computation)**: explicit calculations and worked examples;

- **P (Proof)**: short lemmas and correctness checks;
- **T (Translation)**: rewriting the same content in another language (e.g. matrices $\leftrightarrow$ bilinear forms; vector calculus $\leftrightarrow$ differential forms).

The long-term goal is not to "read many pages" but to build a toolbox that can be used immediately in later volumes and in research-level study.

Note. This front skeleton includes only the cover, preface, and table of contents. Chapter bodies will be filled progressively.

Contents

Chapter 1

Conventions, Notation, and a Symbol Dictionary (Volume I)

1.1 Purpose of this chapter

This chapter is for the author and for future readers of these notes. Its goals are:

- to fix *volume-level* conventions (what objects symbols usually denote, default base field, default regularity, index ranges);
- to impose a disciplined policy for *symbol reuse* and *controlled abuse of notation*;
- to provide a *symbol dictionary* that can be referenced throughout Volume I.

The guiding principle is: *a symbol must have a type and a scope.* When a symbol is reused, the reuse must respect its role ("namespace").

1.2 Typing discipline and scopes

1.2.1 Every symbol has a type

In these notes, a symbol is never "just a letter". It comes with a type such as:

set, vector space, linear map, matrix, function, 1-form, operator, parameter, index, ...

If the type is non-obvious from context, it must be stated explicitly.

1.2.2 Global vs. local symbols

We distinguish two scopes.

- **Global symbols (volume-level).** Fixed meaning throughout Volume I and *never* redefined.
- **Local symbols (chapter/section-level).** Introduced and used only within a declared scope; once the scope ends, the symbol is considered "released".

1.2.3 Local notation protocol (mandatory)

At the beginning of every chapter/section that introduces new notation, include a short list:

- **(New)** symbols introduced in this section, with types;

- **(Inherited)** symbols used with their global meaning;
- **(Warnings)** any controlled abuses of notation used in this section.

This prevents silent shifts in meaning.

1.3 Equality, isomorphism, and controlled identifications

1.3.1 Four common "equality" relations

We use the following conventions:

- $=$ denotes literal equality (same object, same element).
- $:=$ and $=:$ denote definition (introducing a name).
- $\cong$ denotes isomorphism (there exists an isomorphism, possibly non-canonical).
- $\simeq$ denotes a specified/understood equivalence (often canonical, or fixed once and for all).

1.3.2 Policy for "abuse of notation"

Abuse of notation is permitted only if:

1. the identification is invariant under a stated equivalence relation (e.g. up to isomorphism, or modulo an explicitly defined quotient);
2. the abuse is announced once, via a sentence of the form:

$$\text{"We identify } A \text{ with } B \quad \text{via } \phi : A \to B.\text{"}$$

Unannounced abuse is treated as an error in these notes.

1.4 Namespaces: letter pools and font conventions

To keep roles separated, we adopt the following *letter pools*. A symbol may be reused *only* inside its pool unless explicitly promoted to the global dictionary.

1.4.1 Objects and spaces

- Sets/spaces: X, Y, Z (generic ambient spaces).
- Vector spaces: V, W, U (finite-dimensional unless stated).
- Euclidean spaces: $\mathbb{R}^n$ with coordinates $x = (x^1, \ldots, x^n)$.

1.4.2 Maps and operators

- General maps/functions: f, g, h.
- Linear maps/operators: T, S, L.
- Matrices: A, B, M.

1.4.3 Indices and parameters

- Indices: i, j, k, ℓ (range declared when used).
- Dimensions: $m, n \in \mathbb{N}$ (typically $n = \dim V$).
- Small parameters: $\varepsilon, \delta > 0$.

1.4.4 Fonts

- Blackboard bold: $\mathbb{R}, \mathbb{C}, \mathbb{N}, \mathbb{Z}$.
- Calligraphic/script: $\mathcal{U}, \mathcal{V}$ for families/collections; F for function spaces when needed.
- Boldface vectors are avoided unless clarity requires it; we prefer $x \in \mathbb{R}^n$ and component notation x^i.

1.5 Volume I global conventions

1.5.1 Default base field and dimension

Unless explicitly stated:

- all vector spaces are over $\mathbb{R}$;
- all vector spaces are finite-dimensional;
- $\dim V = n$ when V is the main space in a discussion.

1.5.2 Default regularity of maps

When working on open sets $U \subseteq \mathbb{R}^n$, functions $f : U \to \mathbb{R}^m$ are assumed C^1 when differentiation is performed. Higher regularity is stated as needed.

1.5.3 Index ranges and summation

- Indices i, j, k, ℓ typically range over $\{1, \ldots, n\}$ (or $\{1, \ldots, m\}$ depending on context); the range must be declared at first use if ambiguous.
- Einstein summation convention is *not* used by default in early chapters. It may be enabled explicitly in the tensor/index chapter (later in this volume), and then the convention will be stated locally.

1.5.4 Differential notation: D, ∇, d

To avoid a common source of confusion, we reserve:

- $Df(x)$ for the Fréchet derivative (Jacobian as a linear map);
- ∇f for the gradient (when an inner product/Euclidean structure is present);
- df for the differential of a scalar function (a 1-form);
- d for the exterior derivative on differential forms (once forms are introduced).

Partial derivatives are denoted by $\partial_i f := \frac{\partial f}{\partial x^i}$.

1.6 Symbol dictionary (Global ledger for Volume I)

The following table lists the principal global symbols used throughout Volume I. Symbols not listed here are *local* by default and must be declared in their scope.

Symbol	**Type**	**Meaning / convention**
$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	sets	standard number systems
$x = (x^1, \ldots, x^n)$	vector/coords	coordinates on $\mathbb{R}^n$
$U \subseteq \mathbb{R}^n$	open set	domain for multivariable calculus
V, W, U	vector spaces	real, finite-dimensional unless stated
V^*	vector space	dual space $\mathrm{Hom}(V, \mathbb{R})$
$\langle \varphi, v \rangle$	pairing	evaluation $\varphi(v)$ for $\varphi \in V^*, v \in V$
$\langle v, w \rangle$	bilinear form	inner product when specified; induces $\|v\| := \sqrt{\langle v, v \rangle}$
$\|\cdot\|$	norm	norm induced by the current inner product, or stated otherwise
$\mathrm{Hom}(V, W)$	set	linear maps $V \to W$
$\mathrm{End}(V)$	algebra	$\mathrm{Hom}(V, V)$
$\mathrm{GL}(V)$	group	invertible linear maps $V \to V$
$\mathrm{Mat}_{m\times n}(\mathbb{R})$	set	real $m \times n$ matrices
I_n	matrix	identity matrix of size n
$\det(A)$	scalar	determinant of a square matrix A
$\mathrm{tr}(A)$	scalar	trace of a square matrix A
$\ker(T),\ \mathrm{im}(T)$	subspaces	kernel and image of a linear map T
$\mathrm{span}(S)$	subspace	linear span of a subset $S \subseteq V$
$\oplus$	operation	direct sum (internal or external as declared)
$\otimes$	operation	tensor product (introduced later; type must be specified)
$\wedge$	operation	wedge product (introduced with exterior algebra/forms)
f, g, h	maps	generic functions/maps (type declared when needed)
$Df(x)$	linear map	derivative of f at x as a linear map
$J_f(x)$	matrix	Jacobian matrix of f at x, in chosen coordinates
∇f	vector field	gradient (requires inner product/Euclidean structure)
∂_i	operator	partial derivative w.r.t. x^i
df	1-form	differential of scalar f
d	operator	exterior derivative on forms (once forms are defined)
$\int_U (\cdot)\, dx$	functional	Lebesgue/Riemann integral on $U \subseteq \mathbb{R}^n$ (as specified)
$dx,\ d^n x$	measure symbol	standard volume element on $\mathbb{R}^n$
ε, δ	scalars	small positive parameters (specified when used)
i, j, k, ℓ	indices	integer indices; range declared when ambiguous

Chapter 2

Sets, Maps, and Quotients as Computation Rules

This chapter treats basic set-theoretic constructions as *computation rules*: how to pass between raw data X, identified data $X/\sim$, and the maps that are allowed to ignore (or must respect) such identifications.

