Abridged Math Foundations For Signals and Systems

An eight-part refresher

August 2026 (revised)

How to use this booklet

Machine learning is usually taught atop a full linear algebra course, yet it uses a surprisingly small slice of the subject — eigenvectors, a pillar of most linear algebra courses, barely appear. What machine learning actually leans on, constantly, is: vectors as feature representations, dot products as scores, norms as loss magnitudes, matrices as linear maps and datasets, spans/rank when reasoning about what a model can express, projections and least squares behind linear regression, and gradients of vector functions for gradient descent.

Part I covers exactly that slice, in seven lessons plus a capstone.

Machine learning also assumes a first course in *probability*, and unlike the eigenvector story, it uses this one heavily: Bayes' rule and conditional independence power the Bayesian-networks and HMM lectures, expectations define the utilities that MDPs and game-playing maximize, and every loss function is an expectation over data. **Part III** covers that probability core — probabilistic models, conditioning and Bayes' rule, discrete and continuous random variables, expectation — in four lessons plus a capstone, following Bertsekas & Tsitsiklis, *Introduction to Probability* (2nd ed.), Chapters 1–3. Do it right after Part I.

Part II continues the linear algebra through the rest of Axler's book (Chapters 5–9): complex vector spaces, eigenvalues and diagonalization, the spectral theorem, unitary matrices and the discrete Fourier transform, the singular value decomposition, and determinants and trace. Its target is *Fourier analysis* — a subject whose linear-algebra backbone is precisely orthonormal expansions, eigen-analysis of linear systems, and the DFT as a unitary matrix. Part II is not needed for the machine-learning core and can be done at a relaxed pace.

Part IV continues the probability toward *statistical inference and statistical signal processing*, which assume a full first course in probability plus linear systems and Fourier transforms, and then build: random vectors and random processes, convergence and limit theorems, Markov and Gaussian processes, stationarity and autocorrelation, minimum-mean-square-error and linear estimation, Kalman filtering, hypothesis testing, and PCA. Part IV is the prerequisite half of that list, from Bertsekas & Tsitsiklis Chapters 4–8: derived distributions, covariance, conditional expectation as an estimator, random vectors and Gaussian vectors, limit theorems, the Bernoulli and Poisson processes, Markov chains, and least-mean-squares estimation. Part IV leans on Part III throughout and, at marked moments, on Part II: Lesson 22 reuses Lesson 6's projection geometry, Lesson 23 reuses Lesson 11's spectral theorem, and Lesson 26 reuses Lesson 10's eigenvalues.

Part V completes the undergraduate prerequisite set with the *signals-and-systems* core, following Oppenheim & Willsky (O&W), *Signals and Systems* (2nd ed.) — the linear-systems layer that both Fourier analysis and statistical inference stand on: signals and system proper-

ties, LTI systems and convolution, Fourier series, the continuous-time and discrete-time Fourier transforms, frequency response and filtering, sampling and aliasing, and a working minimum of Laplace and z-transforms. Part V leans on Part II constantly (complex exponentials, the eigenfunction idea, the DFT): do it right after Part II. Parts IV and V are independent of each other; statistical inference wants both.

Part VI is the *applied* layer: an abridgement of the Course 1 (diiv.io) Phase 4–6 booklets — digital and statistical signal processing (Lyons, Kuo, Hayes, Grewal–Andrews), radar and digital communication (Richards, Proakis–Salehi), image processing (Gonzalez–Woods), and audio and learned signal processing (Zölzer) — plus a working minimum of bench electronics from Scherz–Monk, *Practical Electronics for Inventors* (4th ed.). It is sized for the applied subjects downstream — **digital signal processing**, **digital image processing**, **digital communication**, and **computer vision** — and for the **Course 2 labs** at diiv.io (embedded DSP on the STM32, Pi 5, and Jetson: bench instruments through op-amps, real-time filters, FFT analyzers, Kalman and matched filters, CFAR, adaptive equalizers, and edge-ML audio/image pipelines). Each lesson names the labs it unlocks. When a topic deserves more than its abridged treatment here, the pointer is always the full phase booklet (*DiivCourse1DSPMathFoundations*).

Part VII is the optional *rigor layer*: it explains, with the bare minimum of real analysis (Ross), measure theory (Axler's MIRA), functional analysis (Bowers–Kalton), complex analysis (Bak–Newman), distribution theory (Strichartz), and numerical linear algebra (Trefethen–Bau), *why* Part V's special integrals, derivatives, δ-manipulations, transform inversions, and functional derivatives are legal — when limits pass through integrals, what $\delta(t)$ actually is, why poles and ROCs behave, what space signals live in, and when to trust the arithmetic. Nothing in it is formally required by the earlier parts, but graduate Fourier analysis develops exactly its distribution theory, statistical inference's "almost surely" is its "almost everywhere," and it is deliberately built as the on-ramp to graduate-level DSP (estimation, adaptive filters, wavelets), whose native language — Hilbert spaces, operators, distributions, conditioning — it installs. Do it after Part V, at a relaxed pace, or interleave it with the courses themselves.

Part VIII is the *optimization and information layer*: it adds the parts of Boyd–Vandenberghe's *Convex Optimization* and Polyanskiy–Wu's *Information Theory: From Coding to Learning* that recur in machine learning, signal processing, communications, and sensors. Convex optimization turns fitting, regularization, filter design, detector design, and resource allocation into problems with global certificates; information theory turns uncertainty, distinguishability, compression, channel capacity, and representation quality into computable quantities. Do it after Parts I, III, and V; Parts VI–VII make the signal-processing and rigor connections richer, but the first pass through Part VIII is self-contained.

Each lesson has two interleaved layers:

- **Theory** in the style of Axler's *Linear Algebra Done Right* (LADR, 4th ed.) for Parts I–II, of Bertsekas–Tsitsiklis (B&T) for Parts III–IV, of Oppenheim–Willsky (O&W) for Part V, engineering-recipe style (after Lyons, Hayes, Gonzalez–Woods, Zölzer, and Scherz–Monk) for Part VI, and of Ross, Axler (MIRA), Bak–Newman, Strichartz, Bowers–Kalton, and Trefethen–Bau for Part VII (one source per lesson), and Boyd–Vandenberghe plus Polyanskiy–Wu for Part VIII: precise definitions and theorems with complete proofs where feasible. Understanding *why* a fact is true is what makes it stick.

- **Concrete practice** in the style of Strang's *Introduction to Linear Algebra*: actual small examples — 2- and 3-dimensional vectors, dice, coins, three-state chains — computed by

hand, with geometric pictures in words.

Exercises at the end of each lesson are tagged **[Proof]** or **[Hand]**. Do all of the [Hand] ones and as many [Proof] ones as time allows — the starred proofs are the most valuable.

What each subject needs from this booklet

Machine learning — linear models, SGD, features, neural networks, state-based search, Markov decision processes, game playing, constraint satisfaction, and graphical models (Bayesian networks and hidden Markov models) — needs Part I (the ML core) and Part III (the MDP, game-playing, and graphical-models material) — plus the Markov-chain lesson of Part IV if you want MDPs and HMMs to feel like review rather than news.

Statistical inference assumes all of Part III as a *floor* and then builds on precisely the material of Part IV; its linear-systems and Fourier prerequisite is Part V, whose own linear-algebra spine is Part II. *Fourier analysis* stands on Part V's first half, with Part II as its linear-algebra backbone. With all five parts, the undergraduate prerequisite set for all three subjects is complete.

Every lesson below carries a NEEDED FOR: tag. The booklet-wide map (*ML* = machine learning; *inference* = statistical inference):

Lessons	**Machine learning**	**Fourier / inference**
1–8 (Part I: linear algebra core)	**required**	assumed background for both
9–15 (Part II: eigen/Fourier/SVD)	not needed	**Fourier core**; inference (PCA, DFT)
16–20 (Part III: probability core)	**required**	**inference assumes it** (first-course level)
21 (derived dists, covariance)	not needed	**inference**
22 (conditional expectation, LMS)	not needed	**inference** (MMSE, Kalman lead-in)
23 (random vectors, Gaussians)	not needed	**inference** (PCA, Gaussian processes)
24 (limit theorems)	helpful (Monte Carlo)	**inference** (convergence, concentration)
25 (Bernoulli/Poisson processes)	not needed	**inference** (first random processes)
26 (Markov chains)	**helpful** (MDPs, HMMs)	**inference** (Markov processes)
27 (Capstone IV)	not needed	**inference**
28–35 (Part V: signals & systems)	not needed	**both** (linear-systems prerequisite)
36–44 (Part VI: applied DSP)	not needed	see second table below
45–52 (Part VII: the rigor layer)	not needed	optional for both; **grad DSP**
53–60 (Part VIII: optimization & information)	**ML bridge**	**sensors, DSP, comms**

Part VI is scoped to the Course 2 labs — each lesson covers what its labs need and no more — and doubles as the abridged on-ramp to the four applied subjects:

Lessons	**Subjects**	**Course 2 labs**
36 (DFT in practice, FFT, Goertzel)	**DSP**	6.3–6.5
37 (FIR/IIR design, finite word lengths)	**DSP**	6.1–6.2
38 (multirate, oversampling, $\Sigma\Delta$)	**DSP**	3.2–3.5, 5.4
39 (random signals, spectrum estimation)	**DSP**, communications	6.4
40 (Wiener, LMS, Kalman)	**DSP**, communications	6.6, 6.8
41 (matched filter, CFAR, equalization)	**communications**	6.7–6.9
42 (images: convolution, filtering, motion)	**image processing**, **vision**	9.4–9.6
43 (audio effects and learned DSP)	DSP	8.2–8.4, 9.2–9.3
44 (the electronics bench, after PEfI)	—	Modules 0–4

In one sentence: *for machine learning do Lessons 1–8 and 16–20 (and skim 26); Parts II and V are for Fourier analysis; Parts II–V together are the full statistical-inference load-out; Part VI is the abridged on-ramp to applied DSP, image processing, communications, and computer vision, and to the Course 2 labs; Part VII is the optional "why it all works" layer and the bridge to graduate DSP; Part VIII is the convex-optimization and information-theory bridge for ML, sensors, DSP, and communications.*

Schedule

Part I — the linear algebra sprint:	
Day 1 (Sat 8/8)	Lesson 1: Vectors and linear combinations
Day 2 (Sun 8/9)	Lesson 2: Dot products, norms, orthogonality
Day 3 (Mon 8/10)	Lesson 3: Span, independence, basis, dimension
Day 4 (Tue 8/11)	Lesson 4: Matrices and linear maps
Day 5 (Wed 8/12)	Lesson 5: Null space, rank, and solving $Ax = b$
Day 6 (Thu 8/13)	Lesson 6: Projections and least squares
Day 7 (Fri 8/14)	Lesson 7: Gradients for machine learning
Day 8 (Sat 8/15)	Capstone: linear regression from scratch in NumPy
Day 9 (Sun 8/16)	Buffer: unfinished exercises; skim the self-check list
Part III — the probability core, one lesson per day, starting 8/17:	
Day 1	Lesson 16: Probability models, conditioning, and Bayes' rule
Day 2	Lesson 17: Discrete random variables and expectation
Day 3	Lesson 18: Multiple random variables and conditioning
Day 4	Lesson 19: Continuous random variables and the normal
Day 5	Capstone III: a naive Bayes classifier from scratch
Part II — one lesson per day, after Part III:	
Day 1	Lesson 9: Complex vectors and complex exponentials
Day 2	Lesson 10: Eigenvalues, eigenvectors, diagonalization
Day 3	Lesson 11: Self-adjoint operators and the spectral theorem
Day 4	Lesson 12: Unitary matrices and the discrete Fourier transform
Day 5	Lesson 13: The singular value decomposition
Day 6	Lesson 14: Determinants and trace
Day 7	Capstone II: the DFT in NumPy
Part V — one lesson per day, right after Part II:	
Day 1	Lesson 28: Signals and systems
Day 2	Lesson 29: LTI systems and convolution
Day 3	Lesson 30: Fourier series
Day 4	Lesson 31: The continuous-time Fourier transform
Day 5	Lesson 32: The discrete-time Fourier transform and frequency response
Day 6	Lesson 33: Sampling
Day 7	Lesson 34: Laplace and z-transforms, a working minimum
Day 8	Capstone V: filtering and sampling in NumPy
Part IV — one lesson per day, after Parts II & III:	
Day 1	Lesson 21: Derived distributions, covariance, correlation
Day 2	Lesson 22: Conditional expectation and least mean squares
Day 3	Lesson 23: Random vectors and Gaussian vectors
Day 4	Lesson 24: Limit theorems
Day 5	Lesson 25: The Bernoulli and Poisson processes
Day 6	Lesson 26: Markov chains
Day 7	Capstone IV: estimation and simulation in NumPy

Part VI — after Part V; before the applied courses / Course 2 labs:	
Day 1	Lesson 36: The DFT in practice: leakage, windows, FFT, Goertzel
Day 2	Lesson 37: Digital filters: FIR, IIR, finite word lengths
Day 3	Lesson 38: Multirate, oversampling, $\Sigma\Delta$ conversion
Day 4	Lesson 39: Random signals and spectrum estimation
Day 5	Lesson 40: Optimal and adaptive filters: Wiener, LMS, Kalman
Day 6	Lesson 41: Detection and digital communication
Day 7	Lesson 42: Images: convolution, filtering, edges, motion
Day 8	Lesson 43: Audio effects and learned DSP
Day 9	Lesson 44: The electronics bench (PEfI minimum)
Part VII — optional, after Part V; relaxed pace, or interleaved with the courses:	
Day 1	Lesson 45: Limits, series, and when swapping is legal
Day 2	Lesson 46: Lebesgue integration, a.e., and the swap theorems
Day 3	Lesson 47: Banach/Hilbert spaces, L^2, orthonormal bases
Day 4	Lesson 48: Complex analysis: poles, ROCs, inversion
Day 5	Lesson 49: Distributions: making δ honest
Day 6	Lesson 50: Dual spaces and functional derivatives
Day 7	Lesson 51: Numerical linear algebra: conditioning & structure
Day 8	Capstone VII: the certification lab
Part VIII — after Parts I, III, and V; before ML/sensors/comms depth:	
Day 1	Lesson 53: Convex models: sets, functions, and problems
Day 2	Lesson 54: Regularized fitting and inverse problems
Day 3	Lesson 55: Duality, KKT conditions, and certificates
Day 4	Lesson 56: Gradient, Newton, and proximal algorithms
Day 5	Lesson 57: Entropy, KL divergence, and mutual information
Day 6	Lesson 58: Testing, channels, and sensor information
Day 7	Lesson 59: Rate-distortion, compression, representation learning
Day 8	Lesson 60: Capstone VIII: optimize a sensor pipeline under an information budget

Deeper reading. Pointers appear at the end of each lesson: LADR sections by number (e.g. 1A, 3C), Strang by chapter name (editions of Strang differ in page numbering), B&T by section number (e.g. B&T 1.3–1.5), and O&W by section number (e.g. O&W 4.3–4.4). Part VI's pointers go to the full Course 1 phase booklets (*DiivCourse1DSPMathFoundations*: Phase 4, its Radar & Communications supplement, Phases 5–6) and, for the bench lesson, to Scherz–Monk chapter/section. Part VII points, one book per lesson, to Ross (§), MIRA (chapter/section), Bak–Newman (chapter), Strichartz (§), Bowers–Kalton (chapter), and Trefethen–Bau (lecture). Part VIII points to Boyd–Vandenberghe by chapter and Polyanskiy–Wu by part/topic. Everything here is self-contained, though — the pointers are optional.

Notation. Following Axler, $\mathbb{F}$ denotes $\mathbb{R}$ or $\mathbb{C}$; in Part I you can always read $\mathbb{F} = \mathbb{R}$; Part II uses $\mathbb{F} = \mathbb{C}$ where it matters and says so. Vectors in $\mathbb{R}^n$ are written as columns; $x = (x_1, \ldots, x_n)$ is shorthand for the column vector with entries $x_1, \ldots, x_n$. In Parts III–IV, following B&T, $\mathbf{P}(A)$ is the probability of an event, random variables are capital letters $X, Y, \ldots$, p_X is a probability mass function, f_X a density, $\mathbf{E}[X]$ an expectation, and $\mathrm{var}(X)$, $\mathrm{cov}(X, Y)$ variance and covariance. In Part V, following O&W, $x(t)$ is a continuous-time and $x[n]$ a discrete-time

signal, δ and u are the unit impulse and unit step, h is an impulse response, and $X(j\omega)$, $X(e^{j\omega})$, $X(s)$, $X(z)$ are the CT Fourier, DT Fourier, Laplace, and z-transforms of x. Part V writes the imaginary unit as j, as every signals text does.

Contents

Part I: The Linear Algebra Core

1 Vectors and Linear Combinations

NEEDED FOR: Machine learning (core); assumed by everything later

1.1 Theory: the space $\mathbb{R}^n$

Definition 1.1 ($\mathbb{R}^n$)**.** For a positive integer n, $\mathbb{R}^n$ is the set of all lists of n real numbers:

$$\mathbb{R}^n = \{(x_1, \ldots, x_n) : x_k \in \mathbb{R} \text{ for } k = 1, \ldots, n\}.$$

Elements of $\mathbb{R}^n$ are called *vectors*; the number x_k is the kth *coordinate*.

Definition 1.2 (Addition and scalar multiplication)**.** For $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$ in $\mathbb{R}^n$ and $\lambda \in \mathbb{R}$:

$$x + y = (x_1 + y_1, \ldots, x_n + y_n), \qquad \lambda x = (\lambda x_1, \ldots, \lambda x_n).$$

These two operations obey the familiar rules of arithmetic. Axler abstracts these rules into the definition of a *vector space*: a set V with an addition and scalar multiplication satisfying commutativity, associativity, existence of an additive identity 0 and additive inverses, a multiplicative identity ($1v = v$), and the distributive properties. $\mathbb{R}^n$ is the prototypical example, and for machine learning, $\mathbb{R}^n$ is essentially the only vector space you need. But two quick payoffs of the abstract view are worth internalizing, because their proofs teach the style of reasoning used everywhere below.

Proposition 1.3 (Unique additive identity)**.** *A vector space has exactly one additive identity.*

Proof. Suppose 0 and $0'$ are both additive identities, meaning $v + 0 = v$ and $v + 0' = v$ for every v. Then

$$0' = 0' + 0 = 0 + 0' = 0,$$

where the first equality holds because 0 is an additive identity, the second by commutativity of addition, and the third because $0'$ is an additive identity. Hence $0' = 0$. □

Proposition 1.4 (0 times a vector)**.** *In any vector space, $0v = 0$ for every vector v. (On the left, 0 is the number zero; on the right, the zero vector.)*

Proof. Using the distributive property with $0 = 0 + 0$ in $\mathbb{R}$:

$$0v = (0 + 0)v = 0v + 0v.$$

Adding the additive inverse $-(0v)$ to both sides gives $0 = 0v$. □

> **Why prove obvious things?** These proofs use only the listed axioms, never a picture. That discipline is exactly what you need later when statements stop being obvious — e.g., "the set of solutions of $Ax = 0$ is a subspace" or "a projection minimizes distance." The habit: every step cites a definition or a previously proved fact.

Definition 1.5 (Linear combination)**.** A *linear combination* of vectors $v_1, \ldots, v_m \in \mathbb{R}^n$ is a vector of the form

$$a_1 v_1 + \cdots + a_m v_m, \qquad a_1, \ldots, a_m \in \mathbb{R}.$$

1.2 Practice: vectors in 2D and 3D

A vector $v = (2, 1)$ in $\mathbb{R}^2$ is drawn as an arrow from the origin to the point $(2, 1)$. Addition is the parallelogram rule: place the tail of w at the head of v; the sum $v + w$ is the arrow from the origin to the final head. Scalar multiplication stretches ($\lambda > 1$), shrinks ($0 < \lambda < 1$), or flips ($\lambda < 0$) the arrow.

Example 1.6 (All combinations of one or two vectors)**.** Let $v = (1, 2)$ and $w = (3, 1)$ in $\mathbb{R}^2$.

- The set of all multiples $\{\lambda v : \lambda \in \mathbb{R}\}$ is the line through the origin and $(1, 2)$.
- The set of all combinations $\{av+bw\}$ is all of $\mathbb{R}^2$: given any target (t_1, t_2), we need $a+3b = t_1$ and $2a+b = t_2$; solving gives $b = (2t_1 - t_2)/5$ and $a = t_1 - 3b$, which always works. (Lesson 3 explains why it always works: v and w are linearly independent.)

In $\mathbb{R}^3$, the combinations of two (non-parallel) vectors fill a *plane* through the origin — not all of $\mathbb{R}^3$. Try to reach $(0, 0, 1)$ using only $u = (1, 0, 0)$ and $v = (0, 1, 0)$: every combination $au + bv = (a, b, 0)$ has third coordinate 0, so you cannot. Combinations of a third vector pointing out of that plane, e.g. $w = (0, 0, 1)$, fill all of $\mathbb{R}^3$.

Example 1.7 (Hand computation)**.** With $u = (1, 0, -1)$, $v = (2, 1, 0)$, $w = (0, 1, 2)$:

$$2u - v + 3w = (2, 0, -2) - (2, 1, 0) + (0, 3, 6) = (0, 2, 4).$$

Notice $(0, 2, 4) = 2w + 0u + 0v$ as well — the same vector can arise from different combinations. Lesson 3 turns this observation into the definition of linear *dependence*: indeed $2u - v + w = 0$ here (check it!), so these three vectors are dependent.

> **Machine-learning connection.** A feature extractor ϕ maps an input (an email, a sentence, a game state) to a feature vector $\phi(x) \in \mathbb{R}^n$. Adding evidence and scaling importance are literally vector addition and scalar multiplication. A linear model's weight vector $w \in \mathbb{R}^n$ lives in the same space.

1.3 Exercises

Exercise 1.1 (Hand)**.** Let $v = (2, -1)$ and $w = (1, 3)$. Compute $3v+2w$, $v-w$, and find scalars a, b with $av + bw = (0, 7)$.

Exercise 1.2 (Hand)**.** Which vectors in $\mathbb{R}^3$ are linear combinations of $u = (1, 1, 0)$ and $v = (0, 1, 1)$? Describe the set geometrically, and exhibit one specific vector in $\mathbb{R}^3$ that is *not* a combination of u and v, with justification.

Exercise 1.3 (Proof, $\star$)**.** Prove that in any vector space, $(-1)v = -v$ for every v; that is, $(-1)v$ is the additive inverse of v. (Use Proposition 1.4 and the distributive property. This is LADR 1.B territory.)

Exercise 1.4 (Proof)**.** Prove that the additive inverse of each vector is unique (mirror the proof of the unique-additive-identity proposition).

Deeper reading. LADR 1A ($\mathbb{R}^n$ and $\mathbb{C}^n$), 1B (vector space axioms); Strang, ch. "Introduction to Vectors" (vectors and linear combinations).

2 Dot Products, Norms, and Orthogonality

NEEDED FOR: Machine learning (core)

2.1 Theory: inner products

Definition 2.1 (Dot product)**.** For $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ in $\mathbb{R}^n$, the *dot product* (Axler: *Euclidean inner product*) is

$$x \cdot y = \langle x,\, y \rangle = x_1 y_1 + \cdots + x_n y_n.$$

The dot product on $\mathbb{R}^n$ satisfies, for all $x, y, z \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ (each is a one-line check from the definition):

- *positivity*: $\langle x,\, x \rangle \geq 0$, with equality if and only if $x = 0$;
- *symmetry*: $\langle x,\, y \rangle = \langle y,\, x \rangle$;
- *linearity in each slot*: $\langle x + z,\, y \rangle = \langle x,\, y \rangle + \langle z,\, y \rangle$ and $\langle \lambda x,\, y \rangle = \lambda \langle x,\, y \rangle$.

Every proof in this lesson uses only these three properties — never the coordinate formula. That is the Axler discipline, and it means the results hold for *any* inner product.

Definition 2.2 (Norm, orthogonality)**.** The *norm* of x is $\|x\| = \sqrt{\langle x,\, x \rangle} = \sqrt{x_1^2 + \cdots + x_n^2}$. Vectors x, y are *orthogonal* if $\langle x,\, y \rangle = 0$.

Theorem 2.3 (Pythagorean theorem)**.** *If $x, y \in \mathbb{R}^n$ are orthogonal, then $\|x+y\|^2 = \|x\|^2 + \|y\|^2$.*

Proof. Expanding with linearity and symmetry,

$$\|x + y\|^2 = \langle x + y,\, x + y \rangle = \langle x,\, x \rangle + 2\langle x,\, y \rangle + \langle y,\, y \rangle = \|x\|^2 + 2\langle x,\, y \rangle + \|y\|^2,$$

and orthogonality makes the middle term vanish. □

The next lemma is the engine of Lessons 2 and 6: any vector can be split into a piece along y and a piece orthogonal to y.

Lemma 2.4 (Orthogonal decomposition)**.** *Let $y \neq 0$. For any $x \in \mathbb{R}^n$, set*

$$c = \frac{\langle x,\, y \rangle}{\|y\|^2} \quad \text{and} \quad z = x - cy.$$

Then $\langle z,\, y \rangle = 0$ and $x = cy + z$.

Proof. The identity $x = cy + z$ holds by the definition of z. For orthogonality,

$$\langle z,\, y \rangle = \langle x - cy,\, y \rangle = \langle x,\, y \rangle - c\,\langle y,\, y \rangle = \langle x,\, y \rangle - \frac{\langle x,\, y \rangle}{\|y\|^2}\,\|y\|^2 = 0.$$ □

The vector cy is the *projection of x onto the line spanned by y*.

Theorem 2.5 (Cauchy–Schwarz inequality)**.** *For all $x, y \in \mathbb{R}^n$,*

$$|\langle x,\, y \rangle| \leq \|x\|\,\|y\|,$$

with equality if and only if one of x, y is a scalar multiple of the other.

Proof. If $y = 0$ both sides are 0. Otherwise decompose $x = cy + z$ as in Lemma 2.4. Since cy and z are orthogonal, the Pythagorean theorem gives

$$\|x\|^2 = \|cy\|^2 + \|z\|^2 = \frac{\langle x,\, y\rangle^2}{\|y\|^2} + \|z\|^2 \geq \frac{\langle x,\, y\rangle^2}{\|y\|^2},$$

because $\|z\|^2 \geq 0$. Multiplying by $\|y\|^2$ and taking square roots yields the inequality. Equality holds exactly when $\|z\| = 0$, i.e. $z = 0$, i.e. $x = cy$ is a multiple of y (and if x is a multiple of y, direct computation gives equality). □

Theorem 2.6 (Triangle inequality)**.** $\|x + y\| \leq \|x\| + \|y\|$ *for all* $x, y \in \mathbb{R}^n$.

Proof. Using Cauchy–Schwarz in the middle step,

$$\|x + y\|^2 = \|x\|^2 + 2\langle x,\, y\rangle + \|y\|^2 \leq \|x\|^2 + 2\|x\|\|y\| + \|y\|^2 = (\|x\| + \|y\|)^2. \qquad \square$$

By Cauchy–Schwarz, for nonzero x, y the ratio $\langle x,\, y\rangle/(\|x\|\|y\|)$ lies in $[-1, 1]$, so it is the cosine of a unique angle $\theta \in [0, \pi]$:

$$\cos\theta = \frac{\langle x,\, y\rangle}{\|x\|\,\|y\|}.$$

In $\mathbb{R}^2$ and $\mathbb{R}^3$ this matches the geometric angle between the arrows; in higher dimensions it is the *definition* of angle, and Cauchy–Schwarz is precisely what makes the definition legal.

2.2 Practice: 2D/3D computations

Example 2.7. Let $x = (3, 4)$ and $y = (4, -3)$. Then $\langle x,\, y\rangle = 12 - 12 = 0$: orthogonal. Norms: $\|x\| = \sqrt{9 + 16} = 5 = \|y\|$. Pythagoras check: $x + y = (7, 1)$, and $\|x + y\|^2 = 50 = 25 + 25$. ✓

Example 2.8 (Projection by hand)**.** Project $x = (2, 3)$ onto the line spanned by $y = (1, 1)$:

$$c = \frac{\langle x,\, y\rangle}{\|y\|^2} = \frac{2 + 3}{2} = \tfrac{5}{2}, \qquad cy = \left(\tfrac{5}{2}, \tfrac{5}{2}\right), \qquad z = x - cy = \left(-\tfrac{1}{2}, \tfrac{1}{2}\right).$$

Check: $\langle z,\, y\rangle = -\frac{1}{2} + \frac{1}{2} = 0$. ✓ The point $(\frac{5}{2}, \frac{5}{2})$ is the closest point to $(2, 3)$ on the line $y = x$ — Lesson 6 proves that projections are always the closest points.

Example 2.9 (Angles)**.** For $u = (1, 0, 1)$ and $v = (1, 1, 0)$: $\langle u,\, v\rangle = 1$, $\|u\| = \|v\| = \sqrt{2}$, so $\cos\theta = 1/2$ and $\theta = 60°$.

Machine-learning connection. A linear classifier predicts with the *score* $w \cdot \phi(x)$: positive score, predict $+1$; negative, predict -1. The decision boundary is the set $\{v : w \cdot v = 0\}$ — the hyperplane orthogonal to w. The *margin* of an example involves $\|w\|$, and cosine similarity between feature vectors is exactly $\cos\theta$ above. You will compute dot products constantly.

2.3 Exercises

Exercise 2.1 (Hand)**.** For $x = (1, 2, 2)$ and $y = (2, -2, 1)$: compute $\langle x,\, y\rangle$, $\|x\|$, $\|y\|$, and $\cos\theta$. Are x and y orthogonal?

Exercise 2.2 (Hand)**.** Project $x = (4, 1)$ onto $y = (2, 1)$; write $x = cy + z$ and verify $\langle z,\, y\rangle = 0$. Then compute $\|cy\|^2 + \|z\|^2$ and confirm it equals $\|x\|^2$.

Exercise 2.3 (Proof, $\star$)**.** (Parallelogram equality, LADR 6A) Prove that for all $x, y \in \mathbb{R}^n$,

$$\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2,$$

using only the three inner-product properties. Interpret the identity for the parallelogram with sides x and y.

Exercise 2.4 (Proof)**.** Show that if $\langle x,\, v\rangle = 0$ for *every* $v \in \mathbb{R}^n$, then $x = 0$. (Hint: choose v wisely.)

Deeper reading. LADR 6A (inner products and norms — the proofs above follow Axler's); Strang, ch. "Introduction to Vectors" (lengths and dot products).

3 Span, Linear Independence, Basis, Dimension

NEEDED FOR: Machine learning (core)

3.1 Theory

Definition 3.1 (Span)**.** The *span* of $v_1, \ldots, v_m \in \mathbb{R}^n$ is the set of all their linear combinations:

$$\operatorname{span}(v_1, \ldots, v_m) = \{a_1 v_1 + \cdots + a_m v_m : a_1, \ldots, a_m \in \mathbb{R}\}.$$

The span of the empty list is defined to be $\{0\}$. If $\operatorname{span}(v_1, \ldots, v_m) = V$, we say the list *spans* V.

A *subspace* of $\mathbb{R}^n$ is a subset containing 0 that is closed under addition and scalar multiplication. Spans are exactly the subspaces you can describe by a finite list: $\operatorname{span}(v_1, \ldots, v_m)$ is the smallest subspace containing $v_1, \ldots, v_m$ (Exercise 3.4).

Definition 3.2 (Linear independence)**.** A list $v_1, \ldots, v_m$ is *linearly independent* if the only choice of scalars with

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

is $a_1 = \cdots = a_m = 0$. Otherwise the list is *linearly dependent*.

Equivalently (and this is the useful reading): independence means every vector in the span has *exactly one* representation as a combination. For if $a_1 v_1 + \cdots + a_m v_m = b_1 v_1 + \cdots + b_m v_m$, subtracting gives $\sum_k (a_k - b_k) v_k = 0$, so independence forces $a_k = b_k$ for all k.