> **Guiding slogan.** A rule on a quotient is legitimate *iff* it does not depend on representatives. Equivalently: a map "descends" to the quotient *iff* it factors through the quotient map.

How to read this chapter. Sections §2.1–§2.5 are the *core*: they will be used throughout Volume I whenever we separate *objects* from their *representations*. Sections §2.6–§2.10 are an *optional interlude*: they package a few algorithmic/ML-flavored motivations. You can skip them without loss.

A note on foundations. We deliberately freeze the base layer in **Set** and postpone any discussion of infinite self-reference and higher-categorical foundations. When such questions naturally appear, we record them as motivation for later volumes, but we do not resolve them here.

2.1 Well-definedness as a commuting rule

2.1.1 Equivalence relations and quotient maps

Let X be a set and $\sim$ an equivalence relation on X. We write $[x]$ for the equivalence class of x, and denote the quotient set by

$$X/\sim \;:=\; \{[x] \mid x \in X\}.$$

The quotient map is the surjection

$$\pi : X \to X/\sim, \qquad \pi(x) := [x].$$

By definition,

$$\pi(x) = \pi(x') \iff [x] = [x'] \iff x \sim x'.$$

2.1.2 What "well-defined" means in practice

A common pattern is to propose a rule on the quotient by writing a formula in terms of a representative:

$$\bar{f}([x]) \stackrel{?}{=} f(x).$$

This is *not yet a definition*; it becomes one only after checking *well-definedness*:

Well-definedness test. The rule $\bar{f}([x]) := f(x)$ defines a function $\bar{f} : X/\sim \to Y$ *iff*

$$x \sim x' \implies f(x) = f(x').$$

Equivalently: f is constant on $\sim$-classes.

[The "commuting rule" viewpoint.] The condition above is easy to remember as a two-step commutation rule:

$$\text{(change representative)} \Rightarrow \text{(apply } f\text{)} \quad = \quad \text{(apply } f\text{)} \Rightarrow \text{(no change)}.$$

If you like diagrams, the same content is:

$$[rowsep = large, columnsep = large]X[r, "f"][d, "\pi"']Y X/\sim[ur, dashed, "\bar{f}"']$$

where the dashed arrow exists exactly when f is constant on equivalence classes.

2.1.3 Maps vs. relations (and why we state what a rule is)

In this volume, by default, a *map* means a *function*. We sometimes mention a *relation* $R \subseteq X \times Y$ as a generalized rule that may be multi-valued or partial. This matters because:

- for functions, composing rules is automatic;
- for relations, composition exists but is less "deterministic";
- the phrase "define $\bar{f}([x])$ by picking a representative" hides a choice, and choices are exactly where non-well-definedness appears.

So we state explicitly whether we mean a function, a relation, or a choice-dependent procedure.

2.1.4 A brief note on self-reference (postponed)

During our discussion we encountered the following phenomenon: to define "the closest good approximation" one must choose a notion of distance between structures; yet that choice itself depends on perspective (topological, measure-theoretic, algebraic, ...). Iterating this idea can lead to an infinite ladder of meta-choices.

[Motivation, not a foundation.] We record this as motivation for later "higher" viewpoints, but in Volume I we freeze the base layer: sets and functions. Everything below should be read as already well-defined when stated.

2.2 Inverse images, saturation, and what structure we keep

2.2.1 Inverse image as the structure-preserving choice

Given a function $f : X \to Y$ and a subset $B \subseteq Y$, the *inverse image* (preimage) is

$$f^{-1}(B) := \{x \in X \mid f(x) \in B\} \subseteq X.$$

This operation is "safe" in the sense that it respects basic set operations: for any $B_1, B_2 \subseteq Y$,

$$f^{-1}(B_1 \cap B_2) = f^{-1}(B_1) \cap f^{-1}(B_2), \qquad f^{-1}(B_1 \cup B_2) = f^{-1}(B_1) \cup f^{-1}(B_2),$$

and also $f^{-1}(Y \setminus B) = X \setminus f^{-1}(B)$.

2.2.2 Saturated subsets

Fix an equivalence relation $\sim$ on X with quotient map $\pi : X \to X/\sim$. A subset $A \subseteq X$ is called $\sim$-*saturated* if it is a union of equivalence classes, equivalently:

$$x \in A,\ x \sim x' \implies x' \in A.$$

The key fact is that inverse images of subsets of the quotient are automatically saturated: for any $B \subseteq X/\sim$,

$$\pi^{-1}(B) \subseteq X \quad \text{is saturated.}$$

2.2.3 Two easy but fundamental operations

For $A \subseteq X$, the image $\pi(A) \subseteq X/\sim$ forgets information *within* each class. The composite

$$\mathrm{Sat}(A) := \pi^{-1}(\pi(A))$$

is the *saturation* of A: it is the smallest saturated set containing A.

> **Interpretation.** $\pi(A)$ keeps only the information "which classes are touched by A"; $\mathrm{Sat}(A)$ replaces A by the union of those touched classes.

2.3 Relations, inverses, and where equalities become inclusions

2.3.1 Inverse relation

Given a relation $R \subseteq X \times Y$, its inverse relation is

$$R^{-1} := \{(y, x) \in Y \times X \mid (x, y) \in R\} \subseteq Y \times X.$$

When $f : X \to Y$ is a function, its graph $\Gamma_f \subseteq X \times Y$ is a relation; the inverse relation Γ_f^{-1} is generally *multi-valued* unless f is injective.

> [When does an inverse relation become a function?] If f is injective, then f^{-1} is a function on $\mathrm{im}(f)$. If f is surjective but not injective, the "inverse" is genuinely multi-valued. This is why some "equalities" in image/preimage calculus are safe for injections but become delicate for surjections.

2.3.2 Images and inverse images: typical safe vs. unsafe patterns

For $A \subseteq X$, the *forward image* is

$$f(A) := \{f(x) \mid x \in A\} \subseteq Y.$$

Unlike inverse images, forward images do *not* preserve intersections in general:

$$f(A \cap A') \ \subseteq\ f(A) \cap f(A') \quad \text{(may be strict).}$$

However, they do preserve unions:

$$f(A \cup A') = f(A) \cup f(A').$$

You should read these as *computation rules*:

- use f^{-1} to transport *structure* (logic of subsets) safely;
- use $f(\cdot)$ to transport *data*, but expect inclusions rather than equalities.

2.4 Equivalence relations induced by maps

2.4.1 Kernel equivalence relation

Given $f : X \to Y$, define an equivalence relation $\sim_f$ on X by

$$x \sim_f x' \quad :\Longleftrightarrow \quad f(x) = f(x').$$

This is the most canonical equivalence relation attached to f: it is the *coarsest* relation making f constant on classes.

[Fibers as equivalence classes.] The classes of $\sim_f$ are exactly the fibers $f^{-1}(\{y\})$. In set language, "quotient by $\sim_f$" collapses each fiber to a point.

2.5 The factorization theorem through quotients

2.5.1 Statement and meaning

Let $\sim$ be an equivalence relation on X with quotient map $\pi : X \to X/\sim$. For a function $f : X \to Y$, the following are equivalent:

1. f is constant on $\sim$-classes, i.e. $x \sim x' \Rightarrow f(x) = f(x')$.
2. There exists a function $\bar{f} : X/\sim \to Y$ such that $f = \bar{f} \circ \pi$.

In this case one *must* define $\bar{f}([x]) := f(x)$, and the well-definedness test guarantees that this does not depend on the representative.

2.5.2 Uniqueness and the role of surjectivity

If $\bar{f}, \bar{f}' : X/\sim \to Y$ satisfy $\bar{f} \circ \pi = \bar{f}' \circ \pi$, then for any $[x] \in X/\sim$,

$$\bar{f}([x]) = \bar{f}(\pi(x)) = f(x) = \bar{f}'(\pi(x)) = \bar{f}'([x]).$$

This uses only that every class has a representative, i.e. that π is surjective.

2.5.3 A side remark: "coupled products" and constraints

Many constructions in mathematics proceed by:

1. building a large "free" object where formulas are easy;
2. imposing relations (constraints);
3. taking a quotient by those relations.

From the present viewpoint, this is just the factorization theorem repeated as a design pattern: *a constraint is a rule that forces well-definedness under an identification.*

2.6 Hierarchical equivalence: chains, levels, and refinement

[Optional interlude.] The core of this chapter is over. The remaining sections reorganize the same ideas into a toy "multi-level classification" language and introduce a reader-defined notion of obstruction. You can skip to the exercises.

2.6.1 From "equivalence inside equivalence" to a global chain

A repeated "classify-inside-classify" process produces a hierarchy: at level 1 we partition X; inside each class we partition again (level 2); and so on. This data is equivalently encoded by a global chain of equivalence relations

$$E_n \subseteq E_{n-1} \subseteq \cdots \subseteq E_1 \subseteq X \times X,$$

where

$$x \, E_k \, y \quad :\Longleftrightarrow \quad x, y \text{ land in the same block at level } k.$$

The inclusion direction means: higher k gives a *finer* classification.

2.6.2 Maps between levels

Each inclusion $E_{k+1} \subseteq E_k$ induces a canonical quotient map

$$q_{k+1,k} : X/E_{k+1} \longrightarrow X/E_k, \qquad [x]_{E_{k+1}} \longmapsto [x]_{E_k},$$

and these maps satisfy the obvious composition rule $q_{k+2,k} = q_{k+1,k} \circ q_{k+2,k+1}$.

2.6.3 Perfect classification and true refinement

A hierarchy is "perfectly consistent" when each level is genuinely a refinement of the previous one: each class at level k is a union of classes at level $k+1$. In the chain language, this is exactly the inclusion $E_{k+1} \subseteq E_k$. So "refinement" is nothing mysterious here: it is just *well-definedness of the level map* $q_{k+1,k}$.

2.7 From chains to partial orders: nonstandard multi-level structure

2.7.1 Nonstandard n-level data as a family of equivalence relations

In real classification tasks, levels may not be totally ordered (not every pair of criteria is comparable). Instead of a chain $E_n \subseteq \cdots \subseteq E_1$, we may have a family $\{E_\alpha\}_{\alpha \in I}$ where comparability fails:

$$E_\alpha \not\subseteq E_\beta, \qquad E_\beta \not\subseteq E_\alpha.$$

When that happens, there is no canonical "map between levels"—we lose the simple quotient triangle that made the chain case clean.

2.7.2 A graph viewpoint: quotient objects and commuting triangles

You can picture each E_α as an "interface" $X \to X/E_\alpha$. In the chain case, these interfaces connect by commuting triangles. In the non-chain case, triangles fail to exist: there is no well-defined way to translate between two quotients without *adding extra structure* (a choice of reconciliation rule).

2.7.3 Why obstructions appear once comparability fails

When E_α and E_β are incomparable, there are (at least) two reasonable ways to reconcile them:

- refine until both criteria become visible at once (make things *finer*);
- coarsen until both criteria agree (make things *coarser*).

Each reconciliation discards something. Quantifying that "something" is where an obstruction score naturally appears.

2.8 Meet/join closure and a first notion of obstruction

2.8.1 The lattice operations

Equivalence relations on a fixed set X form a lattice under refinement. Concretely:

- The *meet* $E \wedge F$ is the finest relation coarser than both (intersection):

$$x\,(E \wedge F)\,y \quad :\Longleftrightarrow \quad (x\,E\,y) \text{ and } (x\,F\,y).$$

- The *join* $E \vee F$ is the coarsest relation finer than both: it is the *equivalence closure* of $E \cup F$ (add transitive consequences until you get an equivalence relation).

Intuitively: meet keeps only agreements; join forces compatibility by adding the missing identifications.

2.8.2 Closure vs. cutting

If a family of relations is not closed under $\wedge, \vee$, you can:

- *close* it (add relations you are forced to acknowledge), or
- *cut* it (discard relations to restore a simpler pattern, e.g. a chain).

Both are forms of quotient-thinking: one enlarges the rule-set, the other forgets some rules.

2.8.3 A toy obstruction principle (reader-defined)

At this level we do *not* fix a canonical obstruction. Instead we record a principle:

> **Toy obstruction principle.** Any time you must choose between "close" and "cut", you are measuring incompatibility between constraints. Any reasonable obstruction score should penalize (i) information loss and (ii) complexity growth, and should vanish exactly in the perfectly comparable (chain) case.

We intentionally leave the precise definition to the reader, because it depends on what you care about: speed, memory, interpretability, stability, ...

2.9 A dynamic toy model: cut vs. expand as an evolving rule

2.9.1 A control viewpoint

Imagine a process in which the current "state" is a family of equivalence relations $\mathcal{E}_t$ (your current classification interfaces), and at each step you may:

- *expand* (close under $\wedge, \vee$, add consequences),
- *prune* (cut relations to keep the family manageable).

You can then define a cost that mixes three ingredients:

$$\text{total cost} \;\approx\; \text{obstruction (incompatibility)} + \text{time} + \text{complexity}.$$

This viewpoint is deliberately schematic: it is only meant to say that "keeping structure" and "keeping the computation feasible" are competing constraints.

2.9.2 A remark on "greedy obsession"

A natural "obsession" is to minimize immediate step-cost at each time (greedy choice). In general combinatorial state spaces, greedy behavior need not yield global optimality: cheap expansions may cause later blow-up; cheap pruning may destroy useful information. Introducing time-penalties and computation-penalties makes the problem resemble constrained optimization and (partially observed) control.

2.9.3 Differentiable relaxations (optional playground)

Readers with ML intuition may consider the following relaxation: replace discrete equivalence relations by soft similarity matrices, and replace exact transitivity / closure by differentiable consistency penalties, so that a surrogate "obstruction loss" can be optimized by gradient descent. We do not develop this; it is offered as a playground.

2.10 Epilogue: a reader's playground

This chapter is a long way of saying: *if you want to compute on identified objects, you must prove your computation ignores representatives.* Once you internalize that, quotients stop being scary and become a design pattern.

If you enjoy the optional interlude, try inventing your own obstruction score and test it on small finite sets by brute force. If you do not, skip it: the core sections already contain everything we will use later in Volume I.

Exercises (Chapter 2)

D. Definition / Notation

1. Let $\pi : X \to X/{\sim}$ be the quotient map. Prove that π is surjective and that $\pi(x) = \pi(x')$ iff $x \sim x'$.

2. (Saturation) Let $A \subseteq X$. Show that A is $\sim$-saturated iff $A = \pi^{-1}(\pi(A))$.

3. Give an example where $A \subsetneq f^{-1}(f(A))$ for a function $f : X \to Y$.

C. Computation / Constructions

1. (Mod n) Let $\sim$ on $\mathbb{Z}$ be $a \sim b \iff a - b \in n\mathbb{Z}$. Determine whether the rules

$$[a] \mapsto a^2, \qquad [a] \mapsto 2a, \qquad [a] \mapsto a \bmod 2$$

   define well-defined maps $\mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$ (or to $\mathbb{Z}/2\mathbb{Z}$).

2. Let $f : X \to Y$ and define $x \sim_f x' \iff f(x) = f(x')$. Compute the equivalence classes for a concrete example, e.g. $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2$.

3. (Saturation explicitly) For $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2$, compute $f^{-1}(f(A))$ for $A = [-1, 2]$, and interpret it as a saturation by fibers.