Lemma 3.3 (Linear dependence lemma, LADR 2.19)**.** *If $v_1, \ldots, v_m$ is linearly dependent and $v_1 \neq 0$, then there exists an index $k \in \{2, \ldots, m\}$ such that $v_k \in \operatorname{span}(v_1, \ldots, v_{k-1})$; moreover, removing v_k from the list leaves the span unchanged.*

Proof. By dependence there are scalars $a_1, \ldots, a_m$, not all zero, with $a_1 v_1 + \cdots + a_m v_m = 0$. Let k be the *largest* index with $a_k \neq 0$. If $k = 1$ we would have $a_1 v_1 = 0$ with $a_1 \neq 0$, hence $v_1 = 0$, contradicting our hypothesis; so $k \geq 2$. Solving for v_k:

$$v_k = -\frac{a_1}{a_k} v_1 - \cdots - \frac{a_{k-1}}{a_k} v_{k-1} \in \operatorname{span}(v_1, \ldots, v_{k-1}).$$

For the second claim: any combination using v_k can substitute this expression for v_k, producing the same vector as a combination of the others. Hence the span without v_k contains the span with it; the reverse containment is clear. □

Theorem 3.4 (Independent lists are no longer than spanning lists, LADR 2.22)**.** *In $\mathbb{R}^n$, if $u_1, \ldots, u_p$ is linearly independent and $w_1, \ldots, w_q$ spans $\mathbb{R}^n$ (or any subspace containing the u's), then $p \leq q$.*

Proof. We show that we can trade the w's for the u's one at a time, keeping a spanning list of length q at every step.

Step 1. The list $u_1, w_1, \ldots, w_q$ is linearly dependent (because u_1 is already in the span of the w's, it has a nontrivial representation, giving a dependence relation) and its first vector $u_1 \neq 0$ (independent lists cannot contain 0). By Lemma 3.3 we may remove some w_j — the lemma yields an index $k \geq 2$ whose vector lies in the span of its predecessors, and that vector must be a w, not a u, since the u's alone will always be independent — leaving a spanning list of length q containing u_1.

Step j. Suppose after $j - 1$ steps we have a spanning list of length q containing $u_1, \ldots, u_{j-1}$ (listed first) and $q - j + 1$ of the w's. Adjoin u_j right after $u_1, \ldots, u_{j-1}$: the resulting list of length $q + 1$ is dependent (u_j is in the span of a spanning list). By Lemma 3.3, some vector is in the span of its predecessors; it cannot be one of $u_1, \ldots, u_j$, since those are independent. So it is one of the remaining w's, and removing it keeps a spanning list of length q.

For this process to run through step p, there must be a w available to remove at each step; hence $q \geq p$. □

Definition 3.5 (Basis)**.** A *basis* of a subspace V is a list of vectors in V that is linearly independent and spans V.

Theorem 3.6 (Dimension is well defined, LADR 2.34)**.** *Any two bases of a subspace V have the same length. This common length is called the* dimension *of V, written* $\dim V$.

Proof. Let B_1 (length p) and B_2 (length q) be bases of V. Since B_1 is independent and B_2 spans V, Theorem 3.4 gives $p \leq q$. Swapping roles gives $q \leq p$. Hence $p = q$. □

Theorem 3.7 (Criterion for a basis)**.** *The list $v_1, \ldots, v_m$ is a basis of V if and only if every $v \in V$ can be written in* exactly one *way as $v = a_1 v_1 + \cdots + a_m v_m$.*

Proof. ($\Rightarrow$) Spanning gives at least one representation; independence gives at most one, by the uniqueness argument following the definition of independence. ($\Leftarrow$) Existence of representations for every v is spanning. Uniqueness applied to $v = 0$ says the only combination equal to 0 is the all-zeros one — independence. □

The *standard basis* of $\mathbb{R}^n$ is $e_1 = (1, 0, \ldots, 0), \ldots, e_n = (0, \ldots, 0, 1)$; hence $\dim \mathbb{R}^n = n$, and now "dimension" finally has a theorem-backed meaning matching the intuition: a line through 0 has dimension 1, a plane through 0 has dimension 2.

Why this matters. Theorem 3.7 is the licence to use *coordinates*: once you fix a basis, every vector *is* a unique list of numbers. The dimension theorem says "number of degrees of freedom" is intrinsic — it does not depend on which basis you happened to pick. These two facts are silently invoked every time machine learning says "represent the state as a vector in $\mathbb{R}^d$."

3.2 Practice: 2D/3D

Example 3.8 (Checking independence by hand)**.** Are $v_1 = (1, 2, 0)$, $v_2 = (0, 1, 1)$, $v_3 = (1, 4, 2)$ independent? Solve $av_1 + bv_2 + cv_3 = 0$:

$$\begin{aligned} a + c &= 0 \\ 2a + b + 4c &= 0 \implies a = -c, \quad b = -2c, \quad \text{2nd eq: } -2c - 2c + 4c = 0. \checkmark \\ b + 2c &= 0 \end{aligned}$$

The second equation is automatic, so c is free: taking $c = 1$ gives $-v_1 - 2v_2 + v_3 = 0$. *Dependent*, and indeed $v_3 = v_1 + 2v_2$: the third vector lies in the plane spanned by the first two, exactly as the linear dependence lemma predicts.

Example 3.9 (A basis of a plane)**.** The plane $P = \{(x_1, x_2, x_3) : x_1 + x_2 + x_3 = 0\}$ in $\mathbb{R}^3$ is a subspace (check closure!). The vectors $u = (1, -1, 0)$ and $v = (1, 0, -1)$ lie in P, are independent (neither is a multiple of the other), and span P: given (x_1, x_2, x_3) with $x_1 + x_2 + x_3 = 0$, write it as $-x_2u - x_3v$ — check: $-x_2(1, -1, 0) - x_3(1, 0, -1) = (-x_2 - x_3, x_2, x_3) = (x_1, x_2, x_3)$, using $x_1 = -x_2 - x_3$. So $\dim P = 2$: a plane, as geometry demands.

Machine-learning connection. The set of functions a linear model can represent is the span of its features: with features $\phi_1, \ldots, \phi_d$, the hypothesis class is $\{\sum_k w_k \phi_k\}$. Redundant (dependent) features do not enlarge the span — they add parameters without adding expressive power. "Add a feature to make the model more expressive" means "add a vector outside the current span."

3.3 Exercises

Exercise 3.1 (Hand)**.** Determine whether each list is independent; if dependent, exhibit a dependence relation. (a) $(2, 1), (4, 3)$ in $\mathbb{R}^2$. (b) $(1, 1, 1), (1, 2, 3), (0, 1, 2)$ in $\mathbb{R}^3$. (c) any list containing the zero vector.

Exercise 3.2 (Hand)**.** Find a basis for the line $\{(t, 2t, -t) : t \in \mathbb{R}\}$ and for the plane $x_1 - x_3 = 0$ in $\mathbb{R}^3$. State the dimension of each.

Exercise 3.3 (Proof, $\star$)**.** Prove that any list of $n + 1$ vectors in $\mathbb{R}^n$ is linearly dependent. (One line, given a theorem from this lesson.) Conclude that at most n features on a dataset of n-dimensional inputs can be "genuinely new" in the sense of enlarging the span at every step.

Exercise 3.4 (Proof)**.** Prove that $\operatorname{span}(v_1, \ldots, v_m)$ is a subspace, and that any subspace containing $v_1, \ldots, v_m$ contains $\operatorname{span}(v_1, \ldots, v_m)$.

Deeper reading. LADR 2A (span and linear independence), 2B (bases), 2C (dimension); Strang, ch. "Vector Spaces and Subspaces" (independence, basis and dimension).

4 Matrices and Linear Maps

NEEDED FOR: Machine learning (core)

4.1 Theory: linear maps first

Definition 4.1 (Linear map)**.** A function $T : \mathbb{R}^n \to \mathbb{R}^m$ is a *linear map* if for all $u, v \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$:

$$T(u+v) = Tu + Tv \qquad \text{and} \qquad T(\lambda v) = \lambda\, Tv.$$

Rotations about the origin, reflections through lines through the origin, scalings, shears, and projections onto subspaces are linear. Translation $v \mapsto v + b$ (for $b \neq 0$) is *not*: it moves 0.

Proposition 4.2. *If T is linear, then $T(0) = 0$.*

Proof. $T(0) = T(0+0) = T(0) + T(0)$; add $-T(0)$ to both sides. □

Theorem 4.3 (A linear map is determined by its values on a basis, LADR 3.4)**.** *Let $v_1, \dots, v_n$ be a basis of $\mathbb{R}^n$ and $w_1, \dots, w_n$ any vectors in $\mathbb{R}^m$. Then there exists a unique linear map $T : \mathbb{R}^n \to \mathbb{R}^m$ with $Tv_k = w_k$ for each k.*

Proof. Existence. Every $x \in \mathbb{R}^n$ has a unique representation $x = a_1 v_1 + \cdots + a_n v_n$ (Theorem 3.7); define $Tx = a_1 w_1 + \cdots + a_n w_n$. This is well defined precisely because the representation is unique. Linearity: if $x = \sum a_k v_k$ and $y = \sum b_k v_k$, then $x + y = \sum (a_k + b_k) v_k$ is *the* representation of $x + y$, so $T(x+y) = \sum (a_k + b_k) w_k = Tx + Ty$; homogeneity is similar. Also $Tv_k = w_k$ by construction (take $a_k = 1$, others 0).

Uniqueness. If S is linear with $Sv_k = w_k$, then for $x = \sum a_k v_k$, linearity forces $Sx = \sum a_k S v_k = \sum a_k w_k = Tx$. □

> **The point.** A linear map on $\mathbb{R}^n$ is pinned down by n vectors' worth of information — its values on a basis. Choosing the standard basis, those values are the *columns of a matrix*. This theorem is why matrices and linear maps are interchangeable.

Definition 4.4 (Matrix of a map; matrix–vector product)**.** For linear $T : \mathbb{R}^n \to \mathbb{R}^m$, the *matrix of T* (with respect to standard bases) is the $m \times n$ array A whose kth column is $Te_k \in \mathbb{R}^m$. For $x = (x_1, \dots, x_n)$ we then have, by linearity,

$$Tx = T\Big(\sum_k x_k e_k\Big) = \sum_k x_k\, (Te_k),$$

which we *define* to be the product Ax. In words:

$$Ax = \text{the linear combination of the } \textit{columns} \text{ of } A \text{ with coefficients } x_1, \dots, x_n.$$

Entrywise, $(Ax)_i = \sum_k A_{i,k} x_k$: row i of A dotted with x. Both readings are used daily; the column reading is the more illuminating one.

Definition 4.5 (Matrix multiplication)**.** For linear maps $S : \mathbb{R}^n \to \mathbb{R}^m$ and $T : \mathbb{R}^p \to \mathbb{R}^n$, the composition $S \circ T : \mathbb{R}^p \to \mathbb{R}^m$ is linear (check!). The product AB of the matrix A of S ($m \times n$)

and the matrix B of T ($n \times p$) is *defined* as the matrix of $S \circ T$; it is $m \times p$. Column k of AB is $(S \circ T)e_k = A(Be_k) = A\,(\text{column } k \text{ of } B)$, which yields the familiar formula

$$(AB)_{i,j} = \sum_{k=1}^{n} A_{i,k} B_{k,j}.$$

Because composition of functions is associative but not commutative, matrix multiplication is associative — $(AB)C = A(BC)$, no computation needed — and in general $AB \neq BA$.

Definition 4.6 (Transpose)**.** The *transpose* of an $m \times n$ matrix A is the $n \times m$ matrix A^{T} with $(A^{\mathsf{T}})_{i,j} = A_{j,i}$. Two facts used constantly: $(AB)^{\mathsf{T}} = B^{\mathsf{T}}A^{\mathsf{T}}$, and the dot product is a matrix product: $x \cdot y = x^{\mathsf{T}}y$. The defining property tying transpose to the dot product is

$$\langle Ax,\, y\rangle = \langle x,\, A^{\mathsf{T}}y\rangle \qquad \text{for all } x \in \mathbb{R}^n,\ y \in \mathbb{R}^m,$$

which follows by expanding both sides as $\sum_{i,k} A_{i,k}x_k y_i$.

4.2 Practice: 2D matrices you can see

Example 4.7 (Reading a matrix by its columns)**.**

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}: \qquad Ae_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad Ae_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

$e_1 \mapsto e_2$ and $e_2 \mapsto -e_1$: this is rotation by 90° counterclockwise. In general, rotation by angle θ has matrix $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ — its columns are where e_1, e_2 land, which is all Theorem 4.3 says you need to know.

Example 4.8 (Hand computation, both readings)**.**

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{pmatrix}, \qquad x = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

Column reading: $Ax = 2\begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix} - 1\begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix}$. Row reading: $(Ax)_1 = 1 \cdot 2 + 2 \cdot (-1) = 0$, $(Ax)_2 = 6 - 4 = 2$, $(Ax)_3 = 0 - 1 = -1$. Same answer, as it must be.

Example 4.9 (Composition order matters)**.** Let R = rotation by 90° (above) and $S = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$ = projection onto the x_1-axis. Then

$$SR = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}, \qquad RS = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

Track e_1 through each composition (rightmost map acts first): $SR\,e_1 = S(Re_1) = S(e_2) = 0$, while $RS\,e_1 = R(Se_1) = R(e_1) = e_2$. Rotate-then-project kills e_1; project-then-rotate sends it to e_2. Different maps, different matrices: $AB \neq BA$.

Machine-learning connection. A dataset of m examples with n features is an $m \times n$ matrix X whose *rows* are feature vectors. Then Xw computes all m scores $w \cdot \phi(x_i)$ in one matrix–vector product. A tiny neural network layer is $x \mapsto \sigma(Wx + b)$: a linear map, a shift, a nonlinearity. Reading Ax fluently in both the row picture and the column picture is the single highest-value matrix skill for the course.

4.3 Exercises

Exercise 4.1 (Hand)**.** Write down the 2×2 matrices of: (a) reflection through the line $x_2 = x_1$; (b) scaling by 3 in the x_1 direction only; (c) rotation by 180°. (Use the columns-are-images-of-e_k principle; no formulas to memorize.)

Exercise 4.2 (Hand)**.** With $A = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ (a *shear*) and $B = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, compute AB, BA, and $A^{\mathsf{T}}B^{\mathsf{T}}$. Check $(BA)^{\mathsf{T}} = A^{\mathsf{T}}B^{\mathsf{T}}$.

Exercise 4.3 (Proof, $\star$)**.** Prove that the composition of linear maps is linear, and that $\langle Ax,\, y\rangle = \langle x,\, A^{\mathsf{T}}y\rangle$ for all x, y by expanding both sides in coordinates.

Exercise 4.4 (Proof)**.** Using Theorem 4.3, prove: if two linear maps $S, T : \mathbb{R}^n \to \mathbb{R}^m$ agree on a basis, they are equal. Then explain in one sentence why an $m \times n$ matrix and a linear map $\mathbb{R}^n \to \mathbb{R}^m$ carry exactly the same information.

Deeper reading. LADR 3A (linear maps), 3C (matrices, matrix multiplication); Strang, chs. "Multiplying Matrices" / "Solving Linear Equations" (the idea of elimination, matrix operations).

5 Null Space, Rank, and Solving $Ax = b$

NEEDED FOR: Machine learning (core)

5.1 Theory: the fundamental theorem

Throughout, $T : \mathbb{R}^n \to \mathbb{R}^m$ is linear with matrix A.

Definition 5.1 (Null space and range)**.**

$$\operatorname{null} T = \{x \in \mathbb{R}^n : Tx = 0\}, \qquad \operatorname{range} T = \{Tx : x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m.$$

For the matrix A, $\operatorname{range} T$ is the span of the columns of A (by the column reading of Ax), also called the *column space*; $\operatorname{null} T$ is the solution set of $Ax = 0$. Both are subspaces (Exercise 5.4). The *rank* of A is $\operatorname{rank} A = \dim \operatorname{range} T$.

Proposition 5.2 (Injectivity ⇔ trivial null space, LADR 3.15)**.** *T is injective (one-to-one) if and only if* $\operatorname{null} T = \{0\}$.

Proof. ($\Rightarrow$) $T0 = 0$; if $Tx = 0$ too, injectivity gives $x = 0$. ($\Leftarrow$) If $Tu = Tv$, linearity gives $T(u - v) = 0$, so $u - v \in \operatorname{null} T = \{0\}$, i.e. $u = v$. □

This tiny proposition is used constantly: to check whether a linear map ever collapses two inputs together, you only have to look at what it sends to zero.

Theorem 5.3 (Fundamental theorem of linear maps / rank–nullity, LADR 3.21)**.** *For linear* $T : \mathbb{R}^n \to \mathbb{R}^m$*,*

$$\dim \operatorname{null} T + \dim \operatorname{range} T = n.$$

Proof. Let $u_1, \dots, u_k$ be a basis of $\operatorname{null} T$ (so $\dim \operatorname{null} T = k$). Extend it to a basis $u_1, \dots, u_k, v_1, \dots, v_r$ of $\mathbb{R}^n$ (add vectors outside the current span one at a time until spanning; independence is preserved at each step by the linear dependence lemma). Then $n = k + r$, and it suffices to show $Tv_1, \dots, Tv_r$ is a basis of $\operatorname{range} T$.

Spanning: any $x \in \mathbb{R}^n$ is $x = \sum a_i u_i + \sum b_j v_j$, so

$$Tx = \sum a_i \underbrace{Tu_i}_{=0} + \sum b_j Tv_j = \sum b_j Tv_j,$$

hence every vector of $\operatorname{range} T$ is a combination of the Tv_j.

Independence: suppose $\sum_j c_j Tv_j = 0$. By linearity $T\big(\sum_j c_j v_j\big) = 0$, so $\sum_j c_j v_j \in \operatorname{null} T$, hence $\sum_j c_j v_j = \sum_i d_i u_i$ for some scalars d_i. Rearranged, $\sum_j c_j v_j - \sum_i d_i u_i = 0$ is a linear relation on the basis of $\mathbb{R}^n$, so all c_j (and d_i) are zero. □

Corollary 5.4 (The square case is all-or-nothing)**.** *For* $T : \mathbb{R}^n \to \mathbb{R}^n$*: injective* $\Leftrightarrow$ *surjective* $\Leftrightarrow$ *invertible.*

Proof. By Theorem 5.3, $\dim \operatorname{null} T = 0 \iff \dim \operatorname{range} T = n \iff \operatorname{range} T = \mathbb{R}^n$ (a subspace of $\mathbb{R}^n$ of dimension n must be $\mathbb{R}^n$: take a basis of the subspace; it is an independent list of length n in $\mathbb{R}^n$, and extending it to a basis of $\mathbb{R}^n$ cannot add anything by Theorem 3.4). So injective $\iff$ surjective, and a bijective linear map has a linear inverse (Exercise 5.5). A square matrix A is *invertible* when such A^{-1} exists: $A^{-1}A = AA^{-1} = I$. □

What the fundamental theorem says. An $m \times n$ matrix consumes n input dimensions. Each dimension either survives into the output (contributing to rank) or is crushed to zero (contributing to the null space) — and nothing is lost or created: crushed + surviving $= n$. Consequences you can now read off instantly: a map to a smaller space ($m < n$) must crush something ($\dim \operatorname{null} T \geq n - m > 0$), so it cannot be injective; a map from a smaller space ($n < m$) has rank at most $n < m$, so it cannot be surjective.

Solving $Ax = b$**.** The complete picture, now provable in three lines:

1. $Ax = b$ has a solution $\iff$ $b \in \operatorname{range} T$ (that is what range means).
2. If x_p is one particular solution, the full solution set is $x_p + \operatorname{null} T = \{x_p + h : Ah = 0\}$: for if $Ax = b = Ax_p$ then $A(x - x_p) = 0$; conversely $A(x_p + h) = b + 0 = b$.
3. Hence the solution is unique iff $\operatorname{null} T = \{0\}$; for square A, Corollary 5.4 says a solution then exists for every b, namely $x = A^{-1}b$.

5.2 Practice: elimination by hand

Gaussian elimination is Strang's centerpiece: subtract multiples of rows to reach triangular form, then back-substitute. Row operations do not change the solution set (each is reversible and preserves the equations).

Example 5.5 (A unique solution)**.** Solve $Ax = b$:

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 4 \\ 1 & 1 & 0 \end{pmatrix} x = \begin{pmatrix} 3 \\ 8 \\ 1 \end{pmatrix}.$$

$R_2 \leftarrow R_2 - 2R_1$ and $R_3 \leftarrow R_3 - R_1$:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & -1 & -1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 2 \\ -2 \end{pmatrix}; \qquad R_3 \leftarrow R_3 + R_2: \quad \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 2 \\ 0 \end{pmatrix}.$$

Back-substitute: $x_3 = 0$, then $x_2 = 2 - 2x_3 = 2$, then $x_1 = 3 - 2x_2 - x_3 = -1$. Solution $x = (-1, 2, 0)$; three pivots, so rank 3, null space $\{0\}$, solution unique — the theory and the algorithm agree.

Example 5.6 (Null space by hand)**.** Find $\operatorname{null} A$ for $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix}$. The second row is twice the first, so $Ax = 0$ reduces to the single equation $x_1 + 2x_2 + 3x_3 = 0$: solve $x_1 = -2x_2 - 3x_3$ with x_2, x_3 free. Null space $= \operatorname{span}\big((-2, 1, 0), (-3, 0, 1)\big)$, dimension 2. Rank check: the columns of A are all multiples of $(1, 2)$, so rank $= 1$, and $1 + 2 = 3 = n$. ✓

> **Machine-learning connection.** "Does a weight vector exist that fits these constraints, and is it unique?" is exactly existence/uniqueness for a linear system. A nontrivial null space of the feature matrix means different weight vectors produce identical predictions on the training set — the model is over-parameterized on that data. Rank tells you the number of effective directions your data spans.

5.3 Exercises

Exercise 5.1 (Hand)**.** Solve by elimination, tracking pivots: $\left(\begin{smallmatrix} 2 & 1 \\ 4 & 3 \end{smallmatrix}\right) x = \left(\begin{smallmatrix} 5 \\ 11 \end{smallmatrix}\right)$, then $\left(\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right) x = \left(\begin{smallmatrix} 3 \\ 6 \end{smallmatrix}\right)$, then the same left side with right side $(3, 7)$. Describe all solutions in each case (unique / infinitely many / none) and reconcile with rank and the three-step picture above.

Exercise 5.2 (Hand)**.** For $A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{pmatrix}$, find a basis of $\operatorname{null} A$ and a basis of the column space; verify rank–nullity.

Exercise 5.3 (Proof, $\star$)**.** Prove Proposition 5.2's consequence for systems: $Ax = b$ has *at most one* solution for every b if and only if $Ax = 0$ has only the zero solution.

Exercise 5.4 (Proof)**.** Prove that $\operatorname{null} T$ and $\operatorname{range} T$ are subspaces.

Exercise 5.5 (Proof)**.** Suppose $T : \mathbb{R}^n \to \mathbb{R}^n$ is linear and bijective. Prove that its inverse function T^{-1} is linear. (Apply T to both sides of what you want to show.)

Deeper reading. LADR 3B (null spaces and ranges, fundamental theorem), 3D (invertibility); Strang, chs. "Solving Linear Equations" (elimination) and "Vector Spaces and Subspaces" (the null space, rank).

6 Orthogonal Projections and Least Squares

NEEDED FOR: Machine learning (core); reused in Lesson 22 for MMSE estimation

6.1 Theory

Definition 6.1 (Orthonormal list; orthogonal complement)**.** A list $e_1, \dots, e_k$ is *orthonormal* if $\langle e_i, e_j \rangle = 0$ for $i \neq j$ and $\|e_i\| = 1$ for all i. For a subspace $U \subseteq \mathbb{R}^n$, the *orthogonal complement* is

$$U^\perp = \{x \in \mathbb{R}^n : \langle x, u \rangle = 0 \text{ for all } u \in U\}.$$

Proposition 6.2 (Orthonormal lists are independent, LADR 6.25)**.** *Every orthonormal list is linearly independent, and if $v = a_1 e_1 + \cdots + a_k e_k$ then $a_j = \langle v, e_j \rangle$.*

Proof. Take the inner product of $v = \sum_i a_i e_i$ with e_j: by linearity, $\langle v, e_j \rangle = \sum_i a_i \langle e_i, e_j \rangle = a_j$, since all terms vanish except $i = j$, where $\langle e_j, e_j \rangle = 1$. For independence, apply this with $v = 0$: every coefficient equals $\langle 0, e_j \rangle = 0$. □

So orthonormal bases make coordinates *free*: no system-solving, just k dot products.

Theorem 6.3 (Orthogonal projection)**.** *Let $e_1, \dots, e_k$ be an orthonormal basis of a subspace $U \subseteq \mathbb{R}^n$ (one always exists: apply the Gram–Schmidt procedure below to any basis). Define $P_U : \mathbb{R}^n \to \mathbb{R}^n$ by*

$$P_U x = \langle x, e_1 \rangle \, e_1 + \cdots + \langle x, e_k \rangle \, e_k.$$

Then for every x: (a) $P_U x \in U$ and $x - P_U x \in U^\perp$; (b) $P_U x$ is the unique closest point of U to x:

$$\|x - P_U x\| < \|x - u\| \qquad \text{for every } u \in U \text{ with } u \neq P_U x.$$

Proof. (a) $P_U x \in U$ since it is a combination of the e_i. For the second part, for each j,

$$\langle x - P_U x, e_j \rangle = \langle x, e_j \rangle - \sum_i \langle x, e_i \rangle \langle e_i, e_j \rangle = \langle x, e_j \rangle - \langle x, e_j \rangle = 0,$$

and a vector orthogonal to each e_j is orthogonal to every combination of them, i.e. to all of U.

(b) Let $u \in U$. Write

$$x - u = \underbrace{(x - P_U x)}_{\in U^\perp} + \underbrace{(P_U x - u)}_{\in U}.$$

The two pieces are orthogonal, so by the Pythagorean theorem

$$\|x - u\|^2 = \|x - P_U x\|^2 + \|P_U x - u\|^2 \geq \|x - P_U x\|^2,$$

with equality iff $\|P_U x - u\| = 0$, i.e. $u = P_U x$. □

One picture to keep. Drop a perpendicular from x to the subspace U; its foot is $P_U x$. The proof of (b) is the Pythagorean theorem applied to the right triangle with vertices x, $P_U x$, and any other candidate u. Every least-squares fact below is this picture.

Gram–Schmidt in brief (LADR 6.32). Given an independent list $v_1, \dots, v_k$, define $e_1 = v_1/\|v_1\|$ and, inductively, subtract from v_j its projection onto the span of the previous e's, then normalize:

$$f_j = v_j - \sum_{i<j} \langle v_j,\, e_i \rangle\, e_i, \qquad e_j = f_j/\|f_j\|.$$

Each $f_j \neq 0$ (else v_j would be in the span of its predecessors, contradicting independence), each e_j is orthogonal to the previous ones by the same computation as in Theorem 6.3(a), and the spans agree at every stage.

Theorem 6.4 (Least squares / normal equations)**.** *Let A be $m \times n$ with linearly independent columns, and $b \in \mathbb{R}^m$. Then the minimization problem*

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2$$

has the unique solution given by the normal equations

$$A^{\mathsf{T}} A\, \hat{x} = A^{\mathsf{T}} b.$$

Proof. As x ranges over $\mathbb{R}^n$, the vector Ax ranges exactly over $U = \operatorname{range} A$ (the column space). By Theorem 6.3(b), the closest point of U to b is the unique vector $P_U b$; so minimizers are exactly the solutions of $A\hat{x} = P_U b$. Independence of the columns means A is injective on $\mathbb{R}^n$ (Proposition 5.2 applied to combinations of columns), so exactly one $\hat{x}$ satisfies this.

It remains to convert "$A\hat{x}$ is the projection of b" into equations. By Theorem 6.3(a), $A\hat{x} = P_U b$ iff the residual $b - A\hat{x}$ is orthogonal to U, i.e. orthogonal to every column of A, i.e.

$$A^{\mathsf{T}}(b - A\hat{x}) = 0 \iff A^{\mathsf{T}} A\, \hat{x} = A^{\mathsf{T}} b.$$

(For the converse direction of the iff: if $A^{\mathsf{T}}(b - A\hat{x}) = 0$, then $b - A\hat{x}$ is orthogonal to each column of A, hence to all of U, and adding $A\hat{x} \in U$ shows $A\hat{x}$ is the orthogonal projection of b, by the uniqueness of the decomposition $b = P_U b + (b - P_U b)$ into a U-part plus a $U^{\perp}$-part.)

Finally, $A^{\mathsf{T}} A$ is invertible when A has independent columns: if $A^{\mathsf{T}} A x = 0$ then $0 = \langle A^{\mathsf{T}} A x,\, x \rangle = \langle Ax,\, Ax \rangle = \|Ax\|^2$, so $Ax = 0$, so $x = 0$ by injectivity; now apply Corollary 5.4 to the square matrix $A^{\mathsf{T}} A$. □

6.2 Practice: fitting a line by hand

Example 6.5 (Least-squares line through three points)**.** Fit $y \approx c + dt$ to the data points $(t, y) = (0, 1), (1, 2), (2, 4)$. In matrix form, we want $Ax \approx b$ with unknowns $x = (c, d)$:

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}.$$

No exact solution exists (the three points are not collinear — check: slopes 1 and 2). Normal equations:

$$A^{\mathsf{T}}A = \begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix}, \quad A^{\mathsf{T}}b = \begin{pmatrix} 7 \\ 10 \end{pmatrix} \implies \begin{cases} 3c + 3d = 7 \\ 3c + 5d = 10 \end{cases} \implies d = \tfrac{3}{2},\ c = \tfrac{7-3d}{3} = \tfrac{5}{6}.$$

Best line: $y = \frac{5}{6} + \frac{3}{2}t$. Residual: $b - A\hat{x} = (1, 2, 4) - (\frac{5}{6}, \frac{7}{3}, \frac{23}{6}) = (\frac{1}{6}, -\frac{1}{3}, \frac{1}{6})$. Sanity checks: the residual sums to zero (orthogonal to the all-ones first column) and $\langle \text{residual}, (0, 1, 2) \rangle = 0 - \frac{1}{3} + \frac{1}{3} = 0$ (orthogonal to the second column). The residual is orthogonal to the column space — the picture made numerical.

Machine-learning connection. Linear regression with the squared loss *is* this theorem: training loss $= \frac{1}{m}\|Xw - y\|^2$, and the closed-form minimizer solves $X^{\mathsf{T}}Xw = X^{\mathsf{T}}y$. Machine learning will usually minimize by gradient descent instead (Lesson 7) — but knowing the geometry (prediction = projection of the label vector onto the span of the features; residual $\perp$ features) is what makes regression make sense.

6.3 Exercises

Exercise 6.1 (Hand)**.** Run Gram–Schmidt on $v_1 = (1, 1, 0)$, $v_2 = (1, 0, 1)$ to produce an orthonormal basis e_1, e_2 of their span U. Then compute $P_U x$ for $x = (0, 0, 3)$ and verify $x - P_U x$ is orthogonal to both e_1 and e_2.

Exercise 6.2 (Hand)**.** Fit the least-squares line to $(0, 0), (1, 1), (2, 1), (3, 2)$ via the normal equations. Verify the residual is orthogonal to both columns of your A.

Exercise 6.3 (Proof, $\star$)**.** Prove that $\mathbb{R}^n = U \oplus U^{\perp}$ for every subspace U; that is, every x is uniquely a sum of a vector in U and a vector in $U^{\perp}$. (Existence: Theorem 6.3(a). Uniqueness: show $U \cap U^{\perp} = \{0\}$ — what is $\langle v, v \rangle$ for v in both?)

Exercise 6.4 (Proof)**.** Show $\langle P_U x, y \rangle = \langle x, P_U y \rangle$ for all x, y (projections are self-adjoint), and $P_U(P_U x) = P_U x$ (idempotent), directly from the formula in Theorem 6.3.