P. Proof / Correctness Checks

1. Prove the factorization theorem (§2.5) carefully, including uniqueness.

2. Show that for any function $f : X \to Y$ and any $B \subseteq Y$, $f(f^{-1}(B)) \subseteq B$ and equality holds iff $B \subseteq \operatorname{im}(f)$.

3. Give a counterexample showing that $f(A \cap A') = f(A) \cap f(A')$ can fail. State a sufficient condition under which it holds.

T. Translation (Same idea, different language)

1. Translate "well-defined on a quotient" into the statement "the formula is invariant under the relation": write the invariance condition explicitly for two examples of your choice.
2. Translate the factorization theorem into a commuting triangle diagram and explain (in one paragraph) why it is the right notion of "descend to the quotient".

Optional Playground

1. (Finite partitions) Let $X = \{1, 2, 3, 4\}$ and define two equivalence relations

$$E : \{1, 2\} \mid \{3, 4\}, \qquad F : \{1, 3\} \mid \{2, 4\}.$$

Compute $E \wedge F$ and (by taking the equivalence closure of $E \cup F$) compute $E \vee F$.

Chapter 3

Vector Spaces and Linear Maps over a Field

3.1 Fields as computation rules

Throughout this chapter, let $\Bbbk$ be a field. The point of working over a field is that every nonzero scalar is invertible, so we can freely perform division—this is exactly what makes Gaussian elimination and coordinate calculus uniformly workable.

[Field] A *field* $\Bbbk$ is a set equipped with two binary operations $+$ and $\cdot$ such that:

1. $(\Bbbk, +)$ is an abelian group with identity 0;
2. $(\Bbbk \setminus \{0\}, \cdot)$ is an abelian group with identity 1;
3. multiplication distributes over addition: $a(b + c) = ab + ac$.

1. $\mathbb{R}$ and $\mathbb{C}$ are fields.
2. For a prime p, the finite field $\mathbb{F}_p = \{0, 1, \ldots, p - 1\}$ has operations defined modulo p.

[Why "field"?] Gaussian elimination repeatedly normalizes a pivot $a \neq 0$ by dividing a row by a. This requires a^{-1}, hence we work over a field. Over rings such as $\mathbb{Z}$, nonzero elements need not be invertible, so the same elimination steps are not always allowed.

[A quick contrast with $\mathbb{Z}$] In $\mathbb{Z}$, the element 2 has no multiplicative inverse (no integer x satisfies $2x = 1$), so "divide by 2" is not a legal operation.

3.2 Vector spaces and subspaces

[Vector space] A $\Bbbk$-*vector space* is a set V equipped with

1. an abelian group structure $(V, +)$ with identity 0_V;
2. a scalar multiplication map $\Bbbk \times V \to V,\ (a, v) \mapsto av$

satisfying, for all $a, b \in \Bbbk$ and $u, v \in V$:

$$a(u + v) = au + av, \quad (a + b)v = av + bv, \quad (ab)v = a(bv), \quad 1 \cdot v = v.$$

[Subspace] A subset $W \subseteq V$ is a *subspace* if it is nonempty and closed under addition and scalar multiplication.

[Subspace test] A nonempty subset $W \subseteq V$ is a subspace if and only if for all $u, v \in W$ and $a \in \Bbbk$, one has $u + v \in W$ and $au \in W$. Equivalently: W is closed under all finite $\Bbbk$-linear combinations.

[Standard examples]

1. $V = \mathbb{k}^n$ with componentwise addition and scalar multiplication.
2. $V = \mathbb{k}[x]_{\le d}$, polynomials of degree $\le d$ over $\mathbb{k}$.
3. $V = \mathbb{F}_2^n$ (a particularly computation-friendly testbed).
4. We will frequently specialize $\mathbb{k}$ to $\mathbb{R}$ or $\mathbb{C}$ in concrete calculations.

[A kernel-defined subspace] Let $V = \mathbb{k}[x]_{\le 3}$ and

$$W = \{p(x) \in V : \; p(1) = 0\}.$$

Then W is a subspace (closure follows from $(p+q)(1) = p(1) + q(1)$ and $(ap)(1) = a\,p(1)$). Moreover, $p(1) = 0$ is equivalent to $(x-1) \mid p(x)$, hence

$$W = (x-1)\mathbb{k}[x]_{\le 2} = \operatorname{span}\{(x-1),\, x(x-1),\, x^2(x-1)\},$$

so $\dim_{\mathbb{k}}(W) = 3$ and $\dim_{\mathbb{k}}(V) = 4$ (thus W has codimension 1).

3.3 Span and linear independence

[Span] Given a subset $S \subseteq V$, its *span* is

$$\operatorname{span}(S) = \left\{ \sum_{i=1}^{m} a_i s_i : \; m \ge 0,\; a_i \in \mathbb{k},\; s_i \in S \right\}.$$

It is the smallest subspace containing S.

[Linear independence] Vectors $v_1, \ldots, v_m \in V$ are *linearly independent* if

$$a_1 v_1 + \cdots + a_m v_m = 0 \implies a_1 = \cdots = a_m = 0.$$

Otherwise they are *linearly dependent*.

[Independence as a homogeneous system] Vectors $v_1, \ldots, v_m$ are linearly independent if and only if the homogeneous equation

$$a_1 v_1 + \cdots + a_m v_m = 0$$

has only the trivial solution $(a_1, \ldots, a_m) = (0, \ldots, 0)$. Equivalently, if $A = [v_1 \;\cdots\; v_m]$ is the matrix with these vectors as columns, then $v_1, \ldots, v_m$ are independent if and only if $A\mathbf{a} = 0$ has only $\mathbf{a} = 0$.

Computation recipe (RREF for span/independence). Given $v_1, \ldots, v_m \in \mathbb{k}^n$, form the matrix $A = [v_1 \;\cdots\; v_m]$. Row-reduce A to echelon form.

- Pivot columns correspond to a maximal independent subset of $\{v_i\}$.
- The number of pivots equals $\dim(\operatorname{span}\{v_i\})$.
- Free variables in $A\mathbf{a} = 0$ produce explicit linear relations among the v_i.

Over $\mathbb{F}_p$, all arithmetic is done modulo p.

[$\mathbb{F}_2^3$] In $V = \mathbb{F}_2^3$, let

$$v_1 = (1,0,1), \quad v_2 = (0,1,1), \quad v_3 = (1,1,0).$$

Then $v_1 + v_2 = (1,1,0) = v_3$, hence $v_1 + v_2 + v_3 = 0$ and the vectors are dependent. Equivalently, with $A = [v_1\; v_2\; v_3]$, row reduction over $\mathbb{F}_2$ produces a zero row, so $\operatorname{rank}(A) < 3$.

3.4 Bases and dimension

[Basis and dimension] A *basis* of V is a linearly independent set that spans V. If V has a finite basis with n elements, then V is *finite-dimensional* and we define $\dim_{\Bbbk}(V) = n$.

[Exchange step] Let $B = \{b_1, \dots, b_n\}$ be a spanning set of V and let $v \in V$. Write $v = \sum_{i=1}^{n} a_i b_i$. If $a_j \neq 0$ for some j, then

$$b_j = a_j^{-1} v - \sum_{i \neq j} a_j^{-1} a_i \, b_i,$$

so replacing b_j by v keeps the span unchanged:

$$\operatorname{span}(B) = \operatorname{span}\big((B \setminus \{b_j\}) \cup \{v\}\big).$$

[Steinitz exchange lemma] Let B be a finite spanning set of V with $|B| = n$, and let S be a linearly independent set with $|S| = m$. Then $m \leq n$. Moreover, one can replace m elements of B by the elements of S and still obtain a spanning set.

[Dimension is well-defined] Any two bases of a finite-dimensional vector space V have the same cardinality.