Deeper reading. LADR 6B (orthonormal bases, Gram–Schmidt), 6C (orthogonal complements, minimization — Axler solves this exact least-squares problem); Strang, ch. "Orthogonality" (projections; least squares approximations).

7 Gradients for Machine Learning

NEEDED FOR: Machine learning (core)

This lesson is the bridge from linear algebra to the opening machine-learning lectures. It is multivariable differentiation in the minimal dose: gradients of the two function shapes machine learning actually optimizes.

7.1 Theory

Definition 7.1 (Gradient)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$. The *partial derivative* $\frac{\partial f}{\partial x_k}(x)$ is the ordinary derivative of $t \mapsto f(x + te_k)$ at $t = 0$. If all partials exist and are continuous, the *gradient* is the vector

$$\nabla f(x) = \Big(\tfrac{\partial f}{\partial x_1}(x), \dots, \tfrac{\partial f}{\partial x_n}(x)\Big) \in \mathbb{R}^n,$$

and f admits the first-order approximation $f(x + h) \approx f(x) + \langle \nabla f(x),\, h\rangle$ for small h.

Proposition 7.2 (Steepest ascent)**.** *Suppose $\nabla f(x) \neq 0$. Among all unit vectors u, the directional rate of change $\langle \nabla f(x),\, u\rangle$ is maximized uniquely at $u = \nabla f(x)/\|\nabla f(x)\|$ and minimized uniquely at its negative.*

Proof. By Cauchy–Schwarz (Theorem 2.5), $|\langle \nabla f(x),\, u\rangle| \le \|\nabla f(x)\|\,\|u\| = \|\nabla f(x)\|$, with equality iff u is a scalar multiple of $\nabla f(x)$; the unit multiples are $\pm\nabla f(x)/\|\nabla f(x)\|$, giving the maximum with sign $+$ and the minimum with sign $-$. □

That is the entire justification for *gradient descent*: step against the gradient,

$$x^{(t+1)} = x^{(t)} - \eta\, \nabla f\big(x^{(t)}\big),$$

because $-\nabla f$ is locally the fastest way downhill. ($\eta > 0$ is the *step size* or *learning rate*.)

Proposition 7.3 (The two gradients to memorize)**.** *Fix $a \in \mathbb{R}^n$ and a symmetric $n \times n$ matrix M (meaning $M^{\mathsf{T}} = M$). Then*

$$\textit{(i)}\ \nabla_x \langle a,\, x\rangle = a; \qquad \textit{(ii)}\ \nabla_x\big(x^{\mathsf{T}} M x\big) = 2Mx.$$

Proof. (i) $\langle a,\, x\rangle = \sum_k a_k x_k$, so $\partial/\partial x_k$ picks out a_k; assembling the partials gives a.

(ii) Write $g(x) = x^{\mathsf{T}} M x = \sum_{i,j} M_{i,j} x_i x_j$. Differentiating with respect to x_k: the terms containing x_k are those with $i = k$ or $j = k$ (the term $i = j = k$ is $M_{k,k}x_k^2$, derivative $2M_{k,k}x_k$, consistent with counting it in both groups once each):

$$\frac{\partial g}{\partial x_k} = \sum_j M_{k,j} x_j + \sum_i M_{i,k} x_i = (Mx)_k + (M^{\mathsf{T}} x)_k = 2\,(Mx)_k,$$

using symmetry in the last step. Assembling over k gives $2Mx$. □

Corollary 7.4 (Gradient of the squared-error loss)**.** *For $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, the function $f(x) = \|Ax - b\|^2$ has*

$$\nabla f(x) = 2A^{\mathsf{T}}(Ax - b).$$

Consequently $\nabla f(\hat{x}) = 0 \iff A^{\mathsf{T}} A\hat{x} = A^{\mathsf{T}} b$: the normal equations of Lesson 6 are exactly the vanishing-gradient condition.

Proof. Expand using $\|v\|^2 = \langle v,\, v\rangle$, linearity, and $\langle Ax,\, b\rangle = \langle x,\, A^{\mathsf{T}} b\rangle$ (Lesson 4):

$$f(x) = \langle Ax - b,\, Ax - b\rangle = x^{\mathsf{T}}(A^{\mathsf{T}}A)x - 2\langle A^{\mathsf{T}} b,\, x\rangle + \|b\|^2.$$

The matrix $A^{\mathsf{T}}A$ is symmetric since $(A^{\mathsf{T}}A)^{\mathsf{T}} = A^{\mathsf{T}}A$. Now apply Proposition 7.3: the quadratic term contributes $2A^{\mathsf{T}}Ax$, the linear term $-2A^{\mathsf{T}}b$, the constant nothing. Sum: $2A^{\mathsf{T}}(Ax - b)$. □

Two roads, one summit. Lesson 6 found the least-squares solution by geometry (projection: residual $\perp$ columns). This lesson found it by calculus (set the gradient to zero). They agree — $A^{\mathsf{T}}(A\hat{x} - b) = 0$ either way — and machine learning takes a third road to the same place: iterate $w \leftarrow w - \eta \nabla f(w)$, which for this convex f converges to the same $\hat{x}$ (for suitably small η). Recognizing all three as one is the payoff of this booklet.

7.2 Practice: by hand

Example 7.5 (A full gradient computation)**.** $f(w_1, w_2) = (3w_1 + 2w_2 - 5)^2$. Chain rule per coordinate: with $r = 3w_1 + 2w_2 - 5$,

$$\frac{\partial f}{\partial w_1} = 2r \cdot 3, \qquad \frac{\partial f}{\partial w_2} = 2r \cdot 2, \qquad \nabla f = 2r\,(3, 2).$$

This matches Corollary 7.4 with $A = (3 \;\; 2)$ (one row), $b = 5$: $\nabla f = 2A^{\mathsf{T}}(Aw - b)$. Note the gradient is $2r$ times the feature vector $(3, 2)$ — *residual times features*, the shape of every gradient-descent update in machine learning's learning lectures.

Example 7.6 (Two steps of gradient descent by hand)**.** Minimize $f(w) = (w_1 - 1)^2 + 4(w_2)^2$, i.e. $w^{\mathsf{T}} M w - 2w_1 + 1$ with $M = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 4 \end{smallmatrix}\right)$; so $\nabla f(w) = (2(w_1 - 1),\, 8w_2)$. Start at $w^{(0)} = (0, 1)$ with $\eta = 0.1$:

$$\nabla f(w^{(0)}) = (-2, 8) \;\Rightarrow\; w^{(1)} = (0, 1) - 0.1\,(-2, 8) = (0.2, 0.2);$$

$$\nabla f(w^{(1)}) = (-1.6, 1.6) \;\Rightarrow\; w^{(2)} = (0.2, 0.2) - 0.1(-1.6, 1.6) = (0.36, 0.04).$$

Both coordinates march toward the minimizer $(1, 0)$ — the steep w_2 direction (curvature 4) converges faster at this η, and would oscillate if η were too large (check: the w_2 update is $w_2 \leftarrow (1 - 0.8)\, w_2$ here, and $\eta > 0.25$ makes $|1 - 8\eta| > 1$). Step-size sensitivity is a lived experience in machine learning homework; this two-line example is its essence.

Machine-learning connection. The learning lectures define the training loss as an average of per-example losses,

$$\text{TrainLoss}(w) = \tfrac{1}{m} \textstyle\sum_i \text{Loss}(x_i, y_i, w),$$

and minimize it by (stochastic) gradient descent. For the squared loss, $\nabla_w \text{Loss} = 2\,(w \cdot \phi(x) - y)\,\phi(x)$: residual times feature vector, exactly the pattern above. For the hinge loss on a misclassified example, $\nabla_w \text{Loss} = -y\,\phi(x)$. Every update you will implement has the form $w \leftarrow w - \eta\,(\text{scalar})\,\phi(x)$: a scalar–vector product plus a vector subtraction, both $O(n)$: one cheap pass over the features per example, which is why SGD scales to huge datasets.

7.3 Exercises

Exercise 7.1 (Hand)**.** Compute ∇f for: (a) $f(w) = \langle (1, -2, 3),\, w \rangle$; (b) $f(w) = \|w\|^2$; (c) $f(w) = (w \cdot \phi - y)^2$ for fixed $\phi \in \mathbb{R}^n$, $y \in \mathbb{R}$ (answer in terms of the residual).

Exercise 7.2 (Hand)**.** Run two iterations of gradient descent on $f(w) = (2w_1 + w_2 - 3)^2$ from $w^{(0)} = (0, 0)$ with $\eta = 0.1$, by hand. Does the loss decrease at each step? (Compute it.)

Exercise 7.3 (Proof, $\star$)**.** Prove Corollary 7.4 a second way, without Proposition 7.3: expand $f(x+h) = \|A(x+h)-b\|^2$ into $f(x) + \langle g,\, h\rangle + \|Ah\|^2$ for an explicit vector g, and identify $\nabla f(x) = g$ from the first-order approximation. (This perturbation technique is how gradients are computed cleanly in general.)

Exercise 7.4 (Proof)**.** Show that for symmetric M, the function $g(x) = x^{\mathsf{T}}Mx$ satisfies $g(x+h) - g(x) - \langle 2Mx,\, h\rangle = h^{\mathsf{T}}Mh$ *exactly* (no approximation), and explain how this re-proves Proposition 7.3(ii).

Deeper reading. Strang, ch. "Orthogonality" again (least squares) — and for the calculus, any multivariable-calculus text; introductory machine-learning problem sets exercise exactly these gradient patterns.

8 Capstone: Linear Regression from Scratch

NEEDED FOR: Machine learning (core)

One project, forty-five minutes, everything at once. Type it, don't copy-paste.

Task. Generate synthetic data $y = w^\star \cdot \phi(t) + \text{noise}$ with $\phi(t) = (1, t, t^2)$ and $w^\star = (1, -2, 0.5)$; recover $w^\star$ three ways; verify the three answers coincide.

```
import numpy as np
rng = np.random.default_rng(0)

# --- data ---
m = 200
t = rng.uniform(-3, 3, size=m)
Phi = np.column_stack([np.ones(m), t, t**2])  # m x 3 feature matrix
w_star = np.array([1.0, -2.0, 0.5])
y = Phi @ w_star + 0.3 * rng.standard_normal(m)

# --- way 1: normal equations (Lesson 6, geometry) ---
w1 = np.linalg.solve(Phi.T @ Phi, Phi.T @ y)

# --- way 2: library least squares ---
w2 = np.linalg.lstsq(Phi, y, rcond=None)[0]

# --- way 3: gradient descent (Lesson 7, calculus) ---
def loss(w): return np.mean((Phi @ w - y)**2)
def grad(w): return 2/m * Phi.T @ (Phi @ w - y)

w3 = np.zeros(3)
for step in range(20000):
    w3 -= 0.01 * grad(w3)

for name, w in [("normal eq", w1), ("lstsq", w2), ("grad desc", w3)]:
    print(f"{name:10s} {np.round(w, 3)}  loss={loss(w):.4f}")
print("true      ", w_star)
```

Checkpoints (do all of these).

1. All three w's agree to $\sim$3 decimals and sit *near* (not exactly at) $w^\star$. Why not exactly? The noise: you recover the projection of y onto the span of the features, and y itself left that span when noise was added.

2. Compute the residual $r = y - \Phi w_1$ and check `np.allclose(Phi.T @ r, 0, atol=1e-8)`: the residual is orthogonal to all three feature columns (Lesson 6's picture).

3. `np.linalg.matrix_rank(Phi)` is 3 — the features are independent (Lesson 3), which is what licensed `solve` on $\Phi^\mathsf{T}\Phi$ (Theorem 6.4).

4. Break it on purpose: append a fourth feature column equal to `2*t` (dependent!). Watch `matrix_rank` stay 3 while the normal-equations route fails or misbehaves, and `lstsq` still returns an answer — now one of infinitely many minimizers, since the null space is nontrivial (Lesson 5: solution set $= \hat{x} + \operatorname{null}\Phi$).

5. Raise η to 0.2: gradient descent diverges (Lesson 7's step-size discussion, now lived).

If every checkpoint makes sense — not just runs, but *makes sense* — you are ready for the opening machine-learning lectures and problem sets.

Final self-check list for Part I

You should be able to, from memory:

- compute dot products, norms, cosines; project a vector onto a line (Lessons 1–2);
- decide independence of small lists; produce a basis of a described line/plane (Lesson 3);
- multiply matrices; read Ax as a combination of columns; state why matrices = linear maps (Lesson 4);
- state and prove rank–nullity; describe the full solution set of $Ax = b$ (Lesson 5);
- derive the normal equations two ways: residual-orthogonality and $\nabla f = 0$ (Lessons 6–7);
- compute $\nabla\langle a, x\rangle$, $\nabla\|Ax - b\|^2$, and hand-run two steps of gradient descent (Lesson 7);
- implement all of the above in NumPy without looking anything up (Capstone).

Part I deliberately omitted determinants, eigenvalues, diagonalization, SVD, and complex vector spaces because machine learning does not need them. They are exactly what Fourier analysis does need — so they are Part II. When the linear-algebra sprint is done, go to Part III (probability, which machine learning *does* need) next; return to Part II afterward, at a relaxed pace.

Part II: Eigenstructure, the Spectral Theorem, the DFT, and the SVD

9 Complex Vectors and Complex Exponentials

NEEDED FOR: Fourier analysis (core); statistical-inference background. Not needed for ML.

Fourier analysis lives over the complex numbers: the basic signals are the complex exponentials $e^{i\omega t}$, and the clean statements of orthogonality, eigenvalues, and the DFT all require $\mathbb{C}$. This lesson upgrades Lessons 1–2 from $\mathbb{R}$ to $\mathbb{C}$.

9.1 Theory

Recall $\mathbb{C} = \{a + bi : a, b \in \mathbb{R}\}$ with $i^2 = -1$; the *conjugate* of $z = a + bi$ is $\bar{z} = a - bi$ and the *modulus* is $|z| = \sqrt{a^2 + b^2} = \sqrt{z\bar{z}}$. Euler's formula $e^{i\theta} = \cos\theta + i\sin\theta$ says $e^{i\theta}$ is the point on the unit circle at angle θ; thus $|e^{i\theta}| = 1$, $\overline{e^{i\theta}} = e^{-i\theta}$, and $e^{i\theta}e^{i\varphi} = e^{i(\theta+\varphi)}$ (angles add — multiplication by $e^{i\theta}$ is rotation by θ).

The space $\mathbb{C}^n$ is defined exactly like $\mathbb{R}^n$ with complex coordinates and complex scalars; everything in Lessons 1 and 3–5 (span, independence, basis, dimension, linear maps, rank–nullity) goes through verbatim with $\mathbb{F} = \mathbb{C}$ — the proofs never used anything about $\mathbb{R}$ beyond field arithmetic. The one definition that must change is the inner product.

Definition 9.1 (Inner product on $\mathbb{C}^n$)**.** For $x, y \in \mathbb{C}^n$,

$$\langle x,\, y\rangle = x_1\overline{y_1} + \cdots + x_n\overline{y_n}.$$

Why the conjugate? Because we need $\langle x,\, x\rangle = \sum_k x_k\overline{x_k} = \sum_k |x_k|^2$ to be a nonnegative real number, so that $\|x\| = \sqrt{\langle x,\, x\rangle}$ is a genuine length. Without the conjugate, $\langle x,\, x\rangle$ could be negative or non-real (try $x = (i)$ in $\mathbb{C}^1$: $i \cdot i = -1$).

The price is that symmetry becomes *conjugate symmetry* and linearity holds only in the first slot:

$$\langle y,\, x\rangle = \overline{\langle x,\, y\rangle}, \qquad \langle \lambda x,\, y\rangle = \lambda\langle x,\, y\rangle, \qquad \langle x,\, \lambda y\rangle = \bar{\lambda}\,\langle x,\, y\rangle.$$

(Each is a one-line check.) With these amendments, every proof of Lesson 2 survives unchanged in structure: positivity, the orthogonal decomposition lemma, Cauchy–Schwarz $|\langle x,\, y\rangle| \le \|x\|\|y\|$, the triangle inequality, and the Pythagorean theorem for orthogonal vectors ($\langle x,\, y\rangle = 0$) all hold in $\mathbb{C}^n$, and Lesson 6's orthonormal-basis and projection theory holds verbatim with $\langle \cdot,\, \cdot\rangle$ in this complex sense. This is LADR 6A's actual setting.

Lemma 9.2 (Roots of unity; the key to the DFT)**.** *Fix an integer $N \ge 1$ and let $\omega = e^{-2\pi i/N}$. Then $\omega^N = 1$, and for any integer m:*

$$\sum_{j=0}^{N-1} \omega^{jm} = \begin{cases} N & \text{if } N \mid m, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. $\omega^N = e^{-2\pi i} = 1$. If $N \mid m$, then $\omega^m = (\omega^N)^{m/N} = 1$ and every term of the sum is 1: total N. If $N \nmid m$, then $\omega^m \neq 1$ (since $\omega^m = e^{-2\pi i m/N} = 1$ exactly when m/N is an integer), so the finite geometric series gives

$$\sum_{j=0}^{N-1} (\omega^m)^j = \frac{(\omega^m)^N - 1}{\omega^m - 1} = \frac{(\omega^N)^m - 1}{\omega^m - 1} = \frac{1-1}{\omega^m - 1} = 0. \qquad \square$$

Geometrically: the numbers ω^{jm} for $j = 0, \ldots, N-1$ are equally spaced points on the unit circle (when $N \nmid m$), and equally spaced points on a circle sum to zero by symmetry. The lemma is that picture made algebra, and Lesson 12 will use it to prove that the Fourier basis is orthonormal.

9.2 Practice: by hand

Example 9.3 (Complex arithmetic warm-up)**.** $(1+i)^2 = 1 + 2i + i^2 = 2i$. Also $|1+i| = \sqrt{2}$ and $1 + i = \sqrt{2}\, e^{i\pi/4}$, so $(1+i)^2 = 2e^{i\pi/2} = 2i$: polar form turns powers into arithmetic on angles. Similarly $(1+i)^8 = 16\, e^{2\pi i} = 16$.

Example 9.4 (Fourth roots of unity)**.** For $N = 4$: $\omega = e^{-2\pi i/4} = -i$, and the powers $\omega^0, \omega^1, \omega^2, \omega^3 = 1, -i, -1, i$: the four compass points of the unit circle. Check the lemma for $m = 1$: $1 - i - 1 + i = 0$. ✓ For $m = 4$: $1 + 1 + 1 + 1 = 4$. ✓

Example 9.5 (A complex inner product)**.** $x = (1, i)$, $y = (i, 1)$ in $\mathbb{C}^2$: $\langle x, y\rangle = 1 \cdot \bar{i} + i \cdot \bar{1} = -i + i = 0$ — orthogonal. And $\|x\|^2 = |1|^2 + |i|^2 = 2$. Note $\langle y, x\rangle = \bar{0} = 0$ too, but in general the order matters up to conjugation.

Fourier connection. Fourier analysis's protagonist is $e^{2\pi i s t}$. Fourier series write a periodic signal as $f(t) = \sum_k c_k e^{2\pi i k t}$, with coefficients $c_k = \int_0^1 f(t)\, \overline{e^{2\pi i k t}}\, dt$ — read that as $\langle f, e_k\rangle$ for the inner product $\langle f, g\rangle = \int_0^1 f\bar{g}$ on functions. It is literally Proposition 6.2's formula $a_j = \langle v, e_j\rangle$, with integrals replacing sums: Fourier coefficients are coordinates in an orthonormal basis, and partial Fourier sums are orthogonal projections (Theorem 6.3) onto the span of finitely many exponentials — hence the best least-squares approximants. Carrying that dictionary into Fourier analysis makes the first three weeks feel familiar.

9.3 Exercises

Exercise 9.1 (Hand)**.** Write $z = -1 + i\sqrt{3}$ in polar form $re^{i\theta}$, then compute z^3 two ways (polar and direct expansion) and confirm they agree.

Exercise 9.2 (Hand)**.** List the sixth roots of unity $e^{-2\pi i j/6}$, $j = 0, \ldots, 5$, in the form $a + bi$. Verify by hand that they sum to 0, and that the sum of their squares is also 0 (which case of Lemma 9.2 is that?).

Exercise 9.3 (Proof, ⋆)**.** Prove the complex Pythagorean theorem and Cauchy–Schwarz: adapt the proofs of Theorem 2.3 and Theorem 2.5 to $\mathbb{C}^n$, flagging exactly where conjugate symmetry and conjugate-linearity in the second slot are used. (In the decomposition lemma, take $c = \langle x, y\rangle / \|y\|^2$ and verify $\langle z, y\rangle = 0$ still holds.)

Exercise 9.4 (Proof)**.** Show that for all $x, y \in \mathbb{C}^n$: $\langle x, y\rangle + \langle y, x\rangle = 2\,\mathrm{Re}\langle x, y\rangle$, and deduce $\|x + y\|^2 = \|x\|^2 + 2\,\mathrm{Re}\langle x, y\rangle + \|y\|^2$.

Deeper reading. LADR 1A ($\mathbb{C}^n$), 4 (polynomials over $\mathbb{C}$: the fundamental theorem of algebra, used next lesson), 6A (complex inner products); Strang, ch. "Complex Vectors and Matrices."

10 Eigenvalues, Eigenvectors, and Diagonalization

NEEDED FOR: Fourier analysis (core); statistical inference (Markov chains, PCA). Skimmable for the MDP side of ML.

10.1 Theory

Throughout, $T : V \to V$ is a linear map from a space to itself (an *operator*); V is $\mathbb{R}^n$ or $\mathbb{C}^n$.

Definition 10.1 (Eigenvalue, eigenvector)**.** A scalar $\lambda \in \mathbb{F}$ is an *eigenvalue* of T if there exists $v \neq 0$ with

$$Tv = \lambda v;$$

any such v is an *eigenvector* for λ. Equivalently (Proposition 5.2): λ is an eigenvalue iff $T - \lambda I$ is not injective; the eigenvectors for λ, together with 0, form the subspace $\operatorname{null}(T - \lambda I)$, the *eigenspace*.

An eigenvector is a direction the operator does not turn — only stretches by the factor λ. On such directions, applying T repeatedly is trivial: $T^k v = \lambda^k v$.

Theorem 10.2 (Independence of eigenvectors, LADR 5.11)**.** *Eigenvectors corresponding to distinct eigenvalues are linearly independent.*

Proof. Let $v_1, \ldots, v_m$ be eigenvectors for distinct eigenvalues $\lambda_1, \ldots, \lambda_m$, and suppose for contradiction the list is dependent. Since each $v_i \neq 0$, the linear dependence lemma (Lemma 3.3) gives a smallest index k with $v_k \in \operatorname{span}(v_1, \ldots, v_{k-1})$, and $v_1, \ldots, v_{k-1}$ is independent by minimality of k. Write

$$v_k = a_1 v_1 + \cdots + a_{k-1} v_{k-1}.$$

Apply T to both sides, and separately multiply both sides by λ_k:

$$\lambda_k v_k = a_1 \lambda_1 v_1 + \cdots + a_{k-1} \lambda_{k-1} v_{k-1}, \qquad \lambda_k v_k = a_1 \lambda_k v_1 + \cdots + a_{k-1} \lambda_k v_{k-1}.$$

Subtracting, $0 = \sum_{i<k} a_i(\lambda_i - \lambda_k) v_i$. Independence of $v_1, \ldots, v_{k-1}$ forces $a_i(\lambda_i - \lambda_k) = 0$ for each i; since the eigenvalues are distinct, every $a_i = 0$, making $v_k = 0$ — contradicting that v_k is an eigenvector. □

Corollary 10.3. *An operator on an n-dimensional space has at most n distinct eigenvalues.*

Proof. Eigenvectors for distinct eigenvalues form an independent list, and independent lists in an n-dimensional space have length at most n (Theorem 3.4). □

Do eigenvalues exist at all? Over $\mathbb{R}$, not always: rotation of $\mathbb{R}^2$ by $90°$ turns every direction, so it has no real eigenvalue. Over $\mathbb{C}$, always — and Axler's proof is a gem, using no determinants:

Theorem 10.4 (Existence of eigenvalues over $\mathbb{C}$, LADR 5.19)**.** *Every operator on a nonzero finite-dimensional complex vector space has an eigenvalue.*

Proof. Let T be an operator on $\mathbb{C}^n$, $n \geq 1$, and pick any $v \neq 0$. The $n+1$ vectors

$$v,\, Tv,\, T^2v, \ldots, T^nv$$

must be linearly dependent (Theorem 3.4 again: $n+1$ vectors in an n-dimensional space). So there are scalars $a_0, \ldots, a_n$, not all zero, with $a_0v + a_1Tv + \cdots + a_nT^nv = 0$. Let m be the largest index with $a_m \neq 0$; then $m \geq 1$, because $m = 0$ would give $a_0v = 0$ with $a_0 \neq 0$ and $v \neq 0$.

Now the polynomial $p(z) = a_0 + a_1z + \cdots + a_mz^m$ has degree $m \geq 1$, so by the fundamental theorem of algebra it factors completely over $\mathbb{C}$:

$$p(z) = a_m(z - \lambda_1)(z - \lambda_2)\cdots(z - \lambda_m)$$

for some $\lambda_1, \ldots, \lambda_m \in \mathbb{C}$. Because powers of T commute with each other, the same factorization holds with T substituted for z:

$$0 = a_0v + a_1Tv + \cdots + a_mT^mv = a_m(T - \lambda_1 I)(T - \lambda_2 I)\cdots(T - \lambda_m I)\, v.$$

The right side is a composition of operators applied to the nonzero vector v, producing 0. If every factor $T - \lambda_j I$ were injective, the composition would be injective and could not send $v \neq 0$ to 0. Hence some $T - \lambda_j I$ is not injective: λ_j is an eigenvalue. □

Why this proof is worth knowing. It explains *where eigenvalues come from*: dependence among $v, Tv, T^2v, \ldots$ produces a polynomial identity in T, and eigenvalues are the roots. The fundamental theorem of algebra — every complex polynomial factors into linear pieces — is the sole reason $\mathbb{C}$ guarantees eigenvalues while $\mathbb{R}$ does not (over $\mathbb{R}$, $z^2 + 1$ refuses to factor, and that irreducible quadratic *is* the 90° rotation's obstruction).

Definition 10.5 (Diagonalizable)**.** An operator T on V ($\dim V = n$) is *diagonalizable* if V has a basis $v_1, \ldots, v_n$ of eigenvectors of T. In matrix terms: $A = PDP^{-1}$, where D is diagonal with the eigenvalues $\lambda_1, \ldots, \lambda_n$ on its diagonal and P is the invertible matrix whose columns are the corresponding eigenvectors.

The matrix identity is just bookkeeping for "A acts diagonally on the eigenbasis": $AP = PD$ says column-by-column that $Av_k = \lambda_k v_k$ (compute both sides' kth columns: AP has kth column Av_k, and PD has kth column $\lambda_k v_k$). By Theorem 10.2, an operator on an n-dimensional space with n *distinct* eigenvalues is automatically diagonalizable. Repeated eigenvalues may or may not ruin it: I has one eigenvalue with an n-dimensional eigenspace (diagonalizable — it is diagonal), while the shear $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ has the single eigenvalue 1 with only a one-dimensional eigenspace (not diagonalizable: no basis of eigenvectors exists).

Proposition 10.6 (Powers via diagonalization)**.** *If $A = PDP^{-1}$ then $A^k = PD^kP^{-1}$, and D^k is diagonal with entries λ_j^k.*

Proof. $A^2 = PDP^{-1}PDP^{-1} = PD^2P^{-1}$; induction gives the rest. Diagonal matrices multiply entrywise on the diagonal. □

Understanding a diagonalizable operator's long-run behavior thus reduces to scalar arithmetic: components along eigenvectors with $|\lambda| < 1$ decay, $|\lambda| > 1$ grow, $|\lambda| = 1$ persist.

10.2 Practice: by hand

Example 10.7 (Full eigen-computation, 2×2)**.** $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$. Find λ making $A - \lambda I$ singular.

A 2×2 matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ is singular exactly when $ad - bc = 0$ (its columns are then parallel — Lesson 14 names this quantity the determinant). Here:

$$(2-\lambda)^2 - 1 = 0 \iff 2 - \lambda = \pm 1 \iff \lambda = 1 \text{ or } 3.$$

Eigenvectors: for $\lambda = 3$, solve $(A - 3I)v = 0$: $\left(\begin{smallmatrix} -1 & 1 \\ 1 & -1 \end{smallmatrix}\right) v = 0 \Rightarrow v = (1,1)$. For $\lambda = 1$: $\left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right) v = 0 \Rightarrow v = (1,-1)$. Two distinct eigenvalues $\Rightarrow$ diagonalizable, and note $(1,1) \perp (1,-1)$ — not an accident: A is symmetric, and Lesson 11 proves symmetric matrices always get orthogonal eigenvectors.

Geometrically, A stretches the diagonal direction $(1,1)$ by 3 and the anti-diagonal $(1,-1)$ by 1: an anisotropic stretch along tilted axes.

Example 10.8 (Powers put to work)**.** With A as above, compute $A^{10}(1,0)$ without ten multiplications: decompose $(1,0) = \frac{1}{2}(1,1) + \frac{1}{2}(1,-1)$, then

$$A^{10}(1,0) = \tfrac{1}{2}\,3^{10}(1,1) + \tfrac{1}{2}\,1^{10}(1,-1) = \tfrac{1}{2}\big(3^{10}+1,\ 3^{10}-1\big) = (29525,\, 29524).$$

Eigen-coordinates turn matrix powers into scalar powers — the essential trick behind solving recurrences, Markov chains, and linear ODEs.

> **Fourier connection.** A linear time-invariant (LTI) system — any circuit or filter Fourier analysis studies — has the complex exponentials as its *eigenfunctions*: feed in $e^{2\pi i s t}$ and out comes $H(s)\,e^{2\pi i s t}$, the same signal scaled by a complex number. The scale factor $H(s)$ (the frequency response / transfer function) is precisely an eigenvalue, indexed by frequency. "Diagonalize the system in the Fourier basis" is the slogan under all of Fourier analysis: convolution in time becomes multiplication (by eigenvalues) in frequency. Lesson 12 proves the finite-dimensional version of this exactly.

10.3 Exercises

Exercise 10.1 (Hand)**.** Find all eigenvalues and eigenvectors of (a) $\left(\begin{smallmatrix} 3 & 0 \\ 0 & -2 \end{smallmatrix}\right)$, (b) $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ (reflection through $x_2 = x_1$ — interpret the answer geometrically), (c) $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ over $\mathbb{R}$ and then over $\mathbb{C}$.

Exercise 10.2 (Hand)**.** Diagonalize $A = \left(\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right)$ and use Proposition 10.6 to find a closed form for $A^k(1,0)$.

Exercise 10.3 (Proof, $\star$)**.** Prove that if T is invertible with eigenvalue λ (necessarily $\lambda \neq 0$ — why?), then λ^{-1} is an eigenvalue of T^{-1}, with the same eigenvectors. Also show: λ^k is an eigenvalue of T^k for every $k \geq 1$.

Exercise 10.4 (Proof)**.** Show the shear $S = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ is not diagonalizable: find all eigenvalues, show the eigenspace is one-dimensional, and conclude no basis of eigenvectors exists.

Deeper reading. LADR 5A (invariant subspaces, eigenvalues), 5B (existence via minimal polynomials — a refinement of Theorem 10.4), 5D (diagonalizable operators); Strang, ch. "Eigenvalues and Eigenvectors."

11 Self-Adjoint Operators and the Spectral Theorem

NEEDED FOR: Fourier analysis (core); statistical inference (covariance matrices, PCA — reused in Lesson 23)

11.1 Theory

Work in $\mathbb{F}^n$ ($\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$) with the inner product of Lessons 2 and 9. Recall from Lesson 4 the defining property of the transpose; over $\mathbb{C}$ its correct analogue is the *conjugate transpose* (*adjoint*) $A^* = \overline{A^{\mathsf{T}}}$, which satisfies $\langle Ax,\, y\rangle = \langle x,\, A^*y\rangle$ for all x, y (same expansion as before, tracking conjugates). Over $\mathbb{R}$, $A^* = A^{\mathsf{T}}$.