Computation recipe (dimension). If $V \subseteq \Bbbk^n$ is given by a spanning set (columns) or by linear equations $Ax = 0$, row-reduction computes $\dim V$ by counting pivots (rank) and free variables (nullity).

3.5 Linear maps

[Linear map] A function $T : V \to W$ is $\Bbbk$-*linear* if

$$T(u + v) = T(u) + T(v), \qquad T(av) = aT(v)$$

for all $u, v \in V$ and $a \in \Bbbk$.

[A linear map is determined by its values on a basis] Let $\mathcal{B} = (b_1, \dots, b_n)$ be a basis of V. If $T, S : V \to W$ are linear and $T(b_i) = S(b_i)$ for all i, then $T = S$. Conversely, for any choice of $w_1, \dots, w_n \in W$, there exists a unique linear map T with $T(b_i) = w_i$.

[$\Bbbk^n \to \Bbbk^m$] With the standard basis $e_1, \dots, e_n$ of $\Bbbk^n$, any $x = \sum_i x_i e_i$ satisfies

$$T(x) = \sum_{i=1}^{n} x_i \, T(e_i).$$

[Differentiation on polynomials] Let $V = \Bbbk[x]_{\leq d}$ and $D(p) = p'$. Then D is linear and $D(x^k) = kx^{k-1}$. (Over fields of characteristic p, the coefficient k is computed in $\Bbbk$, so $D(x^p) = 0$ in characteristic p.)

[Mod 2 linear maps] Over $\mathbb{F}_2$, a linear map $T : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is represented by a matrix, and $T(x) = Ax$ is computed with arithmetic modulo 2.

[Kernel and image] For a linear map $T : V \to W$, define

$$\ker(T) = \{v \in V : \; T(v) = 0\}, \qquad \operatorname{im}(T) = \{T(v) : \; v \in V\}.$$

$\ker(T)$ is a subspace of V and $\operatorname{im}(T)$ is a subspace of W.

3.6 Coordinates, matrices, and change of basis

3.6.1 Coordinate maps

Let $\mathcal{B} = (b_1, \ldots, b_n)$ be an ordered basis of V. Each $v \in V$ has a unique expression $v = \sum_{i=1}^n a_i b_i$. Define the coordinate map

$$\mathrm{coord}_{\mathcal{B}} : V \to \Bbbk^n, \qquad v \mapsto (a_1, \ldots, a_n).$$

Its inverse is

$$\mathrm{coord}_{\mathcal{B}}^{-1}(a_1, \ldots, a_n) = \sum_{i=1}^n a_i b_i.$$

3.6.2 Matrix representation

Let $T : V \to W$ be linear. Choose ordered bases $\mathcal{B}$ for V and $\mathcal{C}$ for W. The *matrix of T from $\mathcal{B}$ to $\mathcal{C}$* is the unique matrix $[T]_{\mathcal{C}\leftarrow\mathcal{B}}$ such that for all $v \in V$,

$$\mathrm{coord}_{\mathcal{C}}(T(v)) = [T]_{\mathcal{C}\leftarrow\mathcal{B}}\, \mathrm{coord}_{\mathcal{B}}(v).$$

In particular, the i-th column of $[T]_{\mathcal{C}\leftarrow\mathcal{B}}$ is

$$\mathrm{coord}_{\mathcal{C}}(T(b_i)).$$

3.6.3 Change of basis and similarity

Let $T : V \to V$ be linear and let $\mathcal{B}, \mathcal{B}'$ be two ordered bases. Define the change-of-coordinates matrix P by

$$\mathrm{coord}_{\mathcal{B}}(v) = P\, \mathrm{coord}_{\mathcal{B}'}(v) \quad \text{for all } v \in V.$$

Then the matrices of T in these bases satisfy the similarity relation

$$[T]_{\mathcal{B}'} = P^{-1}\, [T]_{\mathcal{B}}\, P.$$

3.7 Kernel, image, and rank–nullity

[Rank and nullity] For $T : V \to W$ with $\dim V < \infty$, define

$$\mathrm{rank}(T) = \dim(\mathrm{im}\, T), \qquad \mathrm{nullity}(T) = \dim(\ker T).$$

[Rank–Nullity] If $T : V \to W$ is linear and $\dim V < \infty$, then

$$\dim V = \dim(\ker T) + \dim(\mathrm{im}\, T),$$

i.e. $\mathrm{nullity}(T) + \mathrm{rank}(T) = \dim V$.

[Proof sketch] Let $\{k_1, \ldots, k_r\}$ be a basis of $\ker T$ and extend it to a basis $\{k_1, \ldots, k_r, v_{r+1}, \ldots, v_n\}$ of V. Then for any $v \in V$, writing $v = \sum_{i\le r} \alpha_i k_i + \sum_{j>r} \beta_j v_j$ gives $T(v) = \sum_{j>r} \beta_j T(v_j)$ since $T(k_i) = 0$, so $\{T(v_{r+1}), \ldots, T(v_n)\}$ spans $\mathrm{im}\, T$. If $\sum_{j>r} \gamma_j T(v_j) = 0$, then $T(\sum_{j>r} \gamma_j v_j) = 0$, so $\sum_{j>r} \gamma_j v_j \in \ker T = \mathrm{span}\{k_1, \ldots, k_r\}$, forcing all $\gamma_j = 0$ by linear independence of the chosen basis. Hence $\dim(\mathrm{im}\, T) = n - r$, and $n = r + (n - r)$.

Computation recipe (rank/nullity from RREF). If $T(x) = Ax$ with $A \in M_{m\times n}(\Bbbk)$, then $\mathrm{rank}(T)$ is the number of pivots in the RREF of A, and $\mathrm{nullity}(T) = n - \mathrm{rank}(T)$.

3.8 Linear equations and fibers

Let $A \in M_{m\times n}(\Bbbk)$ and define $T_A : \Bbbk^n \to \Bbbk^m$ by $T_A(x) = Ax$. The linear system $Ax = b$ asks for x with $T_A(x) = b$.

[Solvability] The system $Ax = b$ has a solution if and only if $b \in \operatorname{im}(T_A)$.

[Solution set is an affine translate] If x_0 is one solution of $Ax = b$, then the set of all solutions is

$$x_0 + \ker(T_A) = \{x_0 + u : \ u \in \ker(T_A)\}.$$

In particular, solutions form an affine subspace (a coset of the homogeneous solution space).

[Uniqueness criterion] If $\operatorname{rank}(A) = n$ (full column rank), then $\ker(T_A) = \{0\}$ and any solvable system $Ax = b$ has a unique solution.

[Fibers as "leaves" and torsors] For a linear map $T : V \to W$, the fibers $T^{-1}(b)$ (for $b \in \operatorname{im} T$) partition V. Each nonempty fiber is an affine translate of $\ker(T)$, hence carries a free and transitive action of the additive group $\ker(T)$:

$$u \cdot x := x + u, \qquad u \in \ker(T),\ x \in T^{-1}(b).$$

Thus each fiber is a $\ker(T)$-torsor.

[Finite field counting] If $\Bbbk = \mathbb{F}_q$ and $Ax = b$ is solvable, then the number of solutions is

$$|\ker(T_A)| = q^{\dim(\ker T_A)} = q^{n-\operatorname{rank}(A)}.$$

Optional reading: descriptions, universality, and beyond fields

(1) Kernel presentations are canonical

Many subspaces are naturally presented as kernels:

$$W = \ker(T)$$

for a linear map $T : V \to U$. This presentation is coordinate-free and interacts well with quotients. (We will revisit this systematically when discussing quotient spaces.)

(2) From fields to rings: modules

If the field $\Bbbk$ is replaced by a ring R, vector spaces become R-modules. Most kernel/image and quotient constructions survive verbatim, but the "basis + dimension" story splits: modules need not admit bases (only *free* modules do), and rank–nullity can fail in general. This is why we develop the computational linear algebra of this chapter over fields first.

(3) "Natural bases" depend on extra structure

The existence of a space is coordinate-independent, but a "natural basis" typically appears only after choosing additional data (an inner product, a symmetry, a distinguished operator, boundary conditions, etc.). Different descriptions of the same object may induce different universal constructions and thus different canonical choices (up to the appropriate equivalence). For instance, Fourier and Legendre bases arise from different operators/symmetries and different boundary/weight data.

Exercises for Chapter 3

Conventions. Unless stated otherwise, $\Bbbk$ denotes a field, V, W are $\Bbbk$-vector spaces, and all maps are $\Bbbk$-linear. We label problems by type:

D = Definitions/Basic checks, **C** = Computation, **P** = Proof, **T** = Thinking/Transfer.

D: Definitions and basic checks

D1. Let V be a $\Bbbk$-vector space and $W \subseteq V$ a nonempty subset. Show that W is a subspace iff $au + bv \in W$ for all $u, v \in W$ and all $a, b \in \Bbbk$.

D2. In $V = \Bbbk[x]_{\leq 4}$, decide whether each set is a subspace:

$$W_1 = \{p : \ p(0) = 0\}, \quad W_2 = \{p : \ p(0) = 1\}, \quad W_3 = \{p : \ p'(0) = 0\}.$$

D3. Let $T : V \to W$ be linear. Prove that $\ker(T)$ and $\operatorname{im}(T)$ are subspaces.

D4. Let $S \subseteq V$. Prove that $\operatorname{span}(S)$ is a subspace and is the smallest subspace containing S.

D5. State precisely what it means for vectors $v_1, \ldots, v_m$ to be linearly independent. Rewrite the statement using the matrix equation $A\mathbf{a} = 0$ where $A = [v_1 \ \cdots \ v_m]$.

D6. Let $\mathcal{B}$ be an ordered basis of V. Prove that $\operatorname{coord}_{\mathcal{B}} : V \to \Bbbk^n$ is linear and bijective, and write an explicit formula for its inverse.

C: Computations (over $\mathbb{R}, \mathbb{C}$, and finite fields)

C1. Over $\mathbb{R}$, let

$$v_1 = (1, 1, 0), \ v_2 = (2, 1, 1), \ v_3 = (3, 2, 1), \ v_4 = (1, 0, 1) \in \mathbb{R}^3.$$

Compute $\dim(\operatorname{span}\{v_1, v_2, v_3, v_4\})$ and extract a basis using RREF.

C2. Over $\mathbb{F}_5$, repeat **C1** with the same vectors viewed in $\mathbb{F}_5^3$. Compare your answers with the $\mathbb{R}$ case.

C3. Let $W = \{p \in \Bbbk[x]_{\leq 3} : p(1) = 0\}$. Find a basis of W and compute $\dim W$.

C4. In $V = \Bbbk[x]_{\leq 4}$, let $U = \{p : \ p(0) = 0, \ p(1) = 0\}$. Compute $\dim U$ and exhibit a basis.

C5. Let $T : \mathbb{R}^3 \to \mathbb{R}^2$ be given by

$$T(x, y, z) = (x + y, \ y + z).$$

Compute $\ker(T)$ and $\operatorname{im}(T)$ and verify rank–nullity numerically.

C6. Over $\mathbb{F}_2$, let $A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$. Compute $\operatorname{rank}(A)$ and the number of solutions to $Ax = b$ for each $b \in \mathbb{F}_2^3$.

C7. Let $T : \Bbbk^3 \to \Bbbk^3$ satisfy

$$T(e_1) = (1, 0, 0), \quad T(e_2) = (1, 1, 0), \quad T(e_3) = (0, 1, 1).$$

Compute the matrix of T in the standard basis, then compute $T(2, -1, 3)$ over $\mathbb{R}$ and over $\mathbb{F}_5$.

C8. Let $V = \mathbb{R}^2$ with bases

$$\mathcal{B} = (b_1, b_2), \ b_1 = (1, 0), \ b_2 = (1, 1), \qquad \mathcal{B}' = (b_1', b_2'), \ b_1' = (1, 1), \ b_2' = (0, 1).$$

Compute the change-of-coordinates matrix P such that $\operatorname{coord}_{\mathcal{B}}(v) = P \operatorname{coord}_{\mathcal{B}'}(v)$.

C9. Using **C8**, let $T : \mathbb{R}^2 \to \mathbb{R}^2$ be the linear map with matrix $[T]_{\mathcal{B}} = \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}$. Compute $[T]_{\mathcal{B}'}$ via the similarity formula.

P: Proof problems (core theory)

P1. (Exchange step) Let $B = \{b_1, \ldots, b_n\}$ span V and let $v = \sum_{i=1}^{n} a_i b_i$ with $a_j \neq 0$. Prove that $\mathrm{span}(B) = \mathrm{span}((B \setminus \{b_j\}) \cup \{v\})$.

P2. (Steinitz exchange lemma) Prove: if B is a finite spanning set with $|B| = n$ and S is linearly independent with $|S| = m$, then $m \leq n$. (You may use **P1** repeatedly.)

P3. Prove that any two bases of a finite-dimensional vector space have the same cardinality.

P4. Prove: a linear map $T : V \to W$ is uniquely determined by its values on a basis of V.

P5. Prove the column description of a matrix representation: for $T : V \to W$ and bases $\mathcal{B} = (b_i)$, $\mathcal{C} = (c_j)$, show that the i-th column of $[T]_{\mathcal{C}\leftarrow\mathcal{B}}$ equals $\mathrm{coord}_{\mathcal{C}}(T(b_i))$.

P6. Prove the change-of-basis (similarity) formula: if $\mathrm{coord}_{\mathcal{B}}(v) = P\,\mathrm{coord}_{\mathcal{B}'}(v)$ for all $v \in V$, then

$$[T]_{\mathcal{B}'} = P^{-1}[T]_{\mathcal{B}}P.$$

P7. Prove rank–nullity using the "extend a basis of $\ker T$" method.

P8. Let $T : V \to W$ be linear and $b \in W$. Show: if $T^{-1}(b)$ is nonempty, then $T^{-1}(b) = x_0 + \ker T$ for any $x_0 \in T^{-1}(b)$.

T: Thinking and transfer (structure, geometry, and finite fields)

T1. (Fibers as leaves) Let $T : \mathbb{R}^2 \to \mathbb{R}$ be $T(x, y) = x + y$. Describe geometrically the fibers $T^{-1}(b)$ and compare them with the fibers of $T(x, y) = x$.

T2. (Torsor viewpoint) Let $T : V \to W$ be linear. Explain why every nonempty fiber $T^{-1}(b)$ carries a free and transitive action of the additive group $\ker T$. What is the "missing datum" that prevents a fiber from being a vector subspace (unless $b = 0$)?

T3. (Finite field counting) Let $\Bbbk = \mathbb{F}_q$ and $A \in M_{m\times n}(\Bbbk)$. Show: if $Ax = b$ is solvable, then the number of solutions is $q^{n-\mathrm{rank}(A)}$.

T4. Give an explicit example of a matrix A such that $\mathrm{rank}(A)$ over $\mathbb{R}$ differs from $\mathrm{rank}(A)$ over $\mathbb{F}_p$ for some prime p. Explain the mechanism.

T5. Let $D : \Bbbk[x]_{\leq d} \to \Bbbk[x]_{\leq d-1}$ be differentiation. Compute $\dim\ker D$ and $\dim \mathrm{im} D$ when $\mathrm{char}(\Bbbk) = 0$. Then explore what changes when $\mathrm{char}(\Bbbk) = p > 0$ (try $d \geq p$).

T6. (Coordinate-free vs coordinates) Let $T : V \to V$. Explain in your own words which parts of linear algebra are invariant under change of basis and which parts depend on a basis choice.