Definition 11.1 (Self-adjoint)**.** A is *self-adjoint* (*Hermitian*; over $\mathbb{R}$: *symmetric*) if $A^* = A$; equivalently $\langle Ax,\, y\rangle = \langle x,\, Ay\rangle$ for all x, y.

Proposition 11.2 (Real eigenvalues, LADR 7.12)**.** *Every eigenvalue of a self-adjoint matrix is real.*

Proof. Let $Av = \lambda v$ with $v \neq 0$. Then

$$\lambda\|v\|^2 = \langle \lambda v,\, v\rangle = \langle Av,\, v\rangle = \langle v,\, Av\rangle = \langle v,\, \lambda v\rangle = \bar{\lambda}\,\|v\|^2.$$

Since $\|v\|^2 \neq 0$, $\lambda = \bar{\lambda}$: real. □

Proposition 11.3 (Orthogonal eigenspaces)**.** *If A is self-adjoint, eigenvectors for distinct eigenvalues are orthogonal.*

Proof. Let $Av = \lambda v$, $Aw = \mu w$, $\lambda \neq \mu$ (both real by Proposition 11.2). Then

$$\lambda\langle v,\, w\rangle = \langle Av,\, w\rangle = \langle v,\, Aw\rangle = \bar{\mu}\langle v,\, w\rangle = \mu\langle v,\, w\rangle,$$

so $(\lambda - \mu)\langle v,\, w\rangle = 0$, forcing $\langle v,\, w\rangle = 0$. □

Theorem 11.4 (Spectral theorem, LADR 7.29/7.31)**.** *Let A be a self-adjoint $n \times n$ matrix (real symmetric, or complex Hermitian). Then $\mathbb{F}^n$ has an orthonormal basis of eigenvectors of A, and all eigenvalues are real. Equivalently, $A = Q\Lambda Q^*$ with Q having orthonormal columns ($Q^*Q = I$) and Λ real diagonal.*

Proof. Induct on n. For $n = 1$ any unit vector works. For $n > 1$: A has an eigenvalue λ_1. (Over $\mathbb{C}$ this is Theorem 10.4. Over $\mathbb{R}$: regard the real symmetric A as a complex matrix; it is Hermitian, so it has a complex eigenvalue λ_1, real by Proposition 11.2; then $A - \lambda_1 I$ is a real matrix that is singular over $\mathbb{C}$, hence has rank $< n$, hence is singular over $\mathbb{R}$ — rank is the same whether we allow real or complex coefficients, since elimination never leaves the field the entries live in — so a real eigenvector exists.)

Pick a unit eigenvector e_1 for λ_1, and let $U = \{x : \langle x,\, e_1\rangle = 0\}$ be its orthogonal complement, of dimension $n - 1$ (Exercise 6.3). The key point is that A maps U into U: for $u \in U$,

$$\langle Au,\, e_1\rangle = \langle u,\, Ae_1\rangle = \langle u,\, \lambda_1 e_1\rangle = \lambda_1\langle u,\, e_1\rangle = 0,$$

so $Au \in U$. The restriction of A to U is an operator on an $(n-1)$-dimensional inner product space, still self-adjoint (the identity $\langle Ax,\, y\rangle = \langle x,\, Ay\rangle$ holds in particular for $x, y \in U$). By

the induction hypothesis, U has an orthonormal basis $e_2, \ldots, e_n$ of eigenvectors of A. Then $e_1, e_2, \ldots, e_n$ is orthonormal (each later e_j lies in $U \perp e_1$) and consists of eigenvectors.

For the matrix form: let Q have columns $e_1, \ldots, e_n$. Orthonormality of columns says exactly $Q^*Q = I$, and $Ae_k = \lambda_k e_k$ says $AQ = Q\Lambda$; since Q is square with $Q^*Q = I$, $Q^{-1} = Q^*$ and $A = Q\Lambda Q^*$. □

What the spectral theorem buys you. Self-adjoint matrices are exactly the ones that are a *pure stretch along some perpendicular axes*: rotate into the right orthonormal frame (Q^*), scale each axis by a real number (Λ), rotate back (Q). No shearing, no hidden rotation. Because the eigenbasis is orthonormal, coordinates in it are free ($a_j = \langle x, e_j \rangle$, Proposition 6.2) — so analyzing A's action on any vector costs n dot products. This is the finite-dimensional model of what Fourier analysis does for differential operators.

A useful companion definition: a self-adjoint A is *positive semidefinite* (PSD) if $\langle Ax, x \rangle \geq 0$ for all x. By the spectral theorem this holds iff all eigenvalues are ≥ 0 (expand x in the eigenbasis: $\langle Ax, x \rangle = \sum_j \lambda_j |\langle x, e_j \rangle|^2$). The matrices A^*A from least squares (Lesson 6) are always PSD: $\langle A^*Ax, x \rangle = \|Ax\|^2 \geq 0$ — this observation powers the SVD in Lesson 13 (and, in Lesson 23, the fact that covariance matrices are PSD).

11.2 Practice: by hand

Example 11.5 (Spectral decomposition, fully worked)**.** From Example 10.7, $A = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ has eigenpairs $\lambda = 3$, $v = (1, 1)$ and $\lambda = 1$, $v = (1, -1)$, already orthogonal as Proposition 11.3 promised. Normalize:

$$Q = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \qquad A = Q\Lambda Q^{\mathsf{T}}.$$

Verify one entry: $(Q\Lambda Q^{\mathsf{T}})_{11} = \frac{1}{2}(3 \cdot 1 \cdot 1 + 1 \cdot 1 \cdot 1) = 2$. ✓ Also read off positivity: eigenvalues $3, 1 > 0$, so $\langle Ax, x \rangle > 0$ for all $x \neq 0$ — A is *positive definite*.

Example 11.6 (Quadratic forms become sums of squares)**.** For the same A, with new coordinates $y = Q^{\mathsf{T}}x$ (the eigen-coordinates),

$$x^{\mathsf{T}}Ax = x^{\mathsf{T}}Q\Lambda Q^{\mathsf{T}}x = y^{\mathsf{T}}\Lambda y = 3y_1^2 + y_2^2.$$

The level sets $x^{\mathsf{T}}Ax = 1$ are ellipses with axes along $(1, 1)$ and $(1, -1)$ and semi-axis lengths $1/\sqrt{3}$ and 1. Diagonalization untangles every quadratic form into independent squared terms.

Fourier connection. Symmetric structure is why real signals have the Fourier symmetries Fourier analysis drills ($\hat{f}(-s) = \overline{\hat{f}(s)}$, etc.), and self-adjointness of the underlying operators is what makes Fourier eigen-expansions come with orthogonality — which in turn yields Parseval/Rayleigh identities: energy in time equals energy in frequency, i.e., $\|x\|$ is preserved when you switch to an orthonormal (eigen)basis. The next lesson makes the norm-preservation statement exact for the DFT.

11.3 Exercises

Exercise 11.1 (Hand)**.** Find the spectral decomposition $Q\Lambda Q^{\mathsf{T}}$ of $\left(\begin{smallmatrix} 0 & 2 \\ 2 & 3 \end{smallmatrix}\right)$, and write the quadratic form $q(x) = 4x_1x_2 + 3x_2^2$ as a combination of squares in the eigen-coordinates. Is the matrix PSD?

Exercise 11.2 (Hand)**.** The Hermitian matrix $H = \left(\begin{smallmatrix} 1 & i \\ -i & 1 \end{smallmatrix}\right)$ (note $H^* = H$): find its (real!) eigenvalues and a pair of orthogonal eigenvectors in $\mathbb{C}^2$; verify orthogonality with the complex inner product.

Exercise 11.3 (Proof, $\star$)**.** Prove: a self-adjoint matrix with all eigenvalues in $\{0, 1\}$ is exactly an orthogonal projection P_U (as in Theorem 6.3) onto the span U of its $\lambda = 1$ eigenvectors. (Use the spectral theorem to write $A = \sum_j \lambda_j e_j e_j^*$, i.e. $Ax = \sum_j \lambda_j \langle x,\, e_j\rangle e_j$, and compare with the projection formula.)

Exercise 11.4 (Proof)**.** Show that for any (not necessarily self-adjoint) real A, the matrix $S = \frac{1}{2}(A + A^{\mathsf{T}})$ is symmetric and $x^{\mathsf{T}}Ax = x^{\mathsf{T}}Sx$ for all x — so quadratic forms lose nothing by assuming symmetry (as Proposition 7.3 did).

Deeper reading. LADR 7A (self-adjoint and normal operators), 7B (spectral theorem — Axler also proves the normal case over $\mathbb{C}$, the full characterization of orthonormally diagonalizable operators), 7C (positive operators); Strang, ch. "Symmetric Matrices and Positive Definiteness."

12 Unitary Matrices and the Discrete Fourier Transform

NEEDED FOR: Fourier analysis (core); statistical-inference background (transform-domain reasoning)

12.1 Theory

Definition 12.1 (Unitary / orthogonal matrix)**.** A square matrix Q over $\mathbb{C}$ is *unitary* if $Q^*Q = I$ (over $\mathbb{R}$, with $Q^{\mathsf{T}}Q = I$: *orthogonal*).

Theorem 12.2 (Three faces of unitarity, LADR 7.51)**.** *For a square matrix Q, the following are equivalent:*

(a) $Q^*Q = I$ (columns are orthonormal);

(b) $\langle Qx,\, Qy\rangle = \langle x,\, y\rangle$ for all x, y (inner products preserved);

(c) $\|Qx\| = \|x\|$ for all x (lengths preserved).

Moreover a unitary Q is invertible with $Q^{-1} = Q^*$, which is again unitary.

Proof. (a) $\Rightarrow$ (b): $\langle Qx,\, Qy\rangle = \langle x,\, Q^*Qy\rangle = \langle x,\, y\rangle$, using the adjoint's defining property. (b) $\Rightarrow$ (c): take $y = x$ and square roots. (c) $\Rightarrow$ (a): first note entry (j, k) of Q^*Q is $\langle Qe_k,\, Qe_j\rangle$. From (c) and the expansion $\|Q(x+y)\|^2 = \|Qx\|^2 + 2\,\mathrm{Re}\langle Qx,\, Qy\rangle + \|Qy\|^2$ (Exercise 9.4), preserving all norms forces $\mathrm{Re}\langle Qx,\, Qy\rangle = \mathrm{Re}\langle x,\, y\rangle$ for all x, y; replacing x by ix recovers the imaginary parts likewise (over $\mathbb{R}$, the real-part identity is already everything). So (b) holds, and applying

it to standard basis vectors gives $\langle Qe_k,\, Qe_j\rangle = \langle e_k,\, e_j\rangle = \delta_{jk}$: the columns are orthonormal, which is (a).

Finally, $Q^*Q = I$ with Q square makes Q injective (if $Qx = 0$ then $x = Q^*Qx = 0$), hence invertible by Corollary 5.4, and multiplying $Q^*Q = I$ by Q^{-1} on the right gives $Q^* = Q^{-1}$; then $(Q^*)^*Q^* = QQ^{-1} = I$ shows Q^* is unitary. □

Unitary matrices are the rigid motions of $\mathbb{C}^n$ fixing the origin: they preserve all lengths and angles. A change of basis between two orthonormal bases is always unitary, and conversely.

Definition 12.3 (The DFT matrix)**.** Fix $N \geq 1$ and $\omega = e^{-2\pi i/N}$. The *discrete Fourier transform* of $x \in \mathbb{C}^N$ (indexing coordinates from 0) is the vector $\hat{x} = Wx$, where W is the $N \times N$ matrix with entries

$$W_{k,n} = \omega^{kn}, \qquad \text{i.e.} \qquad \hat{x}_k = \sum_{n=0}^{N-1} x_n\, e^{-2\pi ikn/N}.$$

The number $\hat{x}_k$ measures the correlation of the signal x with the frequency-k oscillation.

Theorem 12.4 (The DFT is (essentially) unitary)**.** *$W^*W = NI$. Hence $F = W/\sqrt{N}$ is unitary, the DFT is invertible with*

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} \hat{x}_k\, e^{+2\pi ikn/N} \qquad (\textit{the inverse DFT}),$$

and Parseval's identity holds: $\sum_n |x_n|^2 = \frac{1}{N}\sum_k |\hat{x}_k|^2$.

Proof. Entry (m,n) of W^*W is $\sum_k \overline{W_{k,m}} W_{k,n} = \sum_{k=0}^{N-1} \overline{\omega^{km}}\,\omega^{kn} = \sum_{k=0}^{N-1} \omega^{k(n-m)}$, which by the roots-of-unity lemma (Lemma 9.2) equals N if $m = n$ and 0 otherwise — here $|n-m| < N$, so $N \mid (n-m)$ occurs only for $m = n$. Thus $W^*W = NI$.

Consequently $F^*F = I$: F is unitary, so $W^{-1} = \frac{1}{N}W^*$, whose entry (n,k) is $\frac{1}{N}\overline{\omega^{kn}} = \frac{1}{N}e^{+2\pi ikn/N}$ — the displayed inversion formula. Parseval is Theorem 12.2(c) for F: $\|Fx\| = \|x\|$, i.e. $\frac{1}{N}\|Wx\|^2 = \|x\|^2$. □

Definition 12.5 (Circular convolution)**.** For $c, x \in \mathbb{C}^N$, with indices taken mod N,

$$(c * x)_n = \sum_{j=0}^{N-1} c_j\, x_{n-j \bmod N}.$$

Equivalently $c * x = Cx$ where C is the *circulant* matrix whose column j is c cyclically shifted down by j.

Theorem 12.6 (Convolution theorem)**.** *$\widehat{c * x}_k = \hat{c}_k\hat{x}_k$ for every k. Equivalently: every circulant matrix is diagonalized by the DFT, $C = W^{-1}\,\mathrm{diag}(\hat{c})\,W$, with eigenvalues $\hat{c}_0, \ldots, \hat{c}_{N-1}$.*

Proof. Compute, substituting $m = n - j \bmod N$ (as n runs over residues mod N with j fixed, so does m, and $\omega^{kn} = \omega^{k(j+m)}$ since $\omega^{kN} = 1$ makes exponents matter only mod N):

$$\widehat{c * x}_k = \sum_n \sum_j c_j x_{n-j}\,\omega^{kn} = \sum_j c_j\omega^{kj} \sum_m x_m\omega^{km} = \hat{c}_k\hat{x}_k.$$

For the matrix form: the identity says $W(Cx) = \mathrm{diag}(\hat{c})(Wx)$ for all x, i.e. $WC = \mathrm{diag}(\hat{c})W$, i.e. $C = W^{-1}\,\mathrm{diag}(\hat{c})W$. Reading it as a diagonalization: the columns of W^{-1} (the sampled complex exponentials) are eigenvectors of *every* circulant, with eigenvalues the DFT of the first column. □

The punchline of Part II. Shift-invariant linear operations (circulants) are precisely the ones diagonalized by the Fourier basis, and their eigenvalues are the transform of their impulse response c. Filtering = multiply in frequency. This is Lesson 10's "diagonalization turns matrix powers into scalars" plus Lesson 11's "orthonormal eigenbases make coordinates free," specialized to the one basis that never changes: the complex exponentials. A Fourier-analysis course spends weeks developing the continuous version.

12.2 Practice: by hand

Example 12.7 (The 4-point DFT, fully)**.** $N = 4$, $\omega = -i$ (Lesson 9). The matrix and a transform:

$$W = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}, \qquad x = (1,0,1,0) \;\Rightarrow\; \hat{x} = Wx = (2,0,2,0).$$

Sense check: x alternates with period 2 — energy at frequency 0 (its mean is not zero) and frequency 2 (the Nyquist alternation), none at frequencies 1, 3. Parseval: $\sum |x_n|^2 = 2$ and $\frac{1}{4}\sum |\hat{x}_k|^2 = \frac{1}{4}(4+4) = 2$. ✓

Example 12.8 (Convolution theorem, by hand)**.** Take the averaging filter $c = \frac{1}{2}(1,1,0,0)$ and $x = (1,0,1,0)$ with $N = 4$. Directly: $(c * x)_n = \frac{1}{2}(x_n + x_{n-1}) = \frac{1}{2}(1,1,1,1)$. Via frequency: $\hat{c} = \frac{1}{2}(2,\ 1-i,\ 0,\ 1+i)$, and $\hat{c} \odot \hat{x} = (2,0,0,0)$ (entrywise), whose inverse DFT is $\frac{1}{4}\cdot 2 \cdot (1,1,1,1) = \frac{1}{2}(1,1,1,1)$. ✓ The filter's zero at frequency 2 ($\hat{c}_2 = 0$) is why the alternating component of x is annihilated: averaging adjacent samples kills the fastest oscillation.

Fourier connection. This lesson is the last third of Fourier analysis in miniature: the DFT section of the course defines exactly this matrix, proves exactly this inversion and convolution theorem, and then studies the FFT — an algorithm computing Wx in $O(N \log N)$ instead of the obvious $O(N^2)$, arguably the most important algorithm in engineering. Everything earlier in the course (Fourier series, the continuous transform, sampling) has the same skeleton with integrals in place of sums; if you own this lesson, the Fourier course's DFT weeks are review.

12.3 Exercises

Exercise 12.1 (Hand)**.** Write out the 3-point DFT matrix using $\omega = e^{-2\pi i/3} = -\frac{1}{2} - \frac{\sqrt{3}}{2}i$, and compute $\hat{x}$ for $x = (1,1,1)$ and for $x = (1, \omega^{-1}, \omega^{-2})$. Explain both answers in one sentence each.

Exercise 12.2 (Hand)**.** For $N = 4$, write the circulant matrix C for $c = (0,1,0,0)$ (the cyclic shift). Verify directly that $v = (1, i^{-1}, i^{-2}, i^{-3}) = (1, -i, -1, i)$ satisfies $Cv = \lambda v$, and identify λ as an entry of $\hat{c}$.

Exercise 12.3 (Proof, $\star$)**.** Prove that the product of two unitary matrices is unitary and that every eigenvalue λ of a unitary matrix has $|\lambda| = 1$. (For the latter: apply Theorem 12.2(c) to an eigenvector.) Conclude the DFT eigenvalue fact: $|\hat{c}_k| \le \sum_j |c_j|$ for a circulant's eigenvalues, directly from the definition of $\hat{c}$ and the triangle inequality.

Exercise 12.4 (Proof)**.** Show that the set of circulant $N \times N$ matrices is closed under products and that any two circulants commute. (Two lines with Theorem 12.6: simultaneous diagonalization.)

Deeper reading. LADR 7D (isometries and unitary operators); Strang, ch. "Complex Vectors and Matrices" (the Fast Fourier Transform); Osgood, *Lectures on the Fourier Transform and Its Applications*, chapter on the DFT.

13 The Singular Value Decomposition

NEEDED FOR: Fourier-adjacent; statistical inference (PCA). Not needed for ML.

The spectral theorem needs a self-adjoint square matrix. The SVD is its extension to *every* matrix, square or not — LADR Chapter 7's finale, and the workhorse behind `lstsq`, `matrix_rank`, PCA, and compression.

13.1 Theory

Theorem 13.1 (Singular value decomposition, LADR 7.70)**.** *Let A be a real $m \times n$ matrix of rank r. Then there exist orthonormal vectors $v_1, \dots, v_r \in \mathbb{R}^n$, orthonormal vectors $u_1, \dots, u_r \in \mathbb{R}^m$, and numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ (the* singular values*) such that*

$$A = \sigma_1 u_1 v_1^{\mathsf{T}} + \cdots + \sigma_r u_r v_r^{\mathsf{T}}, \quad \textit{equivalently} \quad Av_i = \sigma_i u_i \ (i \leq r), \quad Av = 0 \textit{ for } v \perp v_1, \dots, v_r.$$

(The same holds over $\mathbb{C}$ with $*$ in place of T.)

Proof. The matrix $A^{\mathsf{T}}A$ is symmetric and PSD (Lesson 11: $\langle A^{\mathsf{T}}Ax, x\rangle = \|Ax\|^2 \geq 0$). By the spectral theorem it has an orthonormal eigenbasis $v_1, \dots, v_n$ of $\mathbb{R}^n$ with real eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$; write $\lambda_i = \sigma_i^2$ with $\sigma_i \geq 0$, and let r' be the number of nonzero σ_i.

Key computation. For any i, j:

$$\langle Av_i,\, Av_j\rangle = \langle A^{\mathsf{T}}Av_i,\, v_j\rangle = \sigma_i^2\langle v_i,\, v_j\rangle = \begin{cases} \sigma_i^2 & i = j \\ 0 & i \neq j. \end{cases}$$

So the vectors $Av_1, \dots, Av_n$ are mutually orthogonal with $\|Av_i\| = \sigma_i$; in particular $Av_i = 0$ exactly when $\sigma_i = 0$. For $i \leq r'$ define $u_i = Av_i/\sigma_i$: unit vectors, orthonormal by the computation above.

The formula. Both sides of the claimed identity are linear maps; by Theorem 4.3 it suffices to check them on the basis $v_1, \dots, v_n$. For $j \leq r'$: the right side gives $\sum_i \sigma_i u_i (v_i^{\mathsf{T}} v_j) = \sigma_j u_j = Av_j$. For $j > r'$: the right side gives $0 = Av_j$. They agree.

Rank. range A is spanned by the Av_j, i.e. by $\sigma_1 u_1, \dots, \sigma_{r'} u_{r'}$, an orthogonal (hence independent) list — so $\operatorname{rank} A = r'$, forcing $r' = r$. □

In matrix form, padding the v's and u's to full orthonormal bases: $A = U\Sigma V^{\mathsf{T}}$ with U $(m \times m)$ and V $(n \times n)$ orthogonal and Σ $(m \times n)$ diagonal with the σ_i's. Read as a pipeline, every matrix is: rotate the input frame (V^{T}), stretch each axis by σ_i and embed into the output space (Σ), rotate the output frame (U). Rotation–stretch–rotation, no exceptions.

Corollary 13.2 (What the SVD reveals at a glance)**.** *With notation as above:*

(a) $\operatorname{rank} A$ *= number of nonzero singular values (this is what* `matrix_rank` *counts, with a tolerance);*

(b) $\|Av\| \le \sigma_1 \|v\|$ *for all* v*, with equality at* $v = v_1$*:* σ_1 *is the operator's maximum stretch factor;*

(c) the null space of A *is* $\operatorname{span}(v_{r+1}, \dots, v_n)$ *and the range is* $\operatorname{span}(u_1, \dots, u_r)$ *— the SVD hands you orthonormal bases for both of Lesson 5's subspaces.*

Proof. (a) was shown in the proof. (c): $Av = 0$ iff all eigen-coordinates of v along $v_1, \dots, v_r$ vanish (by the orthogonality of the Av_i and $\|Av_i\| = \sigma_i > 0$); the range statement was the last line of the proof. (b): expand $v = \sum_i c_i v_i$; then $Av = \sum_{i \le r} c_i \sigma_i u_i$, and by orthonormality of the u_i (Pythagoras, repeatedly):

$$\|Av\|^2 = \sum_{i \le r} \sigma_i^2 c_i^2 \le \sigma_1^2 \sum_i c_i^2 = \sigma_1^2 \|v\|^2,$$

with equality when all weight sits on c_1. □

Dropping all but the k largest terms of $\sum_i \sigma_i u_i v_i^{\mathsf{T}}$ gives the rank-k matrix A_k closest to A (Eckart–Young theorem; statement worth knowing, proof omitted) — the principle behind PCA and lossy compression: keep the directions with the biggest stretch, discard the rest.

13.2 Practice: by hand

Example 13.3 (A full 2×2 SVD)**.** $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ (the shear that refused to diagonalize in Lesson 10 — but nothing refuses an SVD). Compute

$$A^{\mathsf{T}} A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}: \quad (1-\lambda)(2-\lambda) - 1 = 0 \iff \lambda^2 - 3\lambda + 1 = 0 \iff \lambda = \tfrac{3 \pm \sqrt{5}}{2}.$$

So $\sigma_1 = \sqrt{(3+\sqrt{5})/2} \approx 1.618$ and $\sigma_2 = \sqrt{(3-\sqrt{5})/2} \approx 0.618$ (the golden ratio and its reciprocal!). Eigenvector for λ_1: solve $(1-\lambda_1)a + b = 0$, giving $v_1 \propto (1, \lambda_1 - 1)$; then $u_1 = Av_1/\sigma_1$. The shear stretches one tilted direction by φ and compresses an orthogonal one by $1/\varphi$ — invisible to eigen-analysis, plain in the SVD.

Fourier connection. SVD is Fourier-adjacent rather than Fourier-core: it appears when Fourier methods meet data — e.g., analyzing which frequency patterns dominate a family of signals (PCA of spectrograms), or characterizing the maximum gain σ_1 of a non-shift-invariant system where the convolution theorem does not apply. It also closes a Part I loop: `lstsq` (Lesson 6) is implemented via SVD because Corollary 13.2 exposes range and null space in orthonormal coordinates, making the projection of b trivial even when columns are dependent. (In statistical inference it returns as PCA: Lesson 23 shows the covariance matrix's eigendecomposition *is* the SVD story for data.)

13.3 Exercises

Exercise 13.1 (Hand)**.** Compute the SVD ingredients of $A = \left(\begin{smallmatrix} 3 & 0 \\ 0 & -2 \end{smallmatrix}\right)$: singular values, v_i's, u_i's. (Careful: singular values are positive; where did the sign of -2 go?) Compare with the eigenvalues from Exercise 10.1(a).

Exercise 13.2 (Hand)**.** For $A = \left(\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right)$ (rank 1), find σ_1, u_1, v_1 and write $A = \sigma_1 u_1 v_1^{\mathsf{T}}$ exactly.

Exercise 13.3 (Proof, $\star$)**.** Prove that for symmetric PSD A, the singular values equal the eigenvalues and one may take $u_i = v_i$. Then show by example (Exercise 13.1!) that for symmetric A with a negative eigenvalue, singular values are the absolute values of eigenvalues.

Exercise 13.4 (Proof)**.** Using Corollary 13.2(b) applied to A and to A^{-1} (square invertible case), show $\|Ax\|/\|x\|$ ranges over $[\sigma_n, \sigma_1]$, and interpret $\kappa = \sigma_1/\sigma_n$ as the worst-case ratio by which A distorts relative lengths. (κ is the *condition number*; `np.linalg.cond` computes it, and it governs how much rounding error `solve` can amplify.)

Deeper reading. LADR 7E–7F (SVD and its consequences — Axler's proof is the one given here), 7C (positive operators and square roots); Strang, ch. "The Singular Value Decomposition."

14 Determinants and Trace

NEEDED FOR: Fourier analysis (stretch theorem); general background. Not needed for ML.

Axler famously postpones determinants to his final chapter because nothing earlier needed them — and indeed nothing earlier *here* needed them. But they earn their place: as the volume-scaling factor of a linear map (which is exactly how they enter multivariable integrals and multidimensional Fourier transforms) and as a compact certificate of invertibility.

14.1 Theory

Definition 14.1 (Determinant, characterized)**.** The determinant is the unique function $\det : (n \times n \text{ matrices}) \to \mathbb{F}$ that, viewed as a function of the n columns, is

(i) *multilinear*: linear in each column separately, the others held fixed;

(ii) *alternating*: $\det = 0$ whenever two columns are equal;

(iii) *normalized*: $\det I = 1$.

Existence and uniqueness are LADR 9B–9C; uniqueness forces the explicit permutation formula

$$\det A = \sum_{\pi} \operatorname{sign}(\pi)\, A_{\pi(1),1} \cdots A_{\pi(n),n},$$

summed over the $n!$ permutations π of $\{1, \dots, n\}$. For $n = 2, 3$ this yields the familiar

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc, \qquad \det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - ceg - bdi - afh.$$

Two consequences of (i)–(ii), each one line: swapping two columns flips the sign of det (expand det with both columns replaced by their sum, and cancel using (ii)); and adding a multiple of one column to another leaves det unchanged (multilinearity splits it into det A plus a determinant with a repeated column). The same statements hold for rows, since $\det A = \det A^{\mathsf{T}}$ (visible from the permutation formula: transposing reindexes the sum by inverse permutations, which have the same signs).

Theorem 14.2 (Multiplicativity)**.** $\det(AB) = \det(A)\det(B)$.

Proof. Fix A and consider $f(B) = \det(AB)$ as a function of the columns of B. Column j of AB is Ab_j (Lesson 4), so f is multilinear in the columns of B (because $b_j \mapsto Ab_j$ is linear and det is multilinear) and alternating (equal columns $b_i = b_j$ give equal columns $Ab_i = Ab_j$). A short argument from uniqueness — any multilinear alternating function is a scalar multiple of det, the scalar being its value at I — gives $f(B) = f(I)\det(B) = \det(A)\det(B)$. □

Theorem 14.3 (Determinant as invertibility certificate and eigenvalue product)**.** *(a) A is invertible $\iff$ $\det A \neq 0$; in that case $\det(A^{-1}) = 1/\det A$.*

(b) If A (over $\mathbb{C}$) has eigenvalues $\lambda_1, \ldots, \lambda_n$ listed with multiplicity, then $\det A = \lambda_1 \cdots \lambda_n$; consequently λ is an eigenvalue of A iff $\det(A - \lambda I) = 0$ — the characteristic polynomial route used informally in Example 10.7.

Proof. (a) If A is invertible, $1 = \det I = \det(AA^{-1}) = \det A \det A^{-1}$ by Theorem 14.2, so $\det A \neq 0$ and the reciprocal formula holds. If A is not invertible, its columns are dependent (Lesson 5), so some column is a combination of the others (dependence lemma); subtracting that combination from it — which does not change det — produces a zero column, and multilinearity with the zero column gives $\det A = 0$.

(b) Over $\mathbb{C}$, every matrix is similar to an upper-triangular one: $A = STS^{-1}$ with T upper triangular whose diagonal lists the eigenvalues with multiplicity (LADR 5C, from Theorem 10.4 by induction; statement used here without the inductive details). By multiplicativity, $\det A = \det S \det T \det S^{-1} = \det T$, and the determinant of a triangular matrix is the product of its diagonal entries — expand by the permutation formula: only the identity permutation avoids all the zero entries below the diagonal. Hence $\det A = \lambda_1 \cdots \lambda_n$. Applying this to $A - \lambda I$, whose eigenvalues are $\lambda_i - \lambda$: it is singular iff some $\lambda_i = \lambda$ iff its determinant $\prod_i (\lambda_i - \lambda)$ vanishes. □

Volume. $|\det A|$ is the factor by which the map $x \mapsto Ax$ scales n-dimensional volume: the unit square (cube) maps to the parallelogram (parallelepiped) spanned by the columns of A, whose volume is $|\det A|$. Properties (i)–(iii) are exactly the axioms signed volume must satisfy (scale one edge, the volume scales; degenerate flat shapes have volume zero; the unit cube has volume 1), which is the honest reason the determinant is defined this way. The sign records orientation: negative when the map flips handedness.

Definition 14.4 (Trace)**.** $\operatorname{tr} A = \sum_i A_{i,i}$, the sum of diagonal entries.

Proposition 14.5. *(a) $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ for any A ($m \times n$) and B ($n \times m$). (b) Over $\mathbb{C}$, $\operatorname{tr} A = \lambda_1 + \cdots + \lambda_n$ (eigenvalues with multiplicity).*

Proof. (a) Both sides equal $\sum_i \sum_j A_{i,j} B_{j,i}$ — expand each definition and swap the order of summation. (b) With $A = STS^{-1}$ as above: $\operatorname{tr} A = \operatorname{tr}\big((ST)S^{-1}\big) = \operatorname{tr}\big(S^{-1}(ST)\big) = \operatorname{tr} T = \sum_i \lambda_i$, using (a) with the grouping shown. □

Trace and determinant are the two coefficients of the characteristic polynomial you can read off instantly; for 2×2: $\det(A - \lambda I) = \lambda^2 - (\operatorname{tr} A)\lambda + \det A$, so eigenvalues of a 2×2 are $\frac{\operatorname{tr} A \pm \sqrt{(\operatorname{tr} A)^2 - 4 \det A}}{2}$ — worth memorizing.