T7. (Optional: from fields to rings) Replace $\Bbbk$ by $R = \mathbb{Z}$ and consider the $\mathbb{Z}$-module homomorphism $T : \mathbb{Z} \to \mathbb{Z}$ given by $T(n) = 2n$. Compute $\ker T$ and $\mathrm{coker}\, T = \mathbb{Z}/\mathrm{im}\, T$. Compare this with what you would expect over a field.

T8. (Optional: free vs non-free) Show that $\mathbb{Z}/2\mathbb{Z}$ is a $\mathbb{Z}$-module that is finitely generated but not free. What goes wrong if one tries to define a "basis" for it?

Chapter 4

Dual Spaces and Index Calculus

This chapter is a computational "grammar" chapter. We introduce dual spaces and fix a consistent index convention. The goal is not geometric sophistication, but the ability to simplify expressions such as $\omega_{ij}v^j$, $g^{ik}g_{kj}$, $\nabla_i(J^i{}_jX^j)$, etc. Many symbols are imported from later subjects (Riemannian/symplectic/complex/contact geometry and geometric analysis). You are *not* required to know what they mean yet: treat them as arrays with stated algebraic properties and simplify by index rules.

4.1 Dual space as the space of linear measurements

Let V be a finite-dimensional vector space over a field $\Bbbk$ (usually $\mathbb{R}$ or $\mathbb{C}$). The *dual space* is

$$V^* := \mathrm{Hom}_{\Bbbk}(V, \Bbbk),$$

the vector space of linear maps $\varphi : V \to \Bbbk$ (also called *covectors* or *linear functionals*).

There is a canonical bilinear pairing

$$\langle \cdot, \cdot \rangle : V^* \times V \to \Bbbk, \qquad \langle \varphi, v \rangle := \varphi(v).$$

This pairing is *canonical* (no choices involved). However, an identification $V \cong V^*$ is *not* canonical unless extra structure (e.g. an inner product / metric) is given.

4.1.1 Dual basis and the Kronecker delta

Fix a basis $\{e_i\}_{i=1}^n$ of V. The *dual basis* $\{\varepsilon^i\}_{i=1}^n \subset V^*$ is defined by

$$\varepsilon^i(e_j) = \delta^i{}_j,$$

where $\delta^i{}_j$ is the Kronecker delta (identity matrix components):

$$\delta^i{}_j = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

Every vector and covector expand uniquely as

$$v = v^i e_i, \qquad \varphi = \varphi_i\, \varepsilon^i,$$

and the canonical pairing becomes

$$\langle \varphi, v \rangle = \varphi_i v^i.$$

Convention (important).

- Indices upstairs are *contravariant* components (vectors).
- Indices downstairs are *covariant* components (covectors).
- Summation convention (Einstein): repeated indices *one up and one down* are summed.
- $\delta^i{}_j$ is the identity map on V in components. We do *not* write δ_{ij} unless a metric is present.

4.2 Change of basis and how indices transform

Let $\{e_i\}$ and $\{e'_i\}$ be two bases. We adopt the following *basis-change convention*:

$$e'_j = P^i{}_j \, e_i \quad \text{for an invertible matrix } P = (P^i{}_j).$$

Then the dual basis transforms by the inverse:

$$\varepsilon'^j = (P^{-1})^j{}_i \, \varepsilon^i.$$

Consequently, the components transform as

$$v = v^i e_i = v'^j e'_j \implies v'^j = (P^{-1})^j{}_i \, v^i,$$

$$\varphi = \varphi_i \varepsilon^i = \varphi'_j \varepsilon'^j \implies \varphi'_j = P^i{}_j \, \varphi_i.$$

The scalar pairing is basis-independent:

$$\varphi_i v^i = \varphi'_j v'^j.$$

4.3 Dual (transpose) map

Let $T : V \to W$ be a linear map. Its *dual map* (also called transpose or pullback) is

$$T^* : W^* \to V^*, \qquad (T^*\psi)(v) := \psi(Tv).$$

In matrix form, if $[T]^a{}_i$ are the components of T relative to chosen bases, then for $\psi \in W^*$ we have

$$(T^*\psi)_i = [T]^a{}_i \, \psi_a.$$

The slogan: *vectors push forward, covectors pull back.*

4.4 Tensors and index bookkeeping

A *tensor of type* (r, s) on V is an element of

$$T^{(r,s)}(V) := \underbrace{V \otimes \cdots \otimes V}_{r \text{ times}} \otimes \underbrace{V^* \otimes \cdots \otimes V^*}_{s \text{ times}}.$$

In a basis, a tensor $A \in T^{(r,s)}(V)$ has components

$$A^{i_1 \cdots i_r}{}_{j_1 \cdots j_s}.$$

The *only* legal automatic summation is one-up/one-down repetition.

4.4.1 Dummy indices and renaming

Repeated indices are *dummy* and may be renamed:

$$A^i{}_j v^j = A^i{}_k v^k.$$

Free indices must match on both sides of an equation. For example,

$$A^i{}_j v^j = w^i \quad \text{has free index } i.$$

4.4.2 Contraction and trace

Contraction means summing one upper with one lower index:

$$\mathrm{tr}(A) := A^i{}_i.$$

More generally, for $B^{ij}{}_k$ we can contract i with k:

$$C^j := B^{ij}{}_i.$$

4.5 Metrics and raising/lowering indices (non-canonical unless given)

A *(pseudo-)Riemannian metric* on V is a non-degenerate symmetric bilinear form

$$g : V \times V \to \Bbbk, \qquad g(u,v) = g(v,u).$$

In a basis, write $g_{ij} := g(e_i, e_j)$. Non-degeneracy means the matrix (g_{ij}) is invertible; denote its inverse by (g^{ij}), i.e.

$$g^{ik} g_{kj} = \delta^i{}_j, \qquad g_{ik} g^{kj} = \delta_i{}^j.$$

With a metric, we can convert vectors and covectors:

$$v_i := g_{ij} v^j, \qquad \alpha^i := g^{ij} \alpha_j.$$

Warning: without a metric, there is no canonical meaning to "lowering" or "raising".

4.6 Standard algebraic gadgets from later geometry

The following symbols will appear in later volumes; here we only record their algebraic rules.

4.6.1 Almost complex structure

An array $J^i{}_j$ satisfying

$$J^i{}_k J^k{}_j = -\delta^i{}_j$$

is treated as a linear operator with $J^2 = -\mathrm{Id}$.

4.6.2 Symplectic form

An array ω_{ij} is *skew-symmetric* if

$$\omega_{ij} = -\omega_{ji}.$$

If ω is non-degenerate, it has an inverse ω^{ij} (also skew) such that

$$\omega^{ik} \omega_{kj} = \delta^i{}_j.$$

Think of ω^{ij} as the "inverse matrix" of ω_{ij}.

4.6.3 Levi-Civita symbol

In dimension n, the totally antisymmetric symbol $\varepsilon_{i_1\cdots i_n}$ satisfies

$$\varepsilon_{12\cdots n} = +1, \quad \varepsilon_{i_1\cdots i_n} \text{ changes sign under odd permutations,}$$

and is 0 if any indices repeat. (We will later relate it to volume forms and Hodge star; for now it is a combinatorial tensor.)

4.6.4 Contact form

A covector α_i is just a covector; sometimes we also see a skew object

$$(d\alpha)_{ij} = -(d\alpha)_{ji}.$$

No geometry is needed here: simplify by skew-symmetry and index rules.

4.7 A checklist for safe index manipulations

When simplifying an expression, follow this order:

1. Identify free indices (must remain free in the final expression).
2. Rename dummy indices to avoid collisions.
3. Apply $\delta^i{}_j$ to eliminate indices: $\delta^i{}_j v^j = v^i$.
4. Use inverse relations: $g^{ik}g_{kj} = \delta^i{}_j$, $\omega^{ik}\omega_{kj} = \delta^i{}_j$.
5. Use symmetry/antisymmetry rules to cancel terms.
6. Only then, if a metric is given, raise/lower indices carefully.