14.2 Practice: by hand

Example 14.6. $A = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$: $\operatorname{tr} A = 4$, $\det A = 3$; eigenvalues solve $\lambda^2 - 4\lambda + 3 = 0$: $\lambda = 1, 3$. Matches Example 10.7, with product $3 = \det$ and sum $4 = \operatorname{tr}$. ✓

Example 14.7 (Volume scaling)**.** The map $A = \left(\begin{smallmatrix} 2 & 1 \\ 0 & 3 \end{smallmatrix}\right)$ sends the unit square to the parallelogram spanned by $(2, 0)$ and $(1, 3)$: base 2, height 3, area $6 = \det A$. The shear part (the 1) moved the top edge sideways without changing the area — exactly the "add a multiple of one column to another" invariance.

Fourier connection. Determinants enter Fourier analysis exactly once, but at a memorable moment: the higher-dimensional Fourier transform. For $f : \mathbb{R}^n \to \mathbb{C}$ and an invertible A, the transform of the stretched signal $f(Ax)$ is $\frac{1}{|\det A|} \hat{f}(A^{-\mathsf{T}} \xi)$ — the general stretch theorem. The $|\det A|$ is the volume-scaling factor from the change of variables in the integral, and $A^{-\mathsf{T}}$ is why frequency space transforms "contragradiently" (images stretched in space compress in frequency, along dual axes). When Osgood derives this for 2-D images and CT scans, you will know exactly where every symbol comes from.

14.3 Exercises

Exercise 14.1 (Hand)**.** Compute by row reduction, tracking swaps and column operations' effects: $\det \begin{pmatrix} 1 & 2 & 0 \\ 2 & 4 & 1 \\ 1 & 0 & 3 \end{pmatrix}$. Cross-check with the 3×3 formula.

Exercise 14.2 (Hand)**.** Using trace and det only, find the eigenvalues of $\left(\begin{smallmatrix} 5 & 4 \\ 1 & 2 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$ (the latter over $\mathbb{C}$; interpret).

Exercise 14.3 (Proof, $\star$)**.** From the three defining properties alone (no permutation formula), derive $\det \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = ad - bc$: expand $\det\big((a, c), (b, d)\big)$ by multilinearity into four determinants of standard-basis columns, and evaluate each using alternation and normalization.

Exercise 14.4 (Proof)**.** Prove that similar matrices ($B = S^{-1}AS$) have the same determinant, trace, and eigenvalues — so all three are properties of the underlying linear map, independent of basis.

Deeper reading. LADR 9A (bilinear and quadratic forms), 9B (alternating multilinear forms), 9C (determinants — the characterization above); for the skipped Chapter 8 (generalized eigenvectors, i.e. what replaces diagonalization when it fails, and the trace's operator-theoretic role in 8D): read after the Fourier work if the shear example left you curious — no Fourier topic requires it. Strang, ch. "Determinants."

15 Capstone II: The DFT in NumPy

NEEDED FOR: Fourier analysis

Forty-five minutes; everything in Part II at once. Type it.

Task. Build the DFT matrix from scratch, certify it is (scaled-)unitary, use it to denoise a signal, and confirm the convolution theorem against a hand-rolled circular convolution.

```
import numpy as np
rng = np.random.default_rng(0)

# --- 1. The matrix and its unitarity (Lessons 9, 12) ---
N = 128
n = np.arange(N)
W = np.exp(-2j * np.pi * np.outer(n, n) / N)
F = W / np.sqrt(N)
assert np.allclose(F.conj().T @ F, np.eye(N), atol=1e-10)  # unitary

# --- 2. A noisy two-tone signal ---
t = n / N
clean = np.sin(2*np.pi*5*t) + 0.5*np.sin(2*np.pi*12*t)
x = clean + 0.4 * rng.standard_normal(N)

# --- 3. Analyze: Parseval + peak-finding (Lessons 11, 12) ---
xh = W @ x                          # same as np.fft.fft(x)
assert np.allclose((np.abs(x)**2).sum(),
                   (np.abs(xh)**2).sum() / N, atol=1e-8)  # Parseval
peaks = np.argsort(np.abs(xh[:N//2]))[-2:]
print("dominant frequencies:", sorted(peaks))             # [5, 12]

# --- 4. Filter: keep low frequencies + mirror band ---
keep = (n <= 15) | (n >= N - 15)
y = np.real(np.conj(W).T @ (xh * keep) / N)   # inverse DFT
print("noise before/after:",
      np.linalg.norm(x - clean), np.linalg.norm(y - clean))

# --- 5. Convolution theorem (Lesson 12) ---
c = np.zeros(N); c[:4] = 0.25                  # moving-average filter
conv_direct = np.array([(c * x[(k - n) % N]).sum() for k in range(N)])
conv_fft    = np.real(np.fft.ifft(np.fft.fft(c) * np.fft.fft(x)))
assert np.allclose(conv_direct, conv_fft, atol=1e-9)
```

Checkpoints.

1. The unitarity assertion passes: columns of F are orthonormal in the complex inner product — Lemma 9.2 at work (change a `conj()` to see it fail).
2. Parseval holds to 10^{-8}: the DFT moves energy between representations without creating or destroying it (Theorem 12.4).

3. The two injected tones are the two largest $|\hat{x}_k|$ despite the noise — orthogonality is what lets each frequency's coefficient ignore the others.

4. Denoising by zeroing high-frequency coefficients is a *projection* (Theorem 6.3) onto the span of 31 Fourier basis vectors — Part I's least-squares geometry, in Part II's basis. The output is real only because the mirror band was kept (Exercise 12.5).

5. Direct circular convolution ($O(N^2)$ as written) and the FFT route ($O(N \log N)$) agree to 10^{-9}: Theorem 12.6, numerically certified. Re-run with $N = 4096$ and time both.

Final self-check list for Part II

From memory: compute with complex exponentials and state the roots-of-unity lemma; find eigenvalues/eigenvectors of a 2×2 and diagonalize it; state and prove that self-adjoint matrices have real eigenvalues and orthogonal eigenspaces; state the spectral theorem and the three equivalent faces of unitarity; write the DFT matrix, prove $W^*W = NI$, and state the convolution theorem; explain what the SVD's rotation–stretch–rotation says about an arbitrary matrix; compute 2×2 and 3×3 determinants and eigenvalues via trace/det. That is LADR Chapters 5–7 and 9 at working strength — and the linear-algebra spine of Fourier analysis.

Part III: The Probability Core

Machine learning assumes a first course in probability, and it collects on the debt: Bayes' rule and conditional independence power the Bayesian-network and HMM lectures, expectations define the utilities that MDPs and expectimax maximize, and every training loss is an expectation over data. This part is the minimum working dose, following Bertsekas & Tsitsiklis (B&T), *Introduction to Probability* (2nd ed.), Chapters 1–3, in the same style as Part I. Statistical inference assumes all of this as a floor, so nothing here is wasted on the longer road either.

16 Probability Models, Conditioning, and Bayes' Rule

NEEDED FOR: Machine learning (core: Bayesian networks, HMMs); statistical inference (assumed)

16.1 Theory

Definition 16.1 (Probability model)**.** A *probabilistic model* consists of a *sample space* Ω — the set of all possible outcomes of an experiment, exactly one of which occurs — and a *probability law* $\mathbf{P}$ assigning to each event $A \subseteq \Omega$ a number $\mathbf{P}(A)$, satisfying the axioms:

(i) *nonnegativity*: $\mathbf{P}(A) \geq 0$ for every event A;

(ii) *additivity*: if $A_1, A_2, \ldots$ are disjoint events, then $\mathbf{P}(A_1 \cup A_2 \cup \cdots) = \mathbf{P}(A_1) + \mathbf{P}(A_2) + \cdots$;

(iii) *normalization*: $\mathbf{P}(\Omega) = 1$.

Everything else in Parts III–IV is a consequence of these three axioms — the same game as Lesson 1's vector-space axioms, and worth playing for the same reason.

Proposition 16.2 (First consequences)**.** *For events A, B:*

(a) $\mathbf{P}(\varnothing) = 0$ *and* $\mathbf{P}(A^c) = 1 - \mathbf{P}(A)$*;*

(b) if $A \subseteq B$ *then* $\mathbf{P}(A) \leq \mathbf{P}(B)$ *(monotonicity);*

(c) $\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B) - \mathbf{P}(A \cap B)$*, hence* $\mathbf{P}(A \cup B) \leq \mathbf{P}(A) + \mathbf{P}(B)$ *(the union bound).*

Proof. (a) Ω and $\varnothing$ are disjoint with union Ω, so $1 = \mathbf{P}(\Omega) = \mathbf{P}(\Omega) + \mathbf{P}(\varnothing)$, giving $\mathbf{P}(\varnothing) = 0$; A and A^c are disjoint with union Ω, so $\mathbf{P}(A) + \mathbf{P}(A^c) = 1$.

(b) $B = A \cup (B \cap A^c)$, a disjoint union, so $\mathbf{P}(B) = \mathbf{P}(A) + \mathbf{P}(B \cap A^c) \geq \mathbf{P}(A)$ by nonnegativity.

(c) Write $A \cup B$ as the disjoint union $A \cup (B \cap A^c)$ and B as the disjoint union $(A \cap B) \cup (B \cap A^c)$. Then

$$\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B \cap A^c) = \mathbf{P}(A) + \big(\mathbf{P}(B) - \mathbf{P}(A \cap B)\big),$$

and the inequality follows since $\mathbf{P}(A \cap B) \geq 0$. □

When Ω is finite with equally likely outcomes, $\mathbf{P}(A) = |A|/|\Omega|$: probability is counting. (B&T Chapter 1.6 develops the counting toolkit — permutations, combinations $\binom{n}{k}$ — which a first probability course drills; we use $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ freely below.)

Definition 16.3 (Conditional probability)**.** For events A, B with $\mathbf{P}(B) > 0$, the *conditional probability of A given B* is

$$\mathbf{P}(A \mid B) = \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(B)}.$$

Conditioning is re-normalization: the outcomes in B become the new sample space, with probabilities scaled up by $1/\mathbf{P}(B)$ so they again sum to 1. One checks directly that $\mathbf{P}(\cdot \mid B)$ satisfies the three axioms — so *every* theorem about probability laws applies verbatim to conditional laws. This innocuous remark is used constantly.

Theorem 16.4 (Multiplication rule, total probability, Bayes' rule)**.** *Let $A_1, \ldots, A_n$ be disjoint events whose union is Ω (a* partition*), each with $\mathbf{P}(A_i) > 0$, and let B be an event with $\mathbf{P}(B) > 0$.*

(a) Multiplication rule: $\mathbf{P}(A \cap B) = \mathbf{P}(B)\,\mathbf{P}(A \mid B) = \mathbf{P}(A)\,\mathbf{P}(B \mid A)$ *(whenever the conditioning events have positive probability); more generally*

$$\mathbf{P}(A_1 \cap \cdots \cap A_n) = \mathbf{P}(A_1)\,\mathbf{P}(A_2 \mid A_1) \cdots \mathbf{P}(A_n \mid A_1 \cap \cdots \cap A_{n-1}).$$

(b) Total probability theorem: $\mathbf{P}(B) = \sum_{i=1}^{n} \mathbf{P}(A_i)\,\mathbf{P}(B \mid A_i)$.

(c) Bayes' rule:

$$\mathbf{P}(A_i \mid B) = \frac{\mathbf{P}(A_i)\,\mathbf{P}(B \mid A_i)}{\sum_{j=1}^{n} \mathbf{P}(A_j)\,\mathbf{P}(B \mid A_j)}.$$

Proof. (a) is the definition of conditional probability rearranged; the general form follows by induction, multiplying one more conditional probability at each step (each factor's numerator cancels the previous factor's intersection).

(b) The sets $B \cap A_1, \ldots, B \cap A_n$ are disjoint (the A_i are) and their union is B (the A_i cover Ω). By additivity and then the multiplication rule,

$$\mathbf{P}(B) = \sum_i \mathbf{P}(B \cap A_i) = \sum_i \mathbf{P}(A_i)\,\mathbf{P}(B \mid A_i).$$

(c) By definition and then (a) for the numerator and (b) for the denominator:

$$\mathbf{P}(A_i \mid B) = \frac{\mathbf{P}(A_i \cap B)}{\mathbf{P}(B)} = \frac{\mathbf{P}(A_i)\,\mathbf{P}(B \mid A_i)}{\sum_j \mathbf{P}(A_j)\,\mathbf{P}(B \mid A_j)}. \qquad \square$$

Bayes' rule is *inference*: the A_i are competing hypotheses ("causes"), B is the observed evidence ("effect"), and the rule converts the easy direction $\mathbf{P}(\text{effect} \mid \text{cause})$ — usually given by a model — into the wanted direction $\mathbf{P}(\text{cause} \mid \text{effect})$. The $\mathbf{P}(A_i)$ are *priors*, the $\mathbf{P}(A_i \mid B)$ *posteriors*.

Definition 16.5 (Independence)**.** Events A and B are *independent* if $\mathbf{P}(A \cap B) = \mathbf{P}(A)\,\mathbf{P}(B)$ (equivalently, when $\mathbf{P}(B) > 0$: $\mathbf{P}(A \mid B) = \mathbf{P}(A)$ — learning B tells you nothing about A). Events A and B are *conditionally independent given C* if $\mathbf{P}(A \cap B \mid C) = \mathbf{P}(A \mid C)\,\mathbf{P}(B \mid C)$.

Two warnings, both of which machine learning's graphical-models lectures weaponize: independence does *not* survive conditioning (two independent coin flips become dependent given "the two flips differ"), and conditional independence does not imply independence (nor conversely). Independence is a modeling *assumption* about the world, not something you can usually read off the numbers.

16.2 Practice: by hand

Example 16.6 (Two rolls of a fair die)**.** $\Omega = \{(i,j) : 1 \le i,j \le 6\}$, all 36 outcomes equally likely. Let $A =$ "the sum is 8" $= \{(2,6),(3,5),(4,4),(5,3),(6,2)\}$, so $\mathbf{P}(A) = 5/36$. Let $B =$ "the first roll is even" (18 outcomes, $\mathbf{P}(B) = 1/2$). Then $A \cap B = \{(2,6),(4,4),(6,2)\}$, so

$$\mathbf{P}(A \mid B) = \frac{3/36}{18/36} = \frac{1}{6} > \frac{5}{36} = \mathbf{P}(A) :$$

evidence that the first roll is even makes a sum of 8 slightly *more* likely — A and B are dependent.

Example 16.7 (The false-positive puzzle — Bayes' rule you must own)**.** A test for a condition affecting 1 in 1000 people is 99% accurate in both directions: $\mathbf{P}(+ \mid \text{sick}) = 0.99$ and $\mathbf{P}(- \mid \text{healthy}) = 0.99$. You test positive. The probability you are sick is *not* 0.99; by Bayes' rule with the partition $\{\text{sick}, \text{healthy}\}$:

$$\mathbf{P}(\text{sick} \mid +) = \frac{0.001 \times 0.99}{0.001 \times 0.99 + 0.999 \times 0.01} = \frac{0.00099}{0.00099 + 0.00999} \approx 0.09.$$

About 9%: the false positives from the huge healthy population swamp the true positives from the tiny sick one. The prior matters as much as the test.

Example 16.8 (Conditional independence, concretely)**.** Two coins: one fair, one with $\mathbf{P}(\text{heads}) = 0.9$. Pick one at random and flip it twice. Given *which* coin was picked, the flips are independent. Unconditionally they are not: a first head is evidence for the biased coin, which makes a second head more likely. Compute: $\mathbf{P}(H_1) = \frac{1}{2}(0.5) + \frac{1}{2}(0.9) = 0.7$, while

$$\mathbf{P}(H_2 \mid H_1) = \frac{\mathbf{P}(H_1 \cap H_2)}{\mathbf{P}(H_1)} = \frac{\frac{1}{2}(0.25) + \frac{1}{2}(0.81)}{0.7} = \frac{0.53}{0.7} \approx 0.757 > 0.7.$$

This two-coin experiment is a two-node Bayesian network, and the calculation you just did is exactly machine learning's "inference by enumeration."

> **Machine-learning connection.** A Bayesian network *is* the multiplication rule with a conditional-independence assumption: the joint distribution factors as $\mathbf{P}(X_1,\dots,X_n) = \prod_i \mathbf{P}(X_i \mid \text{Parents}(X_i))$. Inference in the network — filtering in an HMM, diagnosing from symptoms — is Bayes' rule plus total probability, applied systematically. Example 16.7 *is* a two-variable Bayesian network. When machine learning says "condition on the evidence and renormalize," it is invoking the fact that $\mathbf{P}(\cdot \mid B)$ is a probability law.

16.3 Exercises

Exercise 16.1 (Hand)**.** Three fair coin flips. List the sample space. Let $A =$ "at least two heads" and $B =$ "the first flip is heads." Compute $\mathbf{P}(A)$ and $\mathbf{P}(A \mid B)$, and check whether A and B are independent.

Exercise 16.2 (Hand)**.** A spam filter knows: 30% of email is spam, the word "free" appears in 40% of spam and 5% of non-spam. Apply Bayes' rule to find $\mathbf{P}(\text{spam} \mid \text{"free"})$. Then recompute with a prior of 80% spam and note how the posterior moves — this is Capstone III in miniature.

Exercise 16.3 (Proof, $\star$)**.** Prove the general union bound $\mathbf{P}\big(\bigcup_{i=1}^{n} A_i\big) \le \sum_{i=1}^{n} \mathbf{P}(A_i)$ by induction from Proposition 16.2(c). (This one-line tool underlies half of the analyses in machine learning theory.)

Exercise 16.4 (Proof)**.** Verify that $\mathbf{P}(\cdot \mid B)$ satisfies the three probability axioms. Then prove the "chain rule with a spectator": $\mathbf{P}(A \cap C \mid B) = \mathbf{P}(C \mid B)\,\mathbf{P}(A \mid B \cap C)$ whenever the conditioning events have positive probability.

Exercise 16.5 (Proof)**.** Show that if A and B are independent, so are A and B^c, and so are A^c and B^c.

Deeper reading. B&T 1.1–1.2 (models and axioms), 1.3–1.4 (conditioning, total probability, Bayes), 1.5 (independence), 1.6 (counting); the machine learning "Bayesian networks" module notes re-derive Theorem 16.4 in network notation.

17 Discrete Random Variables and Expectation

NEEDED FOR: Machine learning (core: utilities, expectimax, losses); statistical inference (assumed)

17.1 Theory

Definition 17.1 (Random variable, PMF)**.** A *random variable* is a function $X : \Omega \to \mathbb{R}$ — a number determined by the outcome. It is *discrete* if its range is finite or countable. The *probability mass function* (PMF) of a discrete X is

$$p_X(x) = \mathbf{P}(X = x) \qquad \big(\text{shorthand for } \mathbf{P}(\{\omega \in \Omega : X(\omega) = x\})\big),$$

and satisfies $p_X(x) \ge 0$ and $\sum_x p_X(x) = 1$ (additivity over the disjoint events $\{X = x\}$).

The four discrete distributions to know cold:

- *Bernoulli(p)*: $X \in \{0, 1\}$ with $p_X(1) = p$. Models one yes/no trial.
- *Binomial(n, p)*: X = number of successes in n independent Bernoulli(p) trials; $p_X(k) = \binom{n}{k} p^k (1-p)^{n-k}$ for $k = 0, \dots, n$.
- *Geometric(p)*: X = number of trials up to and including the first success; $p_X(k) = (1-p)^{k-1} p$ for $k = 1, 2, \dots$.
- *Poisson(λ)*: $p_X(k) = e^{-\lambda}\lambda^k / k!$ for $k = 0, 1, 2, \dots$; the $n \to \infty$, $np \to \lambda$ limit of the binomial (Lesson 25 uses this).

Definition 17.2 (Expectation, variance)**.** The *expectation* (mean) of a discrete random variable X is

$$\mathbf{E}[X] = \sum_x x\, p_X(x)$$

(assuming the sum converges absolutely), and its *variance* is

$$\operatorname{var}(X) = \mathbf{E}\big[(X - \mathbf{E}[X])^2\big] \ge 0, \qquad \text{with } \textit{standard deviation } \sigma_X = \sqrt{\operatorname{var}(X)}.$$

Theorem 17.3 (Expected value rule / LOTUS)**.** *For any function $g : \mathbb{R} \to \mathbb{R}$,*

$$\mathbf{E}[g(X)] = \sum_x g(x)\, p_X(x).$$

Proof. Let $Y = g(X)$; Y is discrete with $p_Y(y) = \sum_{x:\, g(x)=y} p_X(x)$ (additivity over the disjoint events $\{X = x\}$ composing $\{Y = y\}$). Then

$$\mathbf{E}[Y] = \sum_y y\, p_Y(y) = \sum_y \sum_{x:\, g(x)=y} y\, p_X(x) = \sum_y \sum_{x:\, g(x)=y} g(x)\, p_X(x) = \sum_x g(x)\, p_X(x),$$

since every x appears in exactly one inner sum. □

The name LOTUS — "law of the unconscious statistician" — records that people apply it without noticing it needs proof: the naive move (average $g(x)$ against p_X, without ever computing $p_{g(X)}$) is correct.

Theorem 17.4 (Linearity of expectation)**.** *For random variables X, Y on the same experiment and constants a, b:*

$$\mathbf{E}[aX + b] = a\,\mathbf{E}[X] + b, \qquad \mathbf{E}[X + Y] = \mathbf{E}[X] + \mathbf{E}[Y].$$

The second identity requires no *independence.*

Proof. The first is LOTUS with $g(x) = ax+b$: $\sum_x (ax+b)p_X(x) = a\sum_x x p_X(x) + b\sum_x p_X(x) = a\mathbf{E}[X] + b$. For the second, work on the underlying sample space (for simplicity, countable Ω, writing $p(\omega) = \mathbf{P}(\{\omega\})$):

$$\mathbf{E}[X + Y] = \sum_\omega \big(X(\omega) + Y(\omega)\big)p(\omega) = \sum_\omega X(\omega)p(\omega) + \sum_\omega Y(\omega)p(\omega) = \mathbf{E}[X] + \mathbf{E}[Y]. \qquad \square$$

Proposition 17.5 (Variance identities)**.** $\operatorname{var}(X) = \mathbf{E}[X^2] - (\mathbf{E}[X])^2$*, and* $\operatorname{var}(aX + b) = a^2 \operatorname{var}(X)$*.*

Proof. Write $\mu = \mathbf{E}[X]$. Expanding the square and using linearity,

$$\operatorname{var}(X) = \mathbf{E}[X^2 - 2\mu X + \mu^2] = \mathbf{E}[X^2] - 2\mu\mathbf{E}[X] + \mu^2 = \mathbf{E}[X^2] - \mu^2.$$

For the second: $aX+b$ has mean $a\mu+b$, so $\operatorname{var}(aX+b) = \mathbf{E}[(aX+b-a\mu-b)^2] = \mathbf{E}[a^2(X-\mu)^2] = a^2 \operatorname{var}(X)$ — the shift b disappears, as it should: shifting a distribution does not change its spread. □

Example 17.6 (The catalog, computed)**.** *Bernoulli(p):* $\mathbf{E}[X] = p$, and $\mathbf{E}[X^2] = p$ (since $X^2 = X$), so $\operatorname{var}(X) = p - p^2 = p(1-p)$ — maximized at $p = \frac{1}{2}$, the most unpredictable coin.

Binomial(n, p): write $X = X_1 + \cdots + X_n$ as a sum of Bernoulli indicators. By linearity (no independence needed!): $\mathbf{E}[X] = np$. For the variance, independence *is* needed (Lesson 18): $\operatorname{var}(X) = np(1-p)$.

Geometric(p): $\mathbf{E}[X] = 1/p$. Slick derivation via the total expectation theorem (next lesson) or the derivative trick: for $|q| < 1$, differentiate $\sum_{k\geq 0} q^k = \frac{1}{1-q}$ to get $\sum_{k\geq 1} kq^{k-1} = \frac{1}{(1-q)^2}$, so with $q = 1-p$: $\mathbf{E}[X] = p\sum_{k\geq 1} kq^{k-1} = p \cdot \frac{1}{p^2} = \frac{1}{p}$.

Poisson(λ): $\mathbf{E}[X] = \sum_{k\geq 1} k\, e^{-\lambda}\frac{\lambda^k}{k!} = \lambda e^{-\lambda}\sum_{k\geq 1}\frac{\lambda^{k-1}}{(k-1)!} = \lambda$; a similar computation gives $\operatorname{var}(X) = \lambda$ too.

17.2 Practice: by hand

Example 17.7 (Expected utility, the machine learning shape)**.** A game: pay 1 to play; roll a die; win 4 if you roll a six, else nothing. Your net gain G has $p_G(3) = \frac{1}{6}$, $p_G(-1) = \frac{5}{6}$, so

$$\mathbf{E}[G] = 3 \cdot \tfrac{1}{6} + (-1) \cdot \tfrac{5}{6} = -\tfrac{1}{3}.$$

Play 100 times and linearity says your expected total is -33.3: no cleverness about dependence between rounds required. Decisions in machine learning (expectimax, MDP policies) are rankings of actions by exactly this kind of number.

Example 17.8 (Indicators make hard counts easy)**.** n people throw their hats in a pile and each takes one back uniformly at random. How many get their own hat, on average? Let X_i be the indicator that person i succeeds: $\mathbf{E}[X_i] = \mathbf{P}(X_i = 1) = 1/n$. The total is $X = \sum_i X_i$, so by linearity

$$\mathbf{E}[X] = \sum_{i=1}^{n} \frac{1}{n} = 1,$$

for *every* n — even though the X_i are dependent (if $n-1$ people got their own hats, so did the last). Linearity through dependence is the single most-used trick in average-case analysis.

Machine-learning connection. An MDP policy is evaluated by its *expected utility*; the value of a chance node in expectimax is $\sum_s \mathbf{P}(s)\, V(s)$ — an expectation; the training loss $\frac{1}{m}\sum_i \text{Loss}_i$ is the expectation of the loss under the empirical distribution of the data, and generalization asks how it relates to the expectation under the true distribution. When machine learning writes $V_\pi(s) = \sum_{s'} T(s,a,s')[\text{Reward} + \gamma V_\pi(s')]$, read it as: value = expected (reward + discounted future value). Expectation is machine learning's unit of account.

17.3 Exercises

Exercise 17.1 (Hand)**.** Let X be the number of heads in 3 fair flips. Tabulate p_X, compute $\mathbf{E}[X]$ and $\text{var}(X)$ from the table, and confirm both against the Binomial$(3, \frac{1}{2})$ formulas.

Exercise 17.2 (Hand)**.** A quiz has 5 multiple-choice questions, 4 options each; you guess uniformly and independently. Let X = number correct. What are $\mathbf{E}[X]$ and $\text{var}(X)$? What is $\mathbf{P}(X \geq 1)$? (Complement trick.)

Exercise 17.3 (Proof, $\star$)**.** Prove that $\mathbf{E}[X]$ minimizes the mean squared error: for any constant c, $\mathbf{E}[(X-c)^2] = \text{var}(X) + (c - \mathbf{E}[X])^2$, so the unique minimizing c is $\mathbf{E}[X]$. (Expand; linearity does the rest. Lesson 22 upgrades this from constants to functions of an observation — it becomes MMSE estimation.)

Exercise 17.4 (Proof)**.** Derive $\mathbf{E}[X] = 1/p$ for the geometric distribution using the memoryless recursion: condition on the first trial to get $\mathbf{E}[X] = 1 + (1-p)\mathbf{E}[X]$, and justify the conditioning step. (This is the total expectation theorem of Lesson 18 in action; compare with the derivative trick of Example 17.6.)

Deeper reading. B&T 2.1–2.4 (PMFs, expectation, variance; the distribution catalog); B&T 2.4 has the geometric-series variance computations spelled out.

18 Multiple Random Variables and Conditioning

NEEDED FOR: Machine learning (core: naive Bayes, HMMs, MDP value computations); statistical inference (assumed)

18.1 Theory

Definition 18.1 (Joint, marginal, conditional PMFs)**.** For discrete random variables X, Y on the same experiment, the *joint PMF* is $p_{X,Y}(x, y) = \mathbf{P}(X = x,\ Y = y)$. The *marginals* recover each variable alone:

$$p_X(x) = \sum_y p_{X,Y}(x, y), \qquad p_Y(y) = \sum_x p_{X,Y}(x, y)$$

(additivity over the partition by the other variable's value). For $p_Y(y) > 0$, the *conditional PMF* is

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x, y)}{p_Y(y)},$$

a genuine PMF in x for each fixed y. X and Y are *independent* if $p_{X,Y}(x, y) = p_X(x)\, p_Y(y)$ for all x, y.

All of Lesson 16 transfers: the multiplication rule reads $p_{X,Y}(x, y) = p_Y(y)\, p_{X|Y}(x \mid y)$, and Bayes' rule swaps which variable is conditioned on. Everything extends to three or more variables by adding coordinates.

Definition 18.2 (Conditional expectation)**.** $\mathbf{E}[X \mid Y = y] = \sum_x x\, p_{X|Y}(x \mid y)$: the mean of X in the re-normalized world where $Y = y$ is known.

Theorem 18.3 (Total expectation theorem)**.**

$$\mathbf{E}[X] = \sum_y p_Y(y)\, \mathbf{E}[X \mid Y = y].$$

Proof. Expand the right side using the definitions and the multiplication rule:

$$\sum_y p_Y(y) \sum_x x\, p_{X|Y}(x \mid y) = \sum_y \sum_x x\, p_{X,Y}(x, y) = \sum_x x \sum_y p_{X,Y}(x, y) = \sum_x x\, p_X(x) = \mathbf{E}[X],$$

where the swap of summation order is legal by absolute convergence. □

Read it as *divide and conquer for averages*: average within each scenario y, then average the scenario averages, weighted by how likely each scenario is. This theorem is the engine of every MDP value calculation and of Lesson 22.

Theorem 18.4 (Expectations and products under independence)**.** *If X and Y are independent, then*

$$\mathbf{E}[XY] = \mathbf{E}[X]\, \mathbf{E}[Y] \qquad \textit{and} \qquad \operatorname{var}(X + Y) = \operatorname{var}(X) + \operatorname{var}(Y).$$

Proof. For the product, using LOTUS on the pair (whose proof is identical to Theorem 17.3's) and factoring the joint PMF:

$$\mathbf{E}[XY] = \sum_{x,y} xy\, p_X(x) p_Y(y) = \Big(\sum_x x p_X(x)\Big)\Big(\sum_y y p_Y(y)\Big) = \mathbf{E}[X]\,\mathbf{E}[Y].$$

For the variance, let $\tilde{X} = X - \mathbf{E}[X]$ and $\tilde{Y} = Y - \mathbf{E}[Y]$ (still independent, with mean zero). Then

$$\operatorname{var}(X+Y) = \mathbf{E}[(\tilde{X}+\tilde{Y})^2] = \mathbf{E}[\tilde{X}^2] + 2\,\mathbf{E}[\tilde{X}\tilde{Y}] + \mathbf{E}[\tilde{Y}^2] = \operatorname{var}(X) + 2\,\mathbf{E}[\tilde{X}]\mathbf{E}[\tilde{Y}] + \operatorname{var}(Y),$$

and the cross term is $2 \cdot 0 \cdot 0 = 0$. □

Contrast with Theorem 17.4: means *always* add; variances add only under independence (Lesson 21 measures the failure by *covariance*). This completes the binomial variance claimed in Example 17.6: a sum of n independent Bernoulli(p)'s has variance $np(1-p)$.

18.2 Practice: by hand

Example 18.5 (A full joint-PMF workout)**.** Let (X, Y) have joint PMF given by the table (rows $x = 0, 1$; columns $y = 0, 1, 2$):

$p_{X,Y}$	$y=0$	$y=1$	$y=2$
$x=0$	0.1	0.2	0.1
$x=1$	0.2	0.1	0.3

Marginals: $p_X(0) = 0.4$, $p_X(1) = 0.6$; $p_Y = (0.3, 0.3, 0.4)$. Conditional: $p_{X|Y}(1 \mid 2) = 0.3/0.4 = 0.75$. Independence fails: $p_{X,Y}(1,2) = 0.3$ but $p_X(1)p_Y(2) = 0.24$. Conditional expectation: $\mathbf{E}[X \mid Y = 2] = 0.75$, while $\mathbf{E}[X] = 0.6$ — observing $Y = 2$ raises the estimate of X. Total expectation check: $0.3 \cdot \frac{0.2}{0.3} + 0.3 \cdot \frac{0.1}{0.3} + 0.4 \cdot 0.75 = 0.2 + 0.1 + 0.3 = 0.6 = \mathbf{E}[X]$. ✓

Example 18.6 (Naive Bayes, by hand)**.** Classify an email as spam ($C = 1$) or not ($C = 0$) from two binary features: F_1 = contains "free," F_2 = contains "meeting." Model: $\mathbf{P}(C = 1) = 0.3$, and *given the class*, features are conditionally independent with

$$\mathbf{P}(F_1 = 1 \mid C = 1) = 0.4, \quad \mathbf{P}(F_1 = 1 \mid C = 0) = 0.05,$$
$$\mathbf{P}(F_2 = 1 \mid C = 1) = 0.02, \quad \mathbf{P}(F_2 = 1 \mid C = 0) = 0.3.$$

Observe $F_1 = 1, F_2 = 0$. By Bayes' rule with the factored likelihood:

$$\mathbf{P}(C = 1 \mid F_1 = 1, F_2 = 0) = \frac{0.3 \times 0.4 \times 0.98}{0.3 \times 0.4 \times 0.98 + 0.7 \times 0.05 \times 0.7} = \frac{0.1176}{0.1176 + 0.0245} \approx 0.83.$$

"Free" without "meeting" pushes the posterior from 30% to 83% spam. The conditional-independence assumption is what let us multiply the per-feature likelihoods — with d features it reduces 2^d joint parameters to $2d$.

Machine-learning connection. Example 18.6 *is* a Bayesian network ($C \to F_1$, $C \to F_2$), and the computation is machine learning's inference-by-enumeration algorithm. An HMM adds time: hidden states form a chain (Lesson 26), each emitting an observation, and

filtering alternates the multiplication rule (predict) with Bayes' rule (update). The total expectation theorem is also how Bellman equations average over next states in MDPs. Master the two examples above and the graphical-models unit becomes bookkeeping.

18.3 Exercises

Exercise 18.1 (Hand)**.** Roll two fair dice; let $X =$ the minimum, $Y =$ the maximum. Tabulate the joint PMF for $X, Y \in \{1, 2, 3\}$ restricted to rolls of a 3-sided die (i.e. two fair 3-sided dice), find both marginals, and compute $\mathbf{E}[X \mid Y = 3]$.

Exercise 18.2 (Hand)**.** Extend Example 18.6: observe $F_1 = 1, F_2 = 1$. Compute the posterior. Explain in one sentence why the two features pull in opposite directions.

Exercise 18.3 (Proof, $\star$)**.** Prove the total expectation theorem in its conditional form: $\mathbf{E}[X \mid A] = \sum_y p_{Y|A}(y)\, \mathbf{E}[X \mid Y = y, A]$ for an event A with $\mathbf{P}(A) > 0$, by applying Theorem 18.3 to the conditional law $\mathbf{P}(\cdot \mid A)$ (recall: it satisfies the axioms, so every theorem applies to it).

Exercise 18.4 (Proof)**.** Show that if X and Y are independent, then so are $g(X)$ and $h(Y)$ for any functions g, h; conclude $\mathbf{E}[g(X)h(Y)] = \mathbf{E}[g(X)]\, \mathbf{E}[h(Y)]$.

Deeper reading. B&T 2.5–2.7 (joint PMFs, conditioning, independence); B&T 4.3 revisits conditional expectation as a random variable, which is Lesson 22's starting point.

19 Continuous Random Variables and the Normal

NEEDED FOR: Machine learning (light: Gaussian noise, continuous features); statistical inference (assumed, heavily)

19.1 Theory

Definition 19.1 (PDF, CDF)**.** A random variable X is *continuous* if there is a nonnegative function f_X (the *probability density function*, PDF) with

$$\mathbf{P}(a \le X \le b) = \int_a^b f_X(x)\, dx \quad \text{for all } a \le b, \qquad \int_{-\infty}^{\infty} f_X(x)\, dx = 1.$$

The *cumulative distribution function* (CDF) of any random variable is $F_X(x) = \mathbf{P}(X \le x)$; for continuous X, $F_X(x) = \int_{-\infty}^{x} f_X(t)\, dt$ and $f_X = F_X'$ wherever the derivative exists.

Two immediate consequences. First, $\mathbf{P}(X = x) = \int_x^x f_X = 0$ for every single point: individual values have probability zero, and only *intervals* carry probability — $f_X(x)$ is probability *per unit length* ($\mathbf{P}(x \le X \le x + \delta) \approx f_X(x)\, \delta$ for small δ), not a probability; it can exceed 1. Second, the CDF is the universal translator: it is defined for discrete and continuous variables alike, and differentiating (continuous) or differencing (discrete) recovers the local description.

Expectations are integrals now,

$$\mathbf{E}[X] = \int_{-\infty}^{\infty} x\, f_X(x)\, dx, \qquad \mathbf{E}[g(X)] = \int_{-\infty}^{\infty} g(x)\, f_X(x)\, dx,$$

and *every* algebraic theorem of Lessons 17–18 (linearity, LOTUS, variance identities, total expectation, independence factoring) holds verbatim with sums replaced by integrals and PMFs by PDFs — the proofs transpose symbol-for-symbol. Joint PDFs $f_{X,Y}$, marginals ($\int f_{X,Y}\,dy$), and conditionals ($f_{X|Y} = f_{X,Y}/f_Y$) likewise mirror the discrete case.

The continuous catalog:

- *Uniform(a,b)*: $f_X(x) = \frac{1}{b-a}$ on $[a,b]$; $\mathbf{E}[X] = \frac{a+b}{2}$, $\mathrm{var}(X) = \frac{(b-a)^2}{12}$.
- *Exponential(λ)*: $f_X(x) = \lambda e^{-\lambda x}$ for $x \geq 0$; $\mathbf{E}[X] = 1/\lambda$, $\mathrm{var}(X) = 1/\lambda^2$; CDF $F_X(x) = 1 - e^{-\lambda x}$. The continuous geometric: memoryless (Lesson 25 proves it and builds the Poisson process on it).
- *Normal (Gaussian) $\mathcal{N}(\mu, \sigma^2)$*:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-(x-\mu)^2/2\sigma^2}, \qquad \mathbf{E}[X] = \mu, \quad \mathrm{var}(X) = \sigma^2.$$

Proposition 19.2 (Standardization)**.** *If $X \sim \mathcal{N}(\mu, \sigma^2)$ and $a \neq 0$, then $aX + b \sim \mathcal{N}(a\mu + b,\, a^2\sigma^2)$. In particular $Z = (X - \mu)/\sigma \sim \mathcal{N}(0,1)$ (the* standard normal*), so every normal probability reduces to the standard CDF $\Phi(z) = \mathbf{P}(Z \leq z)$:*

$$\mathbf{P}(X \leq x) = \Phi\Big(\frac{x-\mu}{\sigma}\Big).$$

Proof. Let $Y = aX + b$ with $a > 0$ (the case $a < 0$ is symmetric). CDF first, then differentiate — the standard recipe for derived distributions (Lesson 21 makes it a method):

$$F_Y(y) = \mathbf{P}(aX + b \leq y) = \mathbf{P}\Big(X \leq \frac{y-b}{a}\Big) = F_X\Big(\frac{y-b}{a}\Big),$$

so by the chain rule

$$f_Y(y) = \frac{1}{a} f_X\Big(\frac{y-b}{a}\Big) = \frac{1}{\sqrt{2\pi}\,a\sigma} \exp\Big(-\frac{(y-b-a\mu)^2}{2a^2\sigma^2}\Big),$$

which is the $\mathcal{N}(a\mu + b, a^2\sigma^2)$ density. Taking $a = 1/\sigma$, $b = -\mu/\sigma$ gives $Z \sim \mathcal{N}(0,1)$, and the displayed identity is the CDF computation just done. □

Why this family rules: sums of many small independent effects are approximately normal (the central limit theorem, Lesson 24 — this is a theorem, not folklore), and the family is closed under the linear operations of Part I (Lesson 23 does the vector version). Useful numbers: $\mathbf{P}(|X - \mu| \leq \sigma) \approx 0.68$, $\leq 2\sigma \approx 0.95$, $\leq 3\sigma \approx 0.997$.

19.2 Practice: by hand

Example 19.3 (Working a normal probability)**.** Scores are $\mathcal{N}(500, 100^2)$. What fraction exceeds 650?

$$\mathbf{P}(X > 650) = 1 - \Phi\Big(\frac{650-500}{100}\Big) = 1 - \Phi(1.5) \approx 1 - 0.9332 = 0.067.$$

No new integral was done — standardization plus one table/`scipy` lookup.

Example 19.4 (Exponential tail)**.** Requests arrive with exponential($\lambda = 2$ per minute) gaps. Probability a gap exceeds 1 minute: $\mathbf{P}(X > 1) = e^{-2} \approx 0.135$. Given a gap has already lasted 1 minute, the chance it lasts past 2 minutes is $\mathbf{P}(X > 2 \mid X > 1) = e^{-4}/e^{-2} = e^{-2}$ — the same 0.135: memorylessness, proved in general in Lesson 25.

Example 19.5 (Continuous Bayes' rule)**.** A signal Θ is $\mathcal{N}(0, 1)$; you observe $Y = \Theta + N$ where the noise N is $\mathcal{N}(0, 1)$, independent. Then $f_{Y|\Theta}(y \mid \theta)$ is the $\mathcal{N}(\theta, 1)$ density, and Bayes' rule for densities gives

$$f_{\Theta|Y}(\theta \mid y) = \frac{f_\Theta(\theta)\, f_{Y|\Theta}(y \mid \theta)}{\int f_\Theta(t)\, f_{Y|\Theta}(y \mid t)\, dt} \propto e^{-\theta^2/2} e^{-(y-\theta)^2/2} \propto e^{-(\theta - y/2)^2},$$

after completing the square in θ. So $\Theta \mid Y = y \sim \mathcal{N}(y/2, \frac{1}{2})$: the posterior mean $y/2$ splits the difference between the prior mean 0 and the data y, weighted by their precisions. This tiny computation is the seed of Kalman filtering (statistical inference) and of every "Gaussian prior $\times$ Gaussian likelihood" argument.

Machine-learning / statistical-inference connection. Machine learning keeps continuous probability in a supporting role: Gaussian noise in continuous-state models and particle filtering, densities for continuous features. Statistical inference promotes it to the lead: signals are continuous random variables, noise is Gaussian, and Example 19.5 — estimate a Gaussian signal from a noisy observation — is statistical inference's central estimation problem, solved there in full generality (Lessons 22–23 build the machinery).

19.3 Exercises

Exercise 19.1 (Hand)**.** $X \sim$ Uniform$(0, 2)$. Compute $\mathbf{E}[X]$, $\mathrm{var}(X)$, and $\mathbf{E}[X^3]$ (LOTUS). Then find the CDF of $Y = X^2$ and differentiate to get f_Y. (CDF-then-differentiate — the derived-distribution recipe of Lesson 21.)

Exercise 19.2 (Hand)**.** $X \sim \mathcal{N}(10, 4)$. Using Φ: compute $\mathbf{P}(X \le 13)$, $\mathbf{P}(|X - 10| > 4)$, and the value c with $\mathbf{P}(X \le c) = 0.99$ (use $\Phi^{-1}(0.99) \approx 2.33$). Careful: $\sigma = 2$, not 4.

Exercise 19.3 (Proof, $\star$)**.** Prove that the normal density integrates to 1: let $I = \int_{-\infty}^{\infty} e^{-x^2/2} dx$, compute I^2 as a double integral, switch to polar coordinates, and conclude $I = \sqrt{2\pi}$. (One of the great tricks; Fourier analysis meets it again when computing the Fourier transform of a Gaussian.)

Exercise 19.4 (Proof)**.** Show the exponential is the *only* memoryless continuous positive distribution: if $\mathbf{P}(X > s + t) = \mathbf{P}(X > s)\mathbf{P}(X > t)$ for all $s, t \ge 0$, then the function $g(t) = \ln \mathbf{P}(X > t)$ is additive, and (assuming right-continuity) $g(t) = -\lambda t$ for some $\lambda \ge 0$, i.e. $\mathbf{P}(X > t) = e^{-\lambda t}$.

Deeper reading. B&T 3.1–3.3 (PDFs, CDFs, the normal), 3.4–3.6 (joint densities, conditioning, continuous Bayes); the polar-coordinates trick is B&T's Section 3.3 sidebar.

20 Capstone III: A Naive Bayes Classifier from Scratch

NEEDED FOR: Machine learning (core)

Forty-five minutes; Lessons 16–18 at once, on the problem that made Bayes' rule famous in software: spam filtering. Type it, don't copy-paste.

Task. Generate a synthetic labeled dataset from a known naive Bayes model, learn the model's parameters back from the data by counting, classify with Bayes' rule in log space, and measure accuracy against the truth.

```
import numpy as np
rng = np.random.default_rng(0)

# --- 1. The true model: 6 binary features, 2 classes ---
d = 6
prior = 0.3                                  # P(class = spam)
p_spam = np.array([.60, .40, .25, .05, .02, .20])  # P(F_j=1 | spam)
p_ham  = np.array([.05, .08, .10, .30, .25, .15])  # P(F_j=1 | ham)

# --- 2. Sample a dataset (Lesson 16: multiplication rule) ---
n = 5000
y = rng.random(n) < prior                    # class labels
probs = np.where(y[:, None], p_spam, p_ham)  # per-row feature probs
X = rng.random((n, d)) < probs               # conditionally indep. features

# --- 3. "Train": estimate parameters by counting (Lesson 17) ---
prior_hat  = y.mean()
p_spam_hat = X[y].mean(axis=0)               # P(F_j=1 | spam), empirical
p_ham_hat  = X[~y].mean(axis=0)

# --- 4. Classify with Bayes' rule in log space (Lessons 16, 18) ---
def log_lik(X, p):        # sum_j log P(F_j = x_j | class)
    return X @ np.log(p) + (1 - X) @ np.log(1 - p)

log_post_spam = np.log(prior_hat)     + log_lik(X, p_spam_hat)
log_post_ham  = np.log(1 - prior_hat) + log_lik(X, p_ham_hat)
y_hat = log_post_spam > log_post_ham
print("accuracy:", (y_hat == y).mean())      # ~0.9

# --- 5. A posterior probability, not just a decision ---
m = np.maximum(log_post_spam, log_post_ham)  # for numerical stability
post = np.exp(log_post_spam - m) / (
       np.exp(log_post_spam - m) + np.exp(log_post_ham - m))
print(post[:3])                              # P(spam | features), per email
```

Checkpoints (do all of these).

1. `prior_hat`, `p_spam_hat`, `p_ham_hat` land near the true parameters — counting *is* estimation, and the $1/\sqrt{n}$ error is Lesson 24's law of large numbers in action. Re-run with `n = 500` and watch the estimates (and accuracy) degrade.

2. Step 4 is Example 18.6 vectorized: the log of the product $\mathbf{P}(C)\prod_j \mathbf{P}(F_j \mid C)$ becomes a sum, and the sum is a matrix–vector product (Lesson 4). Verify one email by hand against the code's answer.

3. Why logs? Multiply 100 probabilities of size ~ 0.1 and you underflow float64. Confirm: `np.prod(np.full(400, 0.1))` is `0.0`, but the log-sum is finite.

4. The decision rule compares $\log \mathbf{P}(C=1) + \sum_j \log \mathbf{P}(F_j \mid C=1)$ against the $C=0$ version: a *linear* function of the feature vector! Compute the implied weight vector w and bias, and check `y_hat` $= (X@w + b > 0)$ agrees. Naive Bayes is a linear classifier (Part I, Lesson 2's score $w \cdot \phi(x)$) whose weights were set by counting instead of gradient descent.

5. Break the model: regenerate the data with feature 2 *copied* from feature 1 (perfectly dependent), retrain, and watch accuracy drop as the conditional-independence assumption double-counts evidence. This failure mode is why machine learning discusses when naive Bayes is and isn't appropriate.

If every checkpoint makes sense, you are ready for machine learning's probability-flavored units — Bayesian networks, HMMs, and the probabilistic view of learning.

Final self-check list for Part III

You should be able to, from memory: state the three axioms and derive the complement and union-bound rules; define conditional probability and prove total probability and Bayes' rule; work a false-positive-style Bayes problem numerically; define PMF, expectation, variance and compute them for Bernoulli, binomial, geometric, Poisson; state and prove linearity of expectation and use indicators; manipulate joint/marginal/conditional PMFs and prove the total expectation theorem; define PDF and CDF, standardize a normal, and run the CDF-then-differentiate recipe; implement naive Bayes with logs in NumPy. That is B&T Chapters 1–3 at working strength — the probability machine learning assumes on day one.

Part IV: Random Vectors, Limit Theorems, and Stochastic Processes

Statistical inference opens where Part III stops and moves fast: random vectors, convergence and limit theorems, random processes (IID, Markov, Gaussian), stationarity and autocorrelation, MMSE and linear estimation, Kalman filtering, hypothesis testing, PCA. This part is the on-ramp: B&T Chapters 4–8, with the linear-algebra connections (Lessons 6, 11, 13) made explicit, because at this level probability and linear algebra stop being separate subjects. *None of this part is required for machine learning* except as noted for Lesson 26 (Markov chains, which make the MDP and HMM material feel like review).

21 Derived Distributions, Covariance, and Correlation

NEEDED FOR: Statistical inference. Not needed for ML.

21.1 Theory

Theorem 21.1 (Derived distributions: the CDF method and the monotone formula)**.** *Let X be continuous with PDF f_X and let $Y = g(X)$.*

(a) Always*: $F_Y(y) = \mathbf{P}(g(X) \le y)$, computed as an integral of f_X over $\{x : g(x) \le y\}$; differentiate to get f_Y.*

(b) If g is strictly monotone and differentiable with inverse h (so $X = h(Y)$), then

$$f_Y(y) = f_X\big(h(y)\big)\,\big|h'(y)\big|.$$

Proof. (a) is the definition of the CDF plus the definition of a PDF. For (b) with g increasing:

$$F_Y(y) = \mathbf{P}(g(X) \le y) = \mathbf{P}(X \le h(y)) = F_X(h(y)),$$

and the chain rule gives $f_Y(y) = f_X(h(y))\,h'(y)$, with $h' > 0$. For g decreasing the event flips to $\{X \ge h(y)\}$, so $F_Y(y) = 1 - F_X(h(y))$ and $f_Y(y) = -f_X(h(y))h'(y)$ with $h' < 0$; both cases are covered by $|h'(y)|$. □

The factor $|h'(y)|$ is one-dimensional volume scaling — the same role $|\det A|$ played in Lesson 14's change of variables, and the preview of the multivariate formula (Jacobian determinant) statistical inference uses. Proposition 19.2 was this theorem applied to $g(x) = ax + b$.

Definition 21.2 (Covariance, correlation)**.** For random variables X, Y with means μ_X, μ_Y:

$$\operatorname{cov}(X, Y) = \mathbf{E}\big[(X - \mu_X)(Y - \mu_Y)\big] = \mathbf{E}[XY] - \mu_X\mu_Y,$$

(the second form by expanding and linearity), and the *correlation coefficient* is

$$\rho(X, Y) = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y} \qquad (\sigma_X, \sigma_Y > 0).$$

X, Y are *uncorrelated* if $\operatorname{cov}(X, Y) = 0$.

Theorem 21.3 (Covariance is an inner product; $|\rho| \le 1$)**.** *On the space of zero-mean random variables (with finite variance), define $\langle X, Y\rangle = \mathbf{E}[XY]$. This satisfies the three inner-product properties of Lesson 2 (positivity — with $\langle X, X\rangle = 0$ forcing $X = 0$ with probability 1 — symmetry, and linearity in each slot). Consequently, by the Cauchy–Schwarz proof of Theorem 2.5, transcribed verbatim:*

$$\big|\mathbf{E}[XY]\big| \le \sqrt{\mathbf{E}[X^2]}\sqrt{\mathbf{E}[Y^2]},$$

i.e. $|\operatorname{cov}(X,Y)| \le \sigma_X\sigma_Y$, i.e. $-1 \le \rho \le 1$, with $|\rho| = 1$ exactly when Y is (with probability 1) a scalar multiple of X.

Proof. Positivity: $\mathbf{E}[X^2] \ge 0$ since $X^2 \ge 0$; and $\mathbf{E}[X^2] = 0$ forces $\mathbf{P}(|X| > \epsilon) = 0$ for every $\epsilon > 0$ (else the expectation would exceed $\epsilon^2\mathbf{P}(|X| > \epsilon) > 0$), so $X = 0$ with probability 1. Symmetry is $XY = YX$; linearity in each slot is linearity of expectation (Theorem 17.4). With the axioms verified, Lesson 2's proof of Cauchy–Schwarz applies word for word — it used nothing but the axioms. Applying it to the centered variables $X - \mu_X$, $Y - \mu_Y$ gives the covariance statement, and the equality case (one variable a multiple of the other) translates to $Y - \mu_Y = c(X - \mu_X)$: a perfect linear relationship. □

This theorem is the hinge of Part IV: *random variables form an inner product space*, in which uncorrelated = orthogonal, standard deviation = norm, and $\rho = \cos\theta$. Lesson 22 will cash this in: best estimation becomes orthogonal projection.

Proposition 21.4 (Variance of a sum, in general)**.**

$$\operatorname{var}(X + Y) = \operatorname{var}(X) + \operatorname{var}(Y) + 2\operatorname{cov}(X, Y),$$

and independent ⇒ uncorrelated (by Theorem 18.4), but not conversely.

Proof. Expand $\mathbf{E}[(\tilde{X} + \tilde{Y})^2]$ with $\tilde{X}, \tilde{Y}$ centered, as in Theorem 18.4; the cross term is now $2\operatorname{cov}(X, Y)$, kept rather than killed. For the non-converse, one example suffices: let X be uniform on $\{-1, 0, 1\}$ and $Y = X^2$. Then $\mathbf{E}[XY] = \mathbf{E}[X^3] = 0 = \mathbf{E}[X]\mathbf{E}[Y]$: uncorrelated. But $\mathbf{P}(X = 0, Y = 0) = \frac{1}{3} \ne \frac{1}{3} \cdot \frac{1}{3}$: dependent — Y is a deterministic function of X! Covariance detects only *linear* relationships. □

21.2 Practice: by hand

Example 21.5 (Derived distribution, start to finish)**.** $X \sim \text{Uniform}(0, 1)$; find the PDF of $Y = -\ln X$. The inverse of $g(x) = -\ln x$ is $h(y) = e^{-y}$, with $|h'(y)| = e^{-y}$, so for $y > 0$:

$$f_Y(y) = f_X(e^{-y}) \cdot e^{-y} = 1 \cdot e^{-y} = e^{-y}:$$

exponential(1). This is how computers generate exponential random variables from uniform ones (*inverse transform sampling*) — Capstone IV uses it.

Example 21.6 (Covariance from a joint table)**.** For the joint PMF of Example 18.5: $\mathbf{E}[X] = 0.6$, $\mathbf{E}[Y] = 0.3 \cdot 0 + 0.3 \cdot 1 + 0.4 \cdot 2 = 1.1$, and

$$\mathbf{E}[XY] = \sum_{x,y} xy\, p_{X,Y}(x, y) = 1 \cdot 1 \cdot 0.1 + 1 \cdot 2 \cdot 0.3 = 0.7,$$

so $\operatorname{cov}(X, Y) = 0.7 - 0.6 \times 1.1 = 0.04 > 0$: weakly positively associated, matching Lesson 18's observation that seeing $Y = 2$ raised the estimate of X. With $\operatorname{var}(X) = 0.24$ and $\operatorname{var}(Y) = 0.69$: $\rho = 0.04/\sqrt{0.24 \times 0.69} \approx 0.098$.

Statistical-inference connection. Replace "two random variables X, Y" by "one random signal sampled at two times X_t, X_s" and covariance becomes the *autocorrelation function* $R_X(t,s) = \mathbf{E}[X_t X_s]$ — statistical inference's central object, whose Fourier transform (for stationary processes) is the power spectral density. Every property proved here ($|R| \le \sigma^2$ via Cauchy–Schwarz, bilinearity) is quoted there in the first two weeks.

21.3 Exercises

Exercise 21.1 (Hand)**.** $X \sim \text{Uniform}(0,1)$. Find the PDF of (a) $Y = X^2$ (CDF method — careful, g is monotone here, so check both routes agree) and (b) $Y = 3X+1$ (monotone formula).

Exercise 21.2 (Hand)**.** Two fair dice, $S =$ sum, $D =$ difference (first minus second). Compute $\text{cov}(S, D)$. (Slick route: $S = X + Y$, $D = X - Y$; expand by bilinearity of covariance.) Are S and D independent? (Check a corner case, e.g. $S = 12$ vs $D = 5$.)

Exercise 21.3 (Proof, $\star$)**.** Prove the bilinearity of covariance,

$$\text{cov}\Big(\sum_i a_i X_i,\ \sum_j b_j Y_j\Big) = \sum_i \sum_j a_i b_j \,\text{cov}(X_i, Y_j),$$

and deduce the general variance-of-a-sum formula

$$\text{var}\Big(\sum_i X_i\Big) = \sum_i \text{var}(X_i) + 2\sum_{i<j} \text{cov}(X_i, X_j).$$

Exercise 21.4 (Proof)**.** Let $Y = aX + b$ with $a \neq 0$. Show $\rho(X,Y) = \text{sign}(a)$, confirming $|\rho| = 1$ iff the relationship is exactly linear, with the sign recording direction.

Deeper reading. B&T 4.1 (derived distributions), 4.2 (covariance and correlation); B&T 4.4–4.5 (transforms and random sums — worth a skim: statistical inference uses moment generating functions in its concentration-inequality unit).

22 Conditional Expectation and Least Mean Squares

NEEDED FOR: Statistical inference (core: MMSE, linear estimation, Kalman lead-in). Not needed for ML.

22.1 Theory

Lesson 18 defined the *number* $\mathbf{E}[X \mid Y = y]$. The upgrade that powers estimation theory is to see it as a *random variable*: define $\mathbf{E}[X \mid Y]$ to be the random variable that takes the value $\mathbf{E}[X \mid Y = y]$ when $Y = y$ — a function of Y.

Theorem 22.1 (Law of iterated expectations)**.** $\mathbf{E}\big[\mathbf{E}[X \mid Y]\big] = \mathbf{E}[X]$.

Proof. $\mathbf{E}[X \mid Y]$ is a function of Y, say $g(Y)$ with $g(y) = \mathbf{E}[X \mid Y = y]$. By LOTUS and then the total expectation theorem (Theorem 18.3):

$$\mathbf{E}[g(Y)] = \sum_y g(y)\, p_Y(y) = \sum_y \mathbf{E}[X \mid Y = y]\, p_Y(y) = \mathbf{E}[X].$$

(Continuous case: same lines with integrals.) □

Theorem 22.2 (Conditional expectation is the least-mean-squares estimator)**.** *Among* all *functions g, the mean squared error $\mathbf{E}\big[(X-g(Y))^2\big]$ is minimized by $g(Y)=\mathbf{E}[X \mid Y]$.*

Proof. Fix y and work in the conditional world given $Y=y$ (a legitimate probability law, Lesson 16). There, $g(y)$ is just a constant c, and Exercise 17.3 proved $\mathbf{E}[(X-c)^2 \mid Y=y]$ is minimized uniquely by $c=\mathbf{E}[X \mid Y=y]$. So for every y,

$$\mathbf{E}\big[(X-g(Y))^2 \mid Y=y\big] \;\geq\; \mathbf{E}\big[(X-\mathbf{E}[X \mid Y=y])^2 \mid Y=y\big],$$

and averaging both sides over y (total expectation, Theorem 18.3) gives $\mathbf{E}[(X-g(Y))^2] \geq \mathbf{E}[(X-\mathbf{E}[X \mid Y])^2]$. □

Theorem 22.3 (Orthogonality principle)**.** *Let $\tilde{X}=X-\mathbf{E}[X \mid Y]$ be the estimation error. Then for* every *function h,*

$$\mathbf{E}\big[\tilde{X}\,h(Y)\big]=0:$$

the error is uncorrelated with every function of the data. In the inner product of Theorem 21.3, the error is orthogonal *to the space of estimators, so $\mathbf{E}[X \mid Y]$ is the orthogonal projection of X onto the space of functions of Y.*

Proof. Condition on Y and iterate (Theorem 22.1):

$$\mathbf{E}[\tilde{X}h(Y)]=\mathbf{E}\Big[\mathbf{E}\big[\tilde{X}h(Y) \mid Y\big]\Big]=\mathbf{E}\Big[h(Y)\,\mathbf{E}\big[\tilde{X} \mid Y\big]\Big],$$

where $h(Y)$ slides out of the inner conditional expectation because, given Y, it is a constant. And $\mathbf{E}[\tilde{X} \mid Y]=\mathbf{E}[X \mid Y]-\mathbf{E}[X \mid Y]=0$. □

This is Lesson 6, re-lived: Theorem 6.3 said the projection is the closest point and its residual is orthogonal to the subspace; Theorems 22.2 and 22.3 say it again with $\|\cdot\|^2=\mathbf{E}[(\cdot)^2]$. Same geometry, new inner product.

Linear least mean squares. $\mathbf{E}[X \mid Y]$ can be a complicated nonlinear function, and computing it needs the full joint distribution. Statistical inference's workhorse compromise: project onto the smaller subspace of *linear* (affine) estimators $\hat{X}=aY+b$ — computable from means, variances, and one covariance alone.

Theorem 22.4 (Linear LMS)**.** *The minimizers of $\mathbf{E}[(X-aY-b)^2]$ are*

$$a=\frac{\operatorname{cov}(X,Y)}{\operatorname{var}(Y)}, \qquad b=\mu_X-a\mu_Y,$$

so the best linear estimator and its error variance are

$$\hat{X}_L=\mu_X+\frac{\operatorname{cov}(X,Y)}{\operatorname{var}(Y)}\,(Y-\mu_Y), \qquad \mathbf{E}[(X-\hat{X}_L)^2]=(1-\rho^2)\,\sigma_X^2.$$

Proof. For fixed a, the best constant shift is $b=\mathbf{E}[X-aY]=\mu_X-a\mu_Y$ (Exercise 17.3 again), leaving $\mathbf{E}[(\tilde{X}-a\tilde{Y})^2]$ with centered variables. Expand:

$$J(a)=\sigma_X^2-2a\operatorname{cov}(X,Y)+a^2\sigma_Y^2,$$

a parabola in a minimized at $a=\operatorname{cov}(X,Y)/\sigma_Y^2$ (set $J'(a)=0$; $J''=2\sigma_Y^2>0$). Substituting back:

$$J(a^*)=\sigma_X^2-\frac{\operatorname{cov}(X,Y)^2}{\sigma_Y^2}=\sigma_X^2\Big(1-\frac{\operatorname{cov}(X,Y)^2}{\sigma_X^2\sigma_Y^2}\Big)=(1-\rho^2)\sigma_X^2.$$ □

Read the error formula as a scoreboard for ρ: correlation is exactly the fraction of standard deviation that linear prediction can remove (ρ^2 is the "variance explained"). And notice: Theorem 22.4 *is* Lesson 6's least-squares projection onto the two-"vector" span $\{1, Y\}$, with the normal equations $\mathbf{E}[(X - aY - b) \cdot 1] = 0$, $\mathbf{E}[(X - aY - b) \cdot Y] = 0$ — residual orthogonal to each spanning vector.