Exercises: index simplification toolbox

A. Warm-up: deltas, traces, dummy indices

1. Simplify: $\delta^i{}_j v^j$.
2. Simplify: $A^i{}_j \delta^j{}_k v^k$.
3. Show that $\mathrm{tr}(A) = A^i{}_i$ is invariant under renaming dummy indices by rewriting $A^i{}_i = A^j{}_j$.
4. Simplify: $\delta^i{}_j \delta^j{}_k \delta^k{}_m v^m$.
5. Let $B^{ij}{}_k$ be any array. Simplify $B^{ij}{}_k \delta^k{}_i$.
6. Suppose $S_{ij} = S_{ji}$ is symmetric and $K^{ij} = -K^{ji}$ is skew. Simplify $S_{ij}K^{ij}$.
7. Suppose $A^i{}_j$ is any array. Simplify

$$A^i{}_j v^j - A^i{}_k v^k.$$

B. Metric-style identities (Riemannian geometry algebra without geometry)

Assume $g_{ij} = g_{ji}$ is invertible with inverse g^{ij}.

1. Simplify: $g^{ik} g_{kj} v^j$.
2. If $v_i := g_{ij} v^j$, simplify $g^{ij} v_j$.
3. Simplify: $g_{ij} g^{ij}$ in terms of $n = \dim V$.
4. Let A_{ij} be any array. Define $A^i{}_j := g^{ik} A_{kj}$. Simplify $A^i{}_j v^j$ in terms of A_{kj} and v^j.
5. Suppose $T_{ij} = T_{ji}$ is symmetric. Simplify
$$T_{ij} v^i w^j - T_{ij} w^i v^j.$$
6. Suppose R_{ijkl} satisfies $R_{ijkl} = -R_{jikl}$ and $R_{ijkl} = -R_{ijlk}$ (skew in first pair and last pair). Show that $R_{iikl} = 0$ (i.e. contract the first pair).

C. Symplectic-style identities (antisymmetry and inverse)

Assume $\omega_{ij} = -\omega_{ji}$ is invertible with inverse $\omega^{ij} = -\omega^{ji}$.

1. Simplify: $\omega^{ik} \omega_{kj} v^j$.
2. If $X^i := \omega^{ij} \alpha_j$, simplify $\omega_{ki} X^i$.
3. Show that $\omega_{ij} v^i v^j = 0$ for any v^i.
4. Let A_{ij} be symmetric: $A_{ij} = A_{ji}$. Simplify $A_{ij} \omega^{ij}$.
5. Simplify the skew-symmetrization
$$B_{[ij]} := \tfrac{1}{2}(B_{ij} - B_{ji})$$
for $B_{ij} = \omega_{ik} A^k{}_j$ in the special case when $A^k{}_j = \delta^k{}_j$.

D. Complex-geometry-flavored identities (just $J^2 = -I$)

Assume $J^i{}_k J^k{}_j = -\delta^i{}_j$.

1. Simplify: $J^i{}_k J^k{}_j v^j$.
2. Show that $J^i{}_j$ is invertible and write a formula for $(J^{-1})^i{}_j$ using indices.
3. Simplify: $J^i{}_j v^j + J^i{}_k v^k$.
4. Suppose g_{ij} is a symmetric invertible array and J is g-orthogonal in the sense
$$g_{ab} J^a{}_i J^b{}_j = g_{ij}.$$
Show (by pure index algebra) that J preserves the inverse metric:
$$g^{ab} J^i{}_a J^j{}_b = g^{ij}.$$
(Hint: multiply the given identity by $g^{pi} g^{qj}$.)

E. Contact / geometric analysis style: "derivatives" as formal indices

In this block, treat ∇_i as a *formal derivation* satisfying the Leibniz rule

$$\nabla_i(AB) = (\nabla_i A)B + A(\nabla_i B),$$

and $\nabla_i(\delta^k{}_j) = 0$. Do not assume anything else.

1. Expand (Leibniz): $\nabla_i(\phi\, v^i)$ where ϕ is a scalar.
2. Expand: $\nabla_i(A^i{}_j X^j)$.
3. If $g^{ij} g_{jk} = \delta^i{}_k$ and $\nabla_\ell g_{ij} = 0$, show that $\nabla_\ell g^{ij} = 0$ by differentiating $g^{ij} g_{jk} = \delta^i{}_k$ and simplifying.
4. Let $(d\alpha)_{ij} = -(d\alpha)_{ji}$. Simplify
$$(d\alpha)_{ij} v^i v^j.$$
5. (Index relabeling practice) Show that
$$(\nabla_i X^i)(\nabla_j Y^j) = (\nabla_a X^a)(\nabla_b Y^b).$$
6. Expand and simplify using δ:
$$\nabla_i(\delta^i{}_j X^j).$$

F. Levi-Civita symbol drills (optional but useful later)

Assume dimension $n = 3$ for simplicity, and ε_{ijk} is totally antisymmetric.

1. Show that $\varepsilon_{ijk} v^i v^j = 0$ for any v^i.
2. Simplify $\varepsilon_{ijk} \delta^i{}_a \delta^j{}_b \delta^k{}_c$.
3. (Pure algebra identity; you may accept it as a rule if you haven't seen it) Use
$$\varepsilon_{ijk} \varepsilon^{imn} = \delta_j{}^m \delta_k{}^n - \delta_j{}^n \delta_k{}^m$$
to simplify $\varepsilon_{ijk} \varepsilon^{imn} A^j{}_m B^k{}_n$.

Short answers (for self-check)

A1. v^i.

A2. $A^i{}_k v^k$.

A4. v^i.

A5. $B^{ij}{}_i$.

A6. 0.

A7. 0.

B1. v^i.

B2. v^i.

B3. n.

B5. 0.

B6. 0.

C1. v^i.

C2. α_k.

C3. 0.

C4. 0.

C5. $B_{[ij]} = \frac{1}{2}(\omega_{ij} - \omega_{ji}) = \omega_{ij}$.

D1. $-v^i$.

D2. $(J^{-1})^i{}_j = -J^i{}_j$.

D3. $2J^i{}_j v^j$.

D4. $g^{ab} J^i{}_a J^j{}_b = g^{ij}$.

E1. $(\nabla_i \phi) v^i + \phi\, \nabla_i v^i$.

E2. $(\nabla_i A^i{}_j) X^j + A^i{}_j \nabla_i X^j$.

E3. $\nabla_\ell g^{ij} = 0$.

E4. 0.

E6. $\nabla_i X^i$.

F2. ε_{abc}.

Chapter 5

Inner Product Spaces and Orthogonal Decompositions

Chapter 6

Determinants, Volume, and the Shadow of Exterior Algebra

Chapter 7

Eigenvalues, Spectral Decomposition, and Quadratic Forms

Chapter 8

Multilinear Algebra and Tensor Products (Minimum Operational Core)

Chapter 9

Tensors and Index Notation (Einstein Summation)

Chapter 10

Matrix Groups and Linear Symmetries (A Minimal Interface)

Chapter 11

Differentiation in $\mathbb{R}^n$: Fréchet Derivative and the Chain Rule

Chapter 12

Higher Derivatives: Hessians and Second-Order Structure

Chapter 13

Inverse Function Theorem: Usable Statements and Worked Examples

Chapter 14

Implicit Function Theorem: Constraints, Local Graphs, and Derivatives

Chapter 15

Multiple Integrals and the Change-of-Variables Formula

Chapter 16

Line/Surface Integrals and Green–Stokes–Gauss as Practice

Chapter 17

Exterior Algebra and Basic Differential Forms (Operational Entry)

Chapter 18

Linear ODE Systems and the Exponential Map e^{tA}

Chapter 19

A Minimal Variational Toolkit and Euler–Lagrange Equations

Chapter 20

Summary, Checklists, and a Toolbox Index for Later Volumes