22.2 Practice: by hand

Example 22.5 (Nonlinear vs. linear estimator)**.** Let $Y \sim \text{Uniform}(-1, 1)$ and $X = Y^2$ (no noise!). The best estimator is $\mathbf{E}[X \mid Y] = Y^2$: zero error. The best *linear* estimator: $\text{cov}(X, Y) = \mathbf{E}[Y^3] - \mathbf{E}[Y^2]\mathbf{E}[Y] = 0$, so $a = 0$ and $\hat{X}_L = \mu_X = \mathbf{E}[Y^2] = \frac{1}{3}$ — a constant, with error variance $\text{var}(Y^2) = \mathbf{E}[Y^4] - \frac{1}{9} = \frac{1}{5} - \frac{1}{9} = \frac{4}{45}$. Linear estimation is blind to this perfectly predictable X because the dependence is purely nonlinear ($\rho = 0$). Moral: LLMS is powerful exactly when correlation captures the relationship — which Lesson 23 shows is always the case for Gaussians.

Example 22.6 (Signal plus noise, the statistical inference primal example)**.** X (signal) has mean 0, variance σ_X^2; N (noise) has mean 0, variance σ_N^2, uncorrelated with X; observe $Y = X + N$. Then $\text{cov}(X, Y) = \sigma_X^2$ and $\text{var}(Y) = \sigma_X^2 + \sigma_N^2$, so

$$\hat{X}_L = \frac{\sigma_X^2}{\sigma_X^2 + \sigma_N^2} Y, \qquad \text{error variance} = \frac{\sigma_X^2 \sigma_N^2}{\sigma_X^2 + \sigma_N^2}.$$

The estimator shrinks the observation by the *signal-to-noise ratio*: trust Y when the signal dominates, shrink toward the prior mean 0 when the noise does. With $\sigma_X^2 = \sigma_N^2 = 1$ this is $\hat{X} = Y/2$ — exactly the posterior mean of Example 19.5, where everything was Gaussian and the linear estimator was fully optimal. Kalman filtering (statistical inference) is this formula applied recursively as observations stream in.

Statistical-inference connection. Statistical inference's estimation unit is these two theorems at scale: MMSE = conditional expectation; LLMS = projection using covariances; the orthogonality principle is the certificate for both, and for the Wiener and Kalman filters built from them. The habit to carry in: *when someone says "best estimate," ask "projection onto what subspace, in which inner product?"*

22.3 Exercises

Exercise 22.1 (Hand)**.** For the joint table of Example 18.5: compute $\mathbf{E}[X \mid Y]$ (a value for each y), verify the law of iterated expectations numerically, and find the LLMS estimator $\hat{X}_L = aY + b$ from the moments computed in Lesson 21. Compare its MSE with that of $\mathbf{E}[X \mid Y]$.

Exercise 22.2 (Hand)**.** In Example 22.6 with $\sigma_X = 2, \sigma_N = 1$: compute $\hat{X}_L$ as a function of Y and the error variance. What fraction of the signal's variance does the observation remove?

Exercise 22.3 (Proof, $\star$)**.** (Law of total variance.) Define $\text{var}(X \mid Y)$ as the variance computed in the conditional world, a function of Y. Prove

$$\text{var}(X) = \mathbf{E}\big[\,\text{var}(X \mid Y)\big] + \text{var}\,\big(\mathbf{E}[X \mid Y]\big),$$

by applying $\text{var} = \mathbf{E}[\square^2] - (\mathbf{E}[\square])^2$ twice and the law of iterated expectations. Interpret: total uncertainty = average leftover uncertainty + uncertainty explained by the data.

Exercise 22.4 (Proof)**.** Derive Theorem 22.4 from the orthogonality principle instead of calculus: require $\mathbf{E}[(X - aY - b) \cdot 1] = 0$ and $\mathbf{E}[(X - aY - b) \cdot Y] = 0$ and solve the two linear equations. (These are normal equations, Lesson 6 — with $\mathbf{E}[\cdot\cdot]$ as the inner product.)

Deeper reading. B&T 4.3 (conditional expectation and variance revisited — iterated expectations, total variance); B&T 8.3–8.4 (Bayesian least mean squares and *linear* LMS estimation — the two theorems above in their inference setting); then statistical inference's linear-estimation notes will read as review.

23 Random Vectors and Gaussian Vectors

NEEDED FOR: Statistical inference (core: PCA, Gaussian processes, Kalman). Uses Lessons 11 & 13. Not needed for ML.

23.1 Theory

Definition 23.1 (Random vector, mean, covariance matrix)**.** A *random vector* $X = (X_1, \dots, X_n)$ is a vector of random variables on one experiment. Its *mean vector* is $\mu = \mathbf{E}[X] = (\mathbf{E}[X_1], \dots, \mathbf{E}[X_n])$, and its *covariance matrix* is the $n \times n$ matrix

$$\Sigma = \mathbf{E}\big[(X - \mu)(X - \mu)^{\mathsf{T}}\big], \qquad \Sigma_{ij} = \operatorname{cov}(X_i, X_j),$$

with variances on the diagonal.

Proposition 23.2 (Covariance matrices transform like quadratic forms)**.** *For any (deterministic) $m \times n$ matrix A and $b \in \mathbb{R}^m$, the random vector $AX + b$ has mean $A\mu + b$ and covariance matrix $A\Sigma A^{\mathsf{T}}$. In particular, for a single linear functional $w \in \mathbb{R}^n$:*

$$\operatorname{var}(w^{\mathsf{T}} X) = w^{\mathsf{T}} \Sigma\, w.$$

Proof. The mean is linearity of expectation, coordinate by coordinate. For the covariance, let $\tilde{X} = X - \mu$; then $AX + b$ centered is $A\tilde{X}$, and

$$\mathbf{E}\big[(A\tilde{X})(A\tilde{X})^{\mathsf{T}}\big] = \mathbf{E}\big[A\tilde{X}\tilde{X}^{\mathsf{T}}A^{\mathsf{T}}\big] = A\,\mathbf{E}[\tilde{X}\tilde{X}^{\mathsf{T}}]\,A^{\mathsf{T}} = A\Sigma A^{\mathsf{T}},$$

pulling the constant matrices through the (entrywise) expectation. The scalar case is w^{T} as a $1 \times n$ matrix. □

Theorem 23.3 (Covariance matrices are exactly the PSD matrices)**.** *Every covariance matrix is symmetric positive semidefinite. Conversely, every symmetric PSD matrix is the covariance matrix of some random vector.*

Proof. Symmetry: $\operatorname{cov}(X_i, X_j) = \operatorname{cov}(X_j, X_i)$. PSD: for any $w \in \mathbb{R}^n$, $w^{\mathsf{T}}\Sigma w = \operatorname{var}(w^{\mathsf{T}}X) \geq 0$ by Proposition 23.2. Conversely, given symmetric PSD Σ, the spectral theorem (Theorem 11.4) gives $\Sigma = Q\Lambda Q^{\mathsf{T}}$ with $\Lambda = \operatorname{diag}(\lambda_i)$, $\lambda_i \geq 0$; set $\Sigma^{1/2} = Q\Lambda^{1/2}Q^{\mathsf{T}}$ (a symmetric matrix with $\Sigma^{1/2}\Sigma^{1/2} = \Sigma$). Take Z with independent, zero-mean, unit-variance coordinates — so $\operatorname{cov}(Z) = I$ — and set $X = \Sigma^{1/2}Z$: by Proposition 23.2, $\operatorname{cov}(X) = \Sigma^{1/2} I\, \Sigma^{1/2} = \Sigma$. □

Lesson 11's abstract PSD theory just became concrete: *every* covariance matrix has an orthonormal eigenbasis with nonnegative eigenvalues, the eigenvectors are the *principal components* (directions of uncorrelated variation), and the eigenvalues are the variances along them. Changing to the eigenbasis ($Y = Q^{\mathsf{T}}\tilde{X}$, which has covariance $Q^{\mathsf{T}}\Sigma Q = \Lambda$, diagonal) *decorrelates* the vector — that transformation is PCA, and the top-k eigenvalue truncation is Lesson 13's Eckart–Young story applied to data.

Definition 23.4 (Gaussian random vector)**.** X is a *Gaussian (normal) random vector*, $X \sim \mathcal{N}(\mu, \Sigma)$, if it can be written $X = \mu + BZ$ where Z has independent $\mathcal{N}(0,1)$ coordinates and B is any matrix with $BB^{\mathsf{T}} = \Sigma$. Equivalently (the equivalence is proved in statistical inference via characteristic functions; we adopt the constructive definition): every linear combination $w^{\mathsf{T}}X$ is a (one-dimensional) normal random variable. For invertible Σ, the density is

$$f_X(x) = \frac{1}{(2\pi)^{n/2}(\det\Sigma)^{1/2}} \exp\Big(-\tfrac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\Big).$$

The $\sqrt{\det\Sigma}$ is Lesson 14's volume factor (the map B stretches the unit Gaussian ball into the ellipsoid of the distribution), and the level sets of f_X are the ellipsoids $x^{\mathsf{T}}\Sigma^{-1}x = \text{const}$ whose axes are — again — the eigenvectors of Σ (Lesson 11's quadratic-form picture).

Theorem 23.5 (The Gaussian miracles)**.** *Let $X \sim \mathcal{N}(\mu, \Sigma)$.*

(a) Linearity: $AX + b \sim \mathcal{N}(A\mu + b,\, A\Sigma A^{\mathsf{T}})$ — *Gaussians stay Gaussian under every linear map.*

(b) Uncorrelated $\Rightarrow$ independent: *if two coordinates (or blocks) of a jointly Gaussian vector are uncorrelated, they are independent.*

Proof. (a) Write $X = \mu + BZ$ with $BB^{\mathsf{T}} = \Sigma$; then $AX + b = (A\mu + b) + (AB)Z$ with $(AB)(AB)^{\mathsf{T}} = A\Sigma A^{\mathsf{T}}$ — literally the definition, plus Proposition 23.2 for the parameters.

(b) In the density case: if Σ is block-diagonal $\left(\begin{smallmatrix}\Sigma_1 & 0\\ 0 & \Sigma_2\end{smallmatrix}\right)$, then Σ^{-1} is block-diagonal too, $\det\Sigma = \det\Sigma_1 \det\Sigma_2$ (Lesson 14 on the block-triangular form), and the exponent splits as a sum of a term in x_1 and a term in x_2 — so f_X *factors* into the two marginal Gaussian densities, which is independence. □

Part (b) is a miracle worth staring at: Proposition 21.4 warned that uncorrelated is far weaker than independent (recall X and X^2) — *except* in the jointly Gaussian world, where covariance captures all dependence. This is also why Lesson 22's *linear* LMS estimator is fully optimal for Gaussians (Example 22.6): no nonlinear function can see structure that covariances don't already report.

23.2 Practice: by hand

Example 23.6 (A 2-D Gaussian, fully worked)**.** Let $\Sigma = \begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}$, $\mu = 0$ — Example 10.7's matrix as a covariance. Eigen-decomposition (known): eigenvalues $3, 1$ with orthonormal eigenvectors $\frac{1}{\sqrt{2}}(1,1)$, $\frac{1}{\sqrt{2}}(1,-1)$. Read off:

- The density's ellipses have long axis along $(1,1)$ (variance 3) and short axis along $(1,-1)$ (variance 1): the cloud is a tilted ellipse.

- PCA: the first principal component is $(1,1)/\sqrt{2}$, "explaining" $3/(3+1) = 75\%$ of the total variance $\operatorname{tr}\Sigma = 4$ (trace = sum of eigenvalues, Lesson 14).
- Whitening: $Y = \Lambda^{-1/2}Q^{\mathsf{T}}X$ has covariance I (check with Proposition 23.2) — rotate to the eigenbasis, divide each axis by its standard deviation.
- $\operatorname{var}(X_1 - X_2) = w^{\mathsf{T}}\Sigma w$ with $w = (1,-1)$: $2+2-2\cdot 1 = 2$ — the quadratic form doing what Lesson 11 promised.

Statistical-inference connection. A *Gaussian process* is this lesson with infinitely many coordinates: any finite set of samples $(X_{t_1}, \ldots, X_{t_n})$ is a Gaussian vector, so the whole process is specified by a mean function and a covariance (autocorrelation) function. Kalman filtering is Theorem 23.5(a) plus Lesson 22's estimator, iterated. PCA of data is Theorem 23.3's eigen-story with Σ estimated from samples. When statistical inference says "jointly Gaussian, hence independent given uncorrelated," that is miracle (b).

23.3 Exercises

Exercise 23.1 (Hand)**.** Let X_1, X_2 be uncorrelated with variances $1, 4$, and $Y_1 = X_1 + X_2$, $Y_2 = X_1 - X_2$. Write the matrix A with $Y = AX$, compute $\operatorname{cov}(Y) = A\Sigma A^{\mathsf{T}}$, and read off $\operatorname{var}(Y_1)$, $\operatorname{var}(Y_2)$, $\operatorname{cov}(Y_1, Y_2)$. When would Y_1, Y_2 be uncorrelated?

Exercise 23.2 (Hand)**.** For $\Sigma = \left(\begin{smallmatrix} 5 & 2 \\ 2 & 2 \end{smallmatrix}\right)$: find eigenvalues and orthonormal eigenvectors, the fraction of variance along the first principal component, and the whitening transform. Sketch (in words) the density's ellipses.

Exercise 23.3 (Proof, $\star$)**.** Prove that $\Sigma^{1/2} = Q\Lambda^{1/2}Q^{\mathsf{T}}$ is the *unique* symmetric PSD square root of a symmetric PSD Σ. (Uniqueness: if $S^2 = \Sigma$ with S symmetric PSD, show S and Σ have the same eigenvectors — diagonalize S and square. LADR 7C proves this as the "positive operators have unique positive square roots" theorem.)

Exercise 23.4 (Proof)**.** Using Theorem 23.5(a), show that if $X \sim \mathcal{N}(\mu, \Sigma)$ with Σ invertible, the whitened vector $\Lambda^{-1/2}Q^{\mathsf{T}}(X - \mu)$ is $\mathcal{N}(0, I)$ — i.e. its coordinates are independent standard normals. (This is why "every Gaussian is a linearly transformed standard one" can be run in both directions.)

Deeper reading. B&T 4.2 (correlation matrices, in the covariance section); the bivariate normal supplement on the B&T website; LADR 7C (positive operators, square roots) for the linear-algebra half; statistical inference's random-vectors notes then formalize the characteristic-function view.

24 Limit Theorems

NEEDED FOR: Statistical inference (core: convergence, concentration); machine learning (background: why Monte Carlo works)

24.1 Theory

Throughout, $X_1, X_2, \ldots$ are independent identically distributed (i.i.d.) with mean μ and variance $\sigma^2 < \infty$, and

$$M_n = \frac{X_1 + \cdots + X_n}{n}$$

is the *sample mean*. By linearity and independence: $\mathbf{E}[M_n] = \mu$ and $\mathrm{var}(M_n) = \sigma^2/n$ — averaging keeps the target and shrinks the spread. The limit theorems sharpen "shrinks" into guarantees.

Theorem 24.1 (Markov and Chebyshev inequalities)**.** *(a) If $X \geq 0$, then for any $a > 0$:* $\mathbf{P}(X \geq a) \leq \dfrac{\mathbf{E}[X]}{a}$.

(b) For any random variable Y with mean μ and variance σ^2, and any $\epsilon > 0$:

$$\mathbf{P}\big(|Y - \mu| \geq \epsilon\big) \leq \frac{\sigma^2}{\epsilon^2}.$$

Proof. (a) Let I be the indicator of $\{X \geq a\}$. Then $aI \leq X$ always: if the event occurs, $aI = a \leq X$; if not, $aI = 0 \leq X$. Taking expectations (monotonicity of $\mathbf{E}$, which follows from nonnegativity of $\mathbf{E}$ applied to $X - aI$): $a\mathbf{P}(X \geq a) = \mathbf{E}[aI] \leq \mathbf{E}[X]$.

(b) Apply (a) to the nonnegative variable $(Y - \mu)^2$ with threshold ϵ^2:

$$\mathbf{P}(|Y - \mu| \geq \epsilon) = \mathbf{P}\big((Y - \mu)^2 \geq \epsilon^2\big) \leq \frac{\mathbf{E}[(Y - \mu)^2]}{\epsilon^2} = \frac{\sigma^2}{\epsilon^2}. \qquad \square$$

Theorem 24.2 (Weak law of large numbers)**.** *For every $\epsilon > 0$,*

$$\mathbf{P}\big(|M_n - \mu| \geq \epsilon\big) \leq \frac{\sigma^2}{n\epsilon^2} \xrightarrow[n\to\infty]{} 0.$$

We say M_n converges in probability *to μ.*

Proof. Chebyshev applied to M_n, whose mean is μ and variance is σ^2/n. $\square$

Three remarks. (1) The bound is quantitative: to pin μ within ϵ with 95% confidence, $n = 20\sigma^2/\epsilon^2$ samples suffice — crude but honest, and the pattern (accuracy costs $1/\epsilon^2$) is right. (2) Chebyshev is distribution-free and correspondingly loose; statistical inference's concentration unit (Chernoff/Hoeffding bounds, built from transforms, B&T 5.4's sidebar and Chapter 4.4) trades generality for exponentially better bounds. (3) The *strong* law (B&T 5.5) upgrades the conclusion to $\mathbf{P}(M_n \to \mu) = 1$; the distinction between the two convergence modes is a statistical-inference lecture of its own.

Theorem 24.3 (Central limit theorem)**.** *Let $Z_n = \dfrac{X_1 + \cdots + X_n - n\mu}{\sigma\sqrt{n}}$ (the sum, standardized to mean 0 and variance 1). Then for every z,*

$$\mathbf{P}(Z_n \leq z) \xrightarrow[n\to\infty]{} \Phi(z),$$

the standard normal CDF — regardless of the distribution of the X_i.

The proof (via transforms: the moment generating function of Z_n converges to the Gaussian's, B&T 5.4 sketches it) is the one omission in this part; the statement and its use are non-negotiable. The practical reading: for large n,

$$X_1 + \cdots + X_n \;\approx\; \mathcal{N}(n\mu,\, n\sigma^2), \qquad M_n \;\approx\; \mathcal{N}(\mu,\, \sigma^2/n),$$

which converts "how unlikely is this deviation?" into a Φ lookup (Lesson 19). The CLT is why the Gaussian owns Lesson 23 and statistical inference: sums of many small independent contributions — thermal noise being the canonical example — are Gaussian by theorem.

24.2 Practice: by hand

Example 24.4 (Polling)**.** Poll n voters; each answer is Bernoulli(p), and M_n estimates p. Since $\sigma^2 = p(1-p) \le \frac{1}{4}$: Chebyshev guarantees $\mathbf{P}(|M_n - p| \ge 0.03) \le \frac{1/4}{n(0.03)^2}$, which is $\le 5\%$ once $n \ge 5556$. The CLT sharpens this: $\mathbf{P}(|M_n - p| \ge 0.03) \approx 2\big(1 - \Phi(0.03\sqrt{n}/\sqrt{p(1-p)}\,)\big) \le 5\%$ once $0.03\sqrt{4n} \ge 1.96$, i.e. $n \ge 1068$ — the famous "$\pm 3\%$ needs about a thousand people," independent of the population size.

Example 24.5 (CLT for a die)**.** Sum 100 fair-die rolls. $\mu = 3.5$, $\sigma^2 = \frac{35}{12} \approx 2.92$ per roll, so the sum is approximately $\mathcal{N}(350, 292)$ and

$$\mathbf{P}(\text{sum} > 380) \approx 1 - \Phi\Big(\tfrac{380-350}{\sqrt{292}}\Big) = 1 - \Phi(1.76) \approx 0.04.$$

An exact computation would need a 100-fold convolution; the CLT needs two moments.

Statistical-inference / machine-learning connection. Statistical inference devotes a unit to exactly this ladder — Markov → Chebyshev → Chernoff, then modes of convergence, then the CLT — and uses it to reason about detection error rates and generalization in learning. For machine learning the payoff is quieter but constant: every Monte Carlo estimate (particle filters, MCTS rollouts, sampled gradients in SGD) is a sample mean whose $\sigma/\sqrt{n}$ error bar and approximate normality come from these theorems.

24.3 Exercises

Exercise 24.1 (Hand)**.** A server handles requests with mean processing time 2 ms and standard deviation 1 ms (distribution unknown). Bound $\mathbf{P}$(total time for 400 requests $>$ 900 ms) two ways: Chebyshev, then CLT approximation. Compare.

Exercise 24.2 (Hand)**.** How many fair-coin flips guarantee, via Chebyshev, that the observed fraction of heads is within 0.01 of $\frac{1}{2}$ with probability at least 0.99? What does the CLT suggest instead?

Exercise 24.3 (Proof, $\star$)**.** Prove the "one-sided Chebyshev" warm-up: for any Y with mean 0 and variance σ^2 and any $a > 0$, $\mathbf{P}(Y \ge a) \le \frac{\sigma^2}{\sigma^2 + a^2}$. (Hint: for any $t \ge 0$, $\{Y \ge a\} \subseteq \{(Y+t)^2 \ge (a+t)^2\}$; apply Markov and minimize over t.)

Exercise 24.4 (Proof)**.** Show that convergence in probability to μ does *not* require $\mathrm{var}(M_n) \to 0$ in general: construct Y_n with $Y_n \to 0$ in probability but $\mathrm{var}(Y_n) \to \infty$. (Take $Y_n = n$ with probability $1/n^1$... adjust the exponent until it works, and say why.)

Deeper reading. B&T 5.1–5.4 (inequalities, WLLN, convergence in probability, CLT), 5.5 (strong law); B&T 4.4 (transforms) is the doorway to the Chernoff bounds statistical inference adds on top.

25 The Bernoulli and Poisson Processes

NEEDED FOR: Statistical inference (first random processes; independent increments). Not needed for ML.

25.1 Theory

A *random process* is a collection of random variables indexed by time. The two canonical arrival processes — coin flips in discrete time, and its continuous-time limit — are where statistical inference starts building intuition for the general theory.

Definition 25.1 (Bernoulli process)**.** A *Bernoulli process* with rate p is a sequence $X_1, X_2, \ldots$ of i.i.d. Bernoulli(p) random variables: independent trials, each a success ("arrival") with probability p.

Everything about it follows from Part III: the number of arrivals in n slots is Binomial(n, p); the time to the first arrival is Geometric(p); and the process is *memoryless* — whatever has happened through time n, the future $X_{n+1}, X_{n+2}, \ldots$ is a fresh Bernoulli process, by independence. Consequently the *interarrival times* (gaps between successive arrivals) are i.i.d. Geometric(p): after each arrival the process restarts.

Definition 25.2 (Poisson process)**.** A *Poisson process* with rate $\lambda > 0$ counts arrivals in continuous time: with $N(t)$ = number of arrivals in $(0, t]$, the process satisfies

(i) (*Poisson marginals*) $N(t) \sim \text{Poisson}(\lambda t)$: $\mathbf{P}(N(t) = k) = e^{-\lambda t}(\lambda t)^k/k!$;

(ii) (*independent, stationary increments*) arrival counts in disjoint intervals are independent, and the count in an interval depends only on its length.

Theorem 25.3 (The Poisson process is the limit of Bernoulli processes)**.** *Chop $(0, t]$ into n slots of width t/n and run a Bernoulli process with per-slot probability $p_n = \lambda t/n$. Then the number of successes $S_n \sim$ Binomial(n, p_n) satisfies, for each fixed k,*

$$\mathbf{P}(S_n = k) \xrightarrow[n\to\infty]{} e^{-\lambda t}\frac{(\lambda t)^k}{k!}.$$

Proof. Write $a = \lambda t$. Then

$$\mathbf{P}(S_n = k) = \binom{n}{k}\left(\frac{a}{n}\right)^k\left(1 - \frac{a}{n}\right)^{n-k} = \frac{a^k}{k!} \cdot \underbrace{\frac{n(n-1)\cdots(n-k+1)}{n^k}}_{\to 1} \cdot \underbrace{\left(1 - \frac{a}{n}\right)^n}_{\to e^{-a}} \cdot \underbrace{\left(1 - \frac{a}{n}\right)^{-k}}_{\to 1},$$

using $(1 - a/n)^n \to e^{-a}$ and the fact that the first fraction is a product of k factors each tending to 1. □

Theorem 25.4 (Interarrival times are exponential)**.** *In a Poisson process of rate λ, the time T_1 to the first arrival is exponential(λ), and successive interarrival times are i.i.d. exponential(λ).*

Proof. $\mathbf{P}(T_1 > t) = \mathbf{P}(N(t) = 0) = e^{-\lambda t}$: exactly the exponential tail, so $T_1 \sim$ exponential(λ). For the later gaps: by independent and stationary increments, the process after any fixed time s is a fresh Poisson process independent of the past (this is the memorylessness of Lesson 19's exponential, now at the process level; the full argument that this also holds after the *random* time of an arrival is B&T 6.2), so each gap has the same exponential distribution, independently. □

Two standard consequences, both provable in a few lines from the definitions (Exercises): *merging* two independent Poisson processes of rates λ_1, λ_2 yields a Poisson process of rate $\lambda_1+\lambda_2$, and *splitting* a rate-λ process by an independent coin (p / $1-p$) yields two *independent* Poisson processes of rates $p\lambda$, $(1-p)\lambda$.

25.2 Practice: by hand

Example 25.5 (A packet link)**.** Packets arrive as a Poisson process at $\lambda = 3$ per second.

$$\mathbf{P}(\text{exactly 2 in one second}) = e^{-3}\frac{3^2}{2!} \approx 0.224, \qquad \mathbf{P}(\text{none in 2 seconds}) = e^{-6} \approx 0.0025.$$

Expected arrivals in a minute: $\lambda t = 180$; and by Lesson 24's CLT (a Poisson is a sum of many thin-slice Bernoullis), the count over a minute is approximately $\mathcal{N}(180, 180)$ — Poisson variance equals its mean.

Example 25.6 (Merging and the memoryless race)**.** Emails arrive at rate 2/hour, texts at rate 4/hour, independent Poisson. Interruptions of either kind: Poisson at rate 6/hour, so the wait for the next interruption is exponential(6) — mean 10 minutes. Given an interruption occurs, it is a text with probability 4/6: in a merged process, each arrival came from stream i with probability $\lambda_i / \sum_j \lambda_j$, independent of the past.

Statistical-inference connection. These are your first *random processes*, and they carry the vocabulary statistical inference runs on: independent increments, stationarity (increment distributions depend only on interval length — the first example of a stationarity property), memorylessness, and the interplay between the counting view $N(t)$ and the interarrival view $T_1, T_2, \ldots$. Shot noise in electronics is modeled directly as a Poisson process; the Wiener process (Brownian motion) that statistical inference introduces later is the same independent-stationary-increments idea with Gaussian increments in place of Poisson.

25.3 Exercises

Exercise 25.1 (Hand)**.** A Bernoulli process has $p = 0.1$ per slot. Compute: the probability of no arrival in 10 slots; the expected time to the third arrival (sum of three geometrics); the probability the first arrival is in slot 5 given none in the first 3 (memorylessness makes this instant).

Exercise 25.2 (Hand)**.** Buses (rate 4/hour) and taxis (rate 8/hour) pass a corner as independent Poisson processes. You wait for either. Find: the mean wait; the probability the first three vehicles are all taxis; the probability exactly 2 buses pass in 30 minutes.

Exercise 25.3 (Proof, $\star$)**.** Prove the merging theorem for marginals: if $N_1(t), N_2(t)$ are independent Poisson($\lambda_1 t$), Poisson($\lambda_2 t$), then $N_1(t) + N_2(t)$ is Poisson($(\lambda_1 + \lambda_2)t$). (Convolve the PMFs and recognize the binomial theorem inside the sum.)

Exercise 25.4 (Proof)**.** For a Poisson process, show $\operatorname{cov}\big(N(s), N(t)\big) = \lambda \min(s,t)$ for $s \le t$ by writing $N(t) = N(s) + \big(N(t) - N(s)\big)$ and using independent increments. (This "min" covariance reappears verbatim for the Wiener process in statistical inference.)

Deeper reading. B&T 6.1 (Bernoulli process), 6.2 (Poisson process — including the random-incidence paradox, a classic statistical-inference exam topic worth reading now).

26 Markov Chains

NEEDED FOR: Statistical inference (core: Markov processes); machine learning (helpful: the substrate of MDPs and HMMs). Uses Lesson 10.

26.1 Theory

Definition 26.1 (Discrete-time Markov chain)**.** A *Markov chain* on states $\{1, \dots, m\}$ is a sequence $X_0, X_1, \dots$ such that the next state depends on the past only through the present (the *Markov property*):

$$\mathbf{P}(X_{n+1} = j \mid X_n = i,\, X_{n-1}, \dots, X_0) = \mathbf{P}(X_{n+1} = j \mid X_n = i) = P_{ij},$$

where the *transition probabilities* P_{ij} form the $m \times m$ *transition matrix* P: each row is a PMF, $P_{ij} \ge 0$, $\sum_j P_{ij} = 1$.

Theorem 26.2 (Chapman–Kolmogorov: n-step transitions are matrix powers)**.** *Let $r_{ij}(n) = \mathbf{P}(X_n = j \mid X_0 = i)$. Then $r_{ij}(n) = (P^n)_{ij}$; and if the row vector $\pi^{(0)}$ is the PMF of X_0, the PMF of X_n is $\pi^{(0)} P^n$.*

Proof. Induct on n; the case $n = 1$ is the definition. For the step, condition on the state at time n (total probability, Theorem 16.4(b)) and use the Markov property:

$$r_{ij}(n+1) = \sum_{k=1}^{m} \mathbf{P}(X_n = k \mid X_0 = i)\, \mathbf{P}(X_{n+1} = j \mid X_n = k) = \sum_k r_{ik}(n) P_{kj},$$

which is exactly the (i,j) entry of $P^n P = P^{n+1}$. The distribution statement is the same computation with the initial PMF weighting row i. □

Probability question → Lesson 4 matrix multiplication; long-run behavior → Lesson 10 eigenvalues. That is the whole design.

Definition 26.3 (Stationary distribution)**.** A row vector π with $\pi_j \ge 0$, $\sum_j \pi_j = 1$ is a *stationary distribution* if

$$\pi P = \pi \qquad \text{(the } \textit{balance equations } \pi_j = \textstyle\sum_k \pi_k P_{kj}\text{, plus normalization).}$$

Started from π, the chain keeps distribution π forever (Theorem 26.2).

$\pi P = \pi$ says: π^{T} is an eigenvector of P^{T} with eigenvalue 1 — Exercise 10.5's "population matrix" was this in disguise. Eigenvalue 1 always exists (P has row sums 1, so $P\mathbf{1} = \mathbf{1}$; P and P^{T} have the same eigenvalues by Lesson 14, $\det(P - \lambda I) = \det(P^{\mathsf{T}} - \lambda I)$). The substantive question is convergence:

Theorem 26.4 (Steady-state convergence — statement)**.** *Suppose the chain is* irreducible *(every state can reach every state) and* aperiodic *(no forced cycling; e.g. some state has $P_{ii} > 0$). Then there is a unique stationary π, with all $\pi_j > 0$, and for* every *starting state i:*

$$r_{ij}(n) \xrightarrow[n\to\infty]{} \pi_j.$$

Moreover π_j is the long-run fraction of time spent in state j.

B&T 7.3 proves this; the eigen-story explains it (for the diagonalizable case): the other eigenvalues of P have $|\lambda| < 1$ under these hypotheses, so in the eigen-expansion of $\pi^{(0)}P^n$ every component except the $\lambda = 1$ one dies geometrically — Lesson 10's "$|\lambda| < 1$ decays" as a theorem about forgetting the initial condition. The convergence *rate* is set by the second-largest $|\lambda|$ (the "spectral gap"), the number statistical inference and MCMC practitioners both watch.

26.2 Practice: by hand

Example 26.5 (A two-state weather chain, end to end)**.** Sunny (1) or rainy (2): $P = \begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix}$.

Balance: $\pi_1 = 0.9\pi_1 + 0.5\pi_2$ gives $0.1\pi_1 = 0.5\pi_2$, so $\pi = (\frac{5}{6}, \frac{1}{6})$: sunny five days in six, whatever today is. Two-step check from rain: $r_{21}(2) = (P^2)_{21} = 0.5\cdot 0.9 + 0.5\cdot 0.5 = 0.70$, already climbing toward $\frac{5}{6} \approx 0.83$. Eigenvalues: $\operatorname{tr} P = 1.4$, $\det P = 0.4$, so $\lambda = 1,\ 0.4$ (Lesson 14's trace/det shortcut) — errors shrink by a factor 0.4 per step, and after n steps the deviation from steady state is proportional to 0.4^n.

Example 26.6 (Absorption — a gambler)**.** States $0, 1, 2, 3$: you hold \$$k$, bet \$1 at fair odds each round, stop at \$0 (ruin) or \$3 (goal). States 0 and 3 are *absorbing* ($P_{00} = P_{33} = 1$). Let a_k = probability of reaching 3 from k. Conditioning on one step (total probability):

$$a_k = \tfrac{1}{2}a_{k-1} + \tfrac{1}{2}a_{k+1}, \qquad a_0 = 0,\ a_3 = 1,$$

so a is linear in k: $a_k = k/3$. Fair game, fair chances — proportional to your stake. (B&T 7.4 develops absorption probabilities and expected absorption times in general; the same first-step conditioning gives both.)

Machine-learning / statistical-inference connection. A Markov chain is the "environment half" of machine learning's MDPs: add actions that choose which row of transition probabilities applies, plus rewards, and value iteration is first-step conditioning (the gambler calculation) iterated to a fixed point. An HMM is a Markov chain observed through noise, and its filtering updates ride on Theorem 26.2. In statistical inference, Markov chains are the first structured random process — and the eigen-analysis of P (stationarity, spectral gap, reversibility) is a recurring exam theme. This is the one Part IV lesson worth doing *before* the machine-learning work if time permits.

26.3 Exercises

Exercise 26.1 (Hand)**.** A machine is up (1) or down (2): $P_{12} = 0.05$, $P_{21} = 0.6$. Find the stationary distribution, the long-run fraction of downtime, and both eigenvalues of P. Roughly how many steps until the initial condition is forgotten (deviation shrunk by $\sim 0.4^n$-style factor — compute the actual base)?

Exercise 26.2 (Hand)**.** For the gambler with unfair odds (win probability 0.4): re-derive the first-step equations and solve for a_1, a_2 with $a_0 = 0, a_3 = 1$. (The general solution is a geometric progression in $(0.6/0.4)^k$; or just solve the 2×2 linear system — Lesson 5.)

Exercise 26.3 (Proof, $\star$)**.** Prove that if π satisfies the *detailed balance* equations $\pi_i P_{ij} = \pi_j P_{ji}$ for all i, j, then π is stationary. (Sum over i.) Verify detailed balance holds for Example 26.5 and exhibit a 3-state chain where stationarity holds but detailed balance fails (hint: probability flowing around a cycle $1 \to 2 \to 3 \to 1$).

Exercise 26.4 (Proof)**.** Show that if P and Q are transition matrices, so is PQ; conclude P^n is a transition matrix for every n (rows of $r_{ij}(n)$ are PMFs, as they must be).

Deeper reading. B&T 7.1–7.3 (Markov chains, classification, steady state), 7.4 (absorption — the MDP-adjacent part), 7.5 (continuous time, inference-relevant); MDP treatments in machine learning re-derive first-step analysis as the Bellman equation.

27 Capstone IV: Estimation and Simulation in NumPy

NEEDED FOR: Statistical inference

Forty-five minutes; Part IV at once, in statistical inference's own shape: a hidden Gaussian signal, a noisy sensor, and the estimation machinery, checked against simulation.

Task. Track a slowly drifting Gaussian signal through noisy observations: build the model, derive the LLMS estimator from covariances, verify the orthogonality principle and the error formula, and watch the CLT and a Markov chain appear in the same program.

```
import numpy as np
rng = np.random.default_rng(0)

# --- 1. A Gauss-Markov signal (Lessons 23, 26) ---
#     S_t = a S_{t-1} + W_t : a Markov chain with Gaussian steps
n, a, sw = 200_000, 0.95, 0.3
S = np.zeros(n)
W = sw * rng.normal(size=n)
for t in range(1, n):
    S[t] = a * S[t-1] + W[t]
# stationary variance: var = a^2 var + sw^2  =>  var = sw^2/(1-a^2)
print(S.var(), sw**2 / (1 - a**2))            # ~0.92 both

# --- 2. Noisy observations (Lesson 22's signal-plus-noise) ---
sn = 1.0
Y = S + sn * rng.normal(size=n)

# --- 3. LLMS from first principles (Lesson 22) ---
c_sy = np.cov(S, Y)[0, 1]                     # ~ var(S)
Shat = (c_sy / Y.var()) * Y
mse  = ((S - Shat)**2).mean()
predicted = S.var() * sn**2 / (S.var() + sn**2)
print(mse, predicted)                         # agree (Example 22.3)
```

```
# --- 4. Orthogonality principle (Lesson 22) ---
err = S - Shat
print(np.corrcoef(err, Y)[0, 1])              # ~0

# --- 5. Autocorrelation of the signal (Lesson 21 -> statistical inference) ---
for k in (1, 5, 20):
    print(k, np.corrcoef(S[:-k], S[k:])[0, 1], a**k)   # rho(k) = a^k

# --- 6. CLT check on the time average (Lesson 24) ---
means = S.reshape(200, 1000).mean(axis=1)     # 200 window averages
print(means.mean(), means.std())              # ~0, small
```

Checkpoints (do all of these).

1. Step 1's variance identity is Proposition 21.4 at steady state: $\mathrm{var}(S_t) = a^2 \,\mathrm{var}(S_{t-1}) + \sigma_W^2$, solved at the fixed point. The process is a Markov chain (Lesson 26) with a continuous state space — statistical inference calls it a *Gauss–Markov* or AR(1) process.

2. Step 3's measured MSE matches Example 22.6's formula $\sigma_S^2\sigma_N^2/(\sigma_S^2 + \sigma_N^2)$. Change `sn` to 0.3 and to 3.0 and confirm the estimator trusts/distrusts Y accordingly.

3. Step 4 is Theorem 22.3 empirically; also check `np.corrcoef(err, Y**2)` — near zero because everything is (jointly) Gaussian, Lesson 23's miracle (b).

4. Step 5: the autocorrelation $\rho(k) = a^k$ decays geometrically — the spectral-gap story of Lesson 26 (a is the second eigenvalue's role) meeting Lesson 21's covariance. This function $R_S(k)$ is exactly what statistical inference will Fourier-transform into a power spectral density (Part II's DFT, waiting).

5. Step 6: window averages of a *dependent* process still concentrate, but more slowly than i.i.d. (var of the mean scales like $\frac{\sigma^2}{n} \cdot \frac{1+a}{1-a}$ — compute and compare). The i.i.d. CLT is the floor, not the whole story: statistical inference's ergodic-theory unit begins here.

6. Optional (the Kalman itch): the true $\mathbf{E}[S_t \mid Y_1, \ldots, Y_t]$ uses all past observations, not just Y_t. Implement the two-line recursive update (predict with a, correct with the signal-to-noise gain) and watch the MSE drop below step 3's — you have written a scalar Kalman filter, statistical inference's finale, from this booklet's parts.

Final self-check list for Part IV

From memory: run the CDF-then-differentiate recipe and the monotone-change-of-variables formula; define covariance/correlation and prove $|\rho| \le 1$ via Cauchy–Schwarz in the random-variable inner product; state and prove iterated expectations, the MMSE property of $\mathbf{E}[X \mid Y]$, the orthogonality principle, and the LLMS formulas; define the covariance matrix, prove it is PSD and transform it through $A\Sigma A^{\mathsf{T}}$; state the two Gaussian miracles and whiten a 2-D Gaussian by hand; prove Markov, Chebyshev, and the WLLN, and state the CLT with its standardization; define the Bernoulli and Poisson processes, prove the Poisson limit and exponential interarrivals; write balance equations, find a stationary distribution by hand and by eigenvector, and connect convergence rate to the spectral gap. That is B&T Chapters 4–8 at working strength — and, with Parts II–III, the full prerequisite load-out for statistical inference.

Part V: The Signals-and-Systems Core

Both Fourier analysis and statistical inference stand on *linear systems and Fourier transforms*. This part is that course in minimum lessons, following Oppenheim & Willsky (O&W), *Signals and Systems* (2nd ed.), Chapters 1–7, 9–10, in the same style. It leans on Part II throughout: complex exponentials (Lesson 9), eigenvalues and the eigenfunction idea (Lesson 10), and the DFT and convolution theorem (Lesson 12) are the finite-dimensional shadows of everything below. *Nothing in this part is needed for machine learning.* With Part V done, the undergraduate prerequisite set for machine learning, Fourier analysis, and statistical inference is complete.

Following every signals text, Part V writes the imaginary unit as j.

28 Signals and Systems

NEEDED FOR: Fourier analysis and statistical inference (prerequisite, linear-systems level). Not needed for ML.

28.1 Theory

Definition 28.1 (Signals)**.** A *continuous-time (CT) signal* is a function $x(t)$, $t \in \mathbb{R}$; a *discrete-time (DT) signal* is a sequence $x[n]$, $n \in \mathbb{Z}$ — i.e. a vector with infinitely many coordinates, which is why Part I's language keeps applying. The *energy* of a signal is $E = \int_{-\infty}^{\infty} |x(t)|^2 dt$ (CT) or $\sum_n |x[n]|^2$ (DT) — a squared norm, Lesson 2 — and its *average power* is the per-unit-time limit of the same quantity.

Definition 28.2 (Transformations of time; periodicity)**.** From x we form $x(t - t_0)$ (*shift right* by t_0), $x(-t)$ (*reversal*), and $x(at)$ (*scaling*: compression for $|a| > 1$, stretch for $|a| < 1$). A signal is *periodic* with period $T > 0$ (resp. integer $N > 0$) if $x(t + T) = x(t)$ for all t (resp. $x[n + N] = x[n]$ for all n). Any signal splits uniquely into even plus odd parts: $x = \underbrace{\frac{1}{2}(x(t) + x(-t))}_{\text{even}} + \underbrace{\frac{1}{2}(x(t) - x(-t))}_{\text{odd}}$.

Proposition 28.3 (CT vs. DT complex exponentials — the crucial asymmetry)**.** *(a) $e^{j\omega_0 t}$ is periodic for every $\omega_0 \neq 0$, with fundamental period $2\pi/|\omega_0|$; distinct frequencies give distinct signals.*

(b) $e^{j\omega_0 n}$ satisfies $e^{j(\omega_0+2\pi)n} = e^{j\omega_0 n}$ for all n: DT frequencies are identical mod 2π, so only $\omega_0 \in [-\pi, \pi)$ are genuinely different, and $\omega_0 = \pm\pi$ is the fastest possible DT oscillation ($e^{j\pi n} = (-1)^n$). Moreover $e^{j\omega_0 n}$ is periodic in n iff $\omega_0/2\pi$ is rational.

Proof. (a) $e^{j\omega_0(t+T)} = e^{j\omega_0 t}e^{j\omega_0 T} = e^{j\omega_0 t}$ iff $\omega_0 T \in 2\pi\mathbb{Z}$; the smallest positive T is $2\pi/|\omega_0|$. (b) $e^{j2\pi n} = 1$ for every integer n (Lesson 9), giving the mod-2π identity. Periodicity: $e^{j\omega_0 N} = 1$ iff $\omega_0 N = 2\pi k$ iff $\omega_0/2\pi = k/N$ is rational. □

Part (b) is the deep reason sampling (Lesson 33) can destroy information: a DT sequence cannot tell frequencies apart mod 2π. It is also why the DTFT of Lesson 32 will be periodic.

Definition 28.4 (Unit impulse and unit step)**.** In DT: $\delta[n] = 1$ if $n = 0$, else 0; and $u[n] = 1$ for $n \geq 0$, else 0; note $u[n] = \sum_{k=0}^{\infty} \delta[n-k]$ and $\delta[n] = u[n] - u[n-1]$. In CT: $u(t) = 1$ for $t > 0$, else 0, and the *unit impulse* $\delta(t)$ is the idealized limit of a pulse of width Δ, height $1/\Delta$, as $\Delta \to 0$; it is defined *operationally* by the *sifting property*

$$\int_{-\infty}^{\infty} x(t)\,\delta(t-t_0)\,dt = x(t_0)$$

for every signal x continuous at t_0. (Making δ a genuine object is distribution theory; O&W 1.4 and 2.5 give the honest operational account, and Fourier analysis develops the rigorous one — every formal δ-manipulation below can be certified by running it against a test function.)

Definition 28.5 (System properties)**.** A *system* maps input signals to output signals, $y = T\{x\}$. It is

- *memoryless* if y at time t depends only on x at time t;
- *causal* if y at time t depends only on x at times $\leq t$;
- *(BIBO) stable* if every bounded input produces a bounded output;
- *time-invariant* if shifting the input shifts the output: $x(t-t_0) \mapsto y(t-t_0)$ for all t_0;
- *linear* if $T\{ax_1 + bx_2\} = aT\{x_1\} + bT\{x_2\}$ — Lesson 4's definition, verbatim, on the vector space of signals.

Proposition 28.6. *A linear system maps the zero signal to the zero signal.*

Proof. $T\{0\} = T\{0 \cdot x\} = 0 \cdot T\{x\} = 0$ — the same one-liner as Proposition 1.4 and Lesson 4, because it is the same fact. □

28.2 Practice: by hand

Example 28.7 (Classifying systems)**.**

system	memoryless	causal	stable	TI	linear
$y(t) = 2x(t) + 3$	✓	✓	✓	✓	×
$y[n] = x[n] - x[n-1]$	×	✓	✓	✓	✓
$y(t) = x(2t)$	×	×	✓	×	✓
$y[n] = n\,x[n]$	✓	✓	×	×	✓
$y(t) = x^2(t)$	✓	✓	✓	✓	×

Spot checks worth doing: $y(t) = 2x(t) + 3$ fails linearity because zero in gives $3 \neq 0$ out. $y(t) = x(2t)$ is not causal ($y(-1) = x(-2)$ is fine, but $y(1) = x(2)$ looks ahead) and not time-invariant: shifting the input by t_0 produces $x(2t - t_0)$, but shifting the output produces $x(2(t-t_0)) = x(2t - 2t_0)$ — different. $y[n] = nx[n]$ is unstable: the bounded input $x[n] = 1$ gives the unbounded output n.

Example 28.8 (Even/odd decomposition)**.** $x[n] = u[n]$: even part $\frac{1}{2}(u[n] + u[-n]) = \frac{1}{2} + \frac{1}{2}\delta[n]$, odd part $\frac{1}{2}(u[n] - u[-n]) = \frac{1}{2}\,\mathrm{sign}[n]$. Check they sum to $u[n]$ at $n = -1, 0, 1$: $0, 1, 1$. ✓

Fourier / statistical-inference connection. "Linear plus time-invariant" is the exact class of systems Fourier methods diagonalize — Lesson 10's LTI-connection box, about to become Lessons 29–32. The properties table is not busywork: statistical inference constantly asks which manipulations of a random process preserve stationarity, and the answer is "the time-invariant ones"; Fourier analysis's transform rules each mirror one row of it.

28.3 Exercises

Exercise 28.1 (Hand)**.** Sketch (or tabulate) $x(t) = u(t+1) - u(t-2)$, then $x(2t)$, $x(t/2)$, and $x(1-t)$. For the last one: which operations, in which order?

Exercise 28.2 (Hand)**.** Classify (all five properties, with one-line justifications): (a) $y[n] = x[-n]$; (b) $y(t) = \int_{-\infty}^{t} x(\tau)\, d\tau$; (c) $y[n] = \cos(x[n])$.

Exercise 28.3 (Proof, $\star$)**.** Prove that the even/odd decomposition is unique: if $x = e_1 + o_1 = e_2 + o_2$ with e_i even and o_i odd, then $e_1 = e_2$ and $o_1 = o_2$. Then show energy splits: $\sum_n |x[n]|^2 = \sum_n |e[n]|^2 + \sum_n |o[n]|^2$ (the cross terms form an odd signal — even and odd signals are *orthogonal*, Lesson 2's Pythagoras again).

Exercise 28.4 (Proof)**.** Determine the fundamental period of $x[n] = e^{j(2\pi/3)n} + e^{j(3\pi/4)n}$ (least common period of the two parts), and prove that a sum of two periodic DT signals is always periodic — then explain why the analogous CT claim is *false* ($\cos t + \cos \pi t$).

Deeper reading. O&W 1.1–1.4 (signals, transformations, exponentials, impulses), 1.5–1.6 (systems and properties).

29 LTI Systems and Convolution

NEEDED FOR: Fourier analysis and statistical inference (prerequisite). Not needed for ML.

29.1 Theory

The sifting property writes any DT signal as a combination of shifted impulses:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\, \delta[n-k].$$

The shifted impulses are a "basis" for signal space, and the coefficients are the signal's own values. Lesson 4 taught what happens next: a linear map is determined by what it does to a basis.

Theorem 29.1 (LTI systems are convolutions)**.** *Let T be a linear time-invariant DT system and let $h = T\{\delta\}$ be its* impulse response*. Then for every input,*

$$y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k].$$

Conversely, every such convolution is LTI. The same holds in CT with $y(t) = (x * h)(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau$ and $h = T\{\delta\}$.

Proof. (DT; the CT statement is its limit, O&W 2.2.) By time invariance, $T\{\delta[\cdot - k]\} = h[\cdot - k]$. By linearity applied to the sifting representation (for inputs of finite duration this is a finite sum; the infinite case needs a continuity assumption that all systems in practice satisfy):

$$y[n] = T\Big\{ \sum_k x[k]\, \delta[\cdot - k] \Big\}[n] = \sum_k x[k]\, h[n-k].$$

Conversely, the formula is linear in x (sums split), and replacing x by $x[\cdot - n_0]$ and substituting $k' = k - n_0$ shifts y by n_0: time-invariant. □

This is Theorem 4.3 for signal space: *one* signal, h, pins down the entire system, exactly as a matrix's columns pin down a linear map. Indeed, writing $y[n] = \sum_k H_{n,k} x[k]$ with $H_{n,k} = h[n-k]$: an LTI system is an (infinite) matrix that is constant along diagonals (*Toeplitz*) — and Lesson 12's circulant matrices were the finite, wrap-around version of exactly this.

Proposition 29.2 (Algebra of convolution)**.** *$x*h = h*x$ (commutative); $x*(h_1*h_2) = (x*h_1)*h_2$ (associative — cascaded LTI systems combine into one, in either order); $x*(h_1+h_2) = x*h_1 + x*h_2$ (distributive — parallel systems add); $x*\delta = x$ (identity).*

Proof. Commutativity: substitute $m = n-k$ in the defining sum, $\sum_k x[k]h[n-k] = \sum_m x[n-m]h[m]$. Identity: sifting. Distributivity: split the sum. Associativity: write out the double sum for both sides and reindex — both equal $\sum_k \sum_m x[k]\, h_1[m]\, h_2[n-k-m]$. □

Theorem 29.3 (Causality and stability, read off h)**.** *An LTI system is causal iff $h[n] = 0$ for all $n < 0$; it is BIBO stable iff h is absolutely summable, $\sum_k |h[k]| < \infty$ (CT: $h(t) = 0$ for $t < 0$; $\int |h| < \infty$).*

Proof. Causality. Write $y[n] = \sum_k h[k]x[n-k]$ (commuted form). If $h[k] = 0$ for $k < 0$, the sum involves only $x[n], x[n-1], \ldots$: causal. If $h[k_0] \neq 0$ for some $k_0 < 0$, the input δ produces output h, which is nonzero at time $k_0 < 0$ although the input was zero before time 0: not causal.

Stability, sufficiency. If $|x[n]| \le B$ for all n, the triangle inequality gives

$$|y[n]| \le \sum_k |h[k]|\, |x[n-k]| \le B \sum_k |h[k]| < \infty.$$

Necessity. Suppose $\sum_k |h[k]| = \infty$. Feed in the bounded input $x[n] = \operatorname{sign}\big(h[-n]\big)$ (magnitude ≤ 1). Then $y[0] = \sum_k h[k]x[-k] = \sum_k h[k]\operatorname{sign}(h[k]) = \sum_k |h[k]| = \infty$: a bounded input with an unbounded output. □

Difference and differential equations. Most implemented systems are described by linear constant-coefficient equations. The simplest, run as a recursion, already shows the shape of everything: $y[n] - ay[n-1] = x[n]$ with initial rest gives, feeding in δ: $h[0] = 1$, $h[1] = a$, $h[2] = a^2, \ldots$, i.e.

$$h[n] = a^n u[n],$$

which is causal always and stable iff $|a| < 1$ (geometric series) — Lesson 10's "$|\lambda| < 1$ decays" as a system-theory fact. Lesson 34 gives the systematic transform treatment.

29.2 Practice: by hand

Example 29.4 (A convolution table)**.** $x = (1, 2, 1)$ at $n = 0, 1, 2$ and $h = (1, 1)$ at $n = 0, 1$. Slide-and-sum:

$$y[0] = 1, \quad y[1] = 2 + 1 = 3, \quad y[2] = 1 + 2 = 3, \quad y[3] = 1; \qquad y = (1, 3, 3, 1).$$

Sanity checks: length $3+2-1 = 4$; $\sum y = (\sum x)(\sum h) = 4 \cdot 2$ ✓ (that identity is the convolution property at frequency zero, Lesson 31). Commuted order gives the same table transposed. ✓

Example 29.5 (Pulse $*$ pulse = triangle)**.** Let $x(t) = h(t) = 1$ on $[0, 1]$, else 0. Then $y(t) = \int x(\tau)h(t-\tau)d\tau$ is the overlap length of $[0, 1]$ and $[t-1, t]$: $y(t) = t$ on $[0, 1]$, $2 - t$ on $[1, 2]$, 0 elsewhere — a triangle. Convolution smooths: the output is continuous although both inputs jump. (Convolve again with the pulse and the result is piecewise-parabolic; each convolution adds a degree of smoothness — the picture behind Fourier analysis's "smoother signal ⇔ faster-decaying transform.")

Example 29.6 (First-order smoother)**.** $y[n] = 0.5\,y[n-1] + x[n]$, input $x = u$: $y[0] = 1$, $y[1] = 1.5$, $y[2] = 1.75$, $y[3] = 1.875$, climbing to $\sum_k (0.5)^k = 2$. The step response is the running convolution $u * h$, and its limit is $\sum h[k]$ — the "DC gain."

Fourier / statistical-inference connection. Convolution is Fourier analysis's second-most-important operation (after the transform itself), and the pair "convolution in time ↔ multiplication in frequency" (Lessons 31–32, finite version already proved as Theorem 12.6) is Fourier analysis's engine. In statistical inference, filtered random processes are convolutions $y = h * x$ with x random; the autocorrelation of the output involves h convolved with itself, and stability (Theorem 29.3) is what keeps output variances finite.

29.3 Exercises

Exercise 29.1 (Hand)**.** Compute $(1, 1, 1) * (1, -1)$ and $(1, 2) * (1, 2)$ by table. Check the lengths and the sum-product identity of Example 29.4.

Exercise 29.2 (Hand)**.** Compute $u * h$ for $h[n] = (0.5)^n u[n]$ in closed form (finite geometric series), and identify the DC gain. Then find the impulse response of the *cascade* of that system with $y[n] = x[n] - x[n-1]$ (a differencer), using associativity — what does the cascade do to a constant input, and why is that obvious from h?

Exercise 29.3 (Proof, $\star$)**.** Prove that the cascade of two causal LTI systems is causal and of two stable LTI systems is stable, directly from Theorem 29.3 and the convolution formula ($h = h_1 * h_2$; bound $\sum |h|$ by $\big(\sum |h_1|\big)\big(\sum |h_2|\big)$).

Exercise 29.4 (Proof)**.** Show that an LTI system is memoryless iff $h[n] = c\,\delta[n]$ for some constant c — so *every* nontrivial LTI system has memory, and "matrix diagonal" is the right mental image.

Deeper reading. O&W 2.1–2.3 (convolution sum and integral, properties), 2.4 (difference and differential equations), 2.5 (singularity functions).

30 Fourier Series

NEEDED FOR: Fourier analysis (its opening act) and statistical inference (prerequisite). Not needed for ML.

30.1 Theory

Proposition 30.1 (Complex exponentials are eigenfunctions of LTI systems)**.** *Let h be the impulse response of a CT LTI system, and let $s \in \mathbb{C}$ be such that $H(s) = \int_{-\infty}^{\infty} h(\tau)e^{-s\tau}d\tau$ converges. Then the input e^{st} produces the output $H(s)\,e^{st}$. In DT, z^n produces $H(z)\,z^n$ with $H(z) = \sum_k h[k]z^{-k}$.*

Proof.

$$y(t) = \int h(\tau)\, e^{s(t-\tau)}d\tau = e^{st}\int h(\tau)e^{-s\tau}d\tau = H(s)\, e^{st}. \qquad \square$$

Eigenvector in, eigenvalue out (Lesson 10) — so if we can write signals as combinations of exponentials, LTI systems act diagonally. Fourier series do exactly that for periodic signals.

Lemma 30.2 (Orthonormality of harmonics)**.** *Fix $T > 0$ and $\omega_0 = 2\pi/T$. Then*

$$\frac{1}{T}\int_T e^{jk\omega_0 t}\,\overline{e^{jl\omega_0 t}}\,dt = \frac{1}{T}\int_T e^{j(k-l)\omega_0 t}\,dt = \begin{cases} 1 & k = l, \\ 0 & k \neq l, \end{cases}$$

the integral over any interval of length T.

Proof. For $k = l$ the integrand is 1. For $m = k - l \neq 0$: $\int_0^T e^{jm\omega_0 t}dt = \frac{e^{jm\omega_0 T}-1}{jm\omega_0} = \frac{e^{j2\pi m}-1}{jm\omega_0} = 0$. $\square$

This is Lemma 9.2 with the sum over roots of unity replaced by an integral over the circle: the harmonics $\{e^{jk\omega_0 t}\}$ are an orthonormal list in the inner product $\langle f,\, g\rangle = \frac{1}{T}\int_T f\bar{g}$.

Theorem 30.3 (Fourier series)**.** *Let x be periodic with period T and representable as $x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$ (see the convergence remark below). Then the coefficients are forced:*

$$a_k = \frac{1}{T}\int_T x(t)\, e^{-jk\omega_0 t}\,dt,$$

and Parseval's relation holds: $\dfrac{1}{T}\displaystyle\int_T |x(t)|^2 dt = \sum_k |a_k|^2.$

Proof. Multiply the expansion by $\overline{e^{jl\omega_0 t}} = e^{-jl\omega_0 t}$, integrate over one period, and swap sum and integral (legal under the convergence assumptions):

$$\frac{1}{T}\int_T x(t)e^{-jl\omega_0 t}dt = \sum_k a_k \cdot \frac{1}{T}\int_T e^{j(k-l)\omega_0 t}dt = a_l$$

by Lemma 30.2 — exactly Proposition 6.2's "coordinates in an orthonormal basis are inner products," $a_l = \langle x,\, e_l\rangle$. Parseval is the Pythagorean theorem in the same inner product: expand $\langle x,\, x\rangle = \sum_k \sum_l a_k \bar{a}_l \langle e_k,\, e_l\rangle = \sum_k |a_k|^2$. $\square$

Convergence, honestly. For x with $\int_T |x|^2 < \infty$, the partial sums converge to x in energy (they are orthogonal projections onto $\text{span}\{e^{jk\omega_0 t} : |k| \le N\}$ — Theorem 6.3 verbatim — hence the best least-squares approximants at every N). Pointwise convergence needs the Dirichlet conditions (O&W 3.4); at a jump the series converges to the midpoint, and near a jump the partial sums overshoot by $\approx 9\%$ no matter how many terms are kept — the *Gibbs phenomenon*.

Two properties used constantly (both one-line from the coefficient formula): a time shift multiplies coefficients by a phase, $x(t - t_0) \leftrightarrow a_k e^{-jk\omega_0 t_0}$; and for real x, $a_{-k} = \overline{a_k}$ (conjugate symmetry — take conjugates inside the integral).

Theorem 30.4 (Filtering a periodic signal)**.** *If $x = \sum_k a_k e^{jk\omega_0 t}$ drives an LTI system with frequency response $H(j\omega) = \int h(\tau) e^{-j\omega\tau} d\tau$, the output is*

$$y(t) = \sum_k a_k \, H(jk\omega_0) \, e^{jk\omega_0 t} :$$

each harmonic passes through independently, scaled by the eigenvalue at its frequency.

Proof. Linearity plus Proposition 30.1 at $s = jk\omega_0$, term by term. □

DT Fourier series = the DFT. A DT signal with period N has only N distinct harmonics ($e^{j(2\pi/N)kn}$, $k = 0, \dots, N-1$, by Proposition 28.3b), so the DT Fourier series is a finite sum — and its analysis/synthesis pair is exactly Lesson 12's DFT and inverse DFT (up to the $1/N$ placement). Everything proved there (unitarity, Parseval, the convolution theorem for circulants) *is* the DT Fourier series theory.

30.2 Practice: by hand

Example 30.5 (The square wave, the one to memorize)**.** Let x have period T, with $x(t) = 1$ for $|t| < T_1$ and 0 elsewhere in the period. Then $a_0 = 2T_1/T$, and for $k \neq 0$:

$$a_k = \frac{1}{T} \int_{-T_1}^{T_1} e^{-jk\omega_0 t} dt = \frac{e^{jk\omega_0 T_1} - e^{-jk\omega_0 T_1}}{jk\omega_0 T} = \frac{2\sin(k\omega_0 T_1)}{k\omega_0 T} = \frac{\sin(k\omega_0 T_1)}{k\pi}.$$

For the half-duty case $T = 4T_1$ ($\omega_0 T_1 = \pi/2$): $a_k = \frac{\sin(k\pi/2)}{k\pi}$ — $\frac{1}{2}$, $\frac{1}{\pi}$, 0, $-\frac{1}{3\pi}$, 0, $\frac{1}{5\pi}$, … for $k = 0, 1, 2, 3, 4, 5$. The coefficients decay like $1/k$: slow, because the signal jumps. (Smooth signals decay faster — differentiate the shift-and-phase property to see each derivative buy one more power of $1/k$.)

Example 30.6 (Filtering the square wave)**.** Send the $T = 4T_1$ square wave through an ideal lowpass filter keeping only $|k| \le 1$: the output is $\frac{1}{2} + \frac{2}{\pi}\cos(\omega_0 t)$ — a DC level plus one smooth cosine; all edges gone. Keep $|k| \le 5$ and the corners begin to return, with Gibbs ears at the jumps. Filtering is coefficient surgery: Theorem 30.4 with $H = 1$ in the passband and 0 outside.

Fourier connection. A Fourier-analysis course opens with precisely this lesson — Osgood's first chapters are Fourier series, coefficients-as-inner-products, and Parseval — before taking $T \to \infty$ to reach the Fourier transform (as Lesson 31 does next). Arriving with Lemma 30.2 and Theorem 30.3 already owned turns those weeks into review, exactly as the Lesson 9 box promised.

30.3 Exercises

Exercise 30.1 (Hand)**.** Compute the Fourier series coefficients of $x(t) = \sin^2(\omega_0 t)$ directly (no integrals: rewrite with the double-angle identity and read off the a_k). Which k are nonzero?

Exercise 30.2 (Hand)**.** From Example 30.5 with $T = 4T_1$, evaluate the series at $t = 0$ to prove $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots = \frac{\pi}{4}$ (Leibniz), and apply Parseval to prove $\sum_{k \text{ odd}} \frac{1}{k^2} = \frac{\pi^2}{8}$.

Exercise 30.3 (Proof, $\star$)**.** Prove the differentiation property: if x has coefficients a_k and x' is also nice-enough periodic, then x' has coefficients $jk\omega_0 a_k$ (integrate by parts over one period; the boundary terms cancel by periodicity). Conclude: m derivatives of smoothness force $|a_k| = O(1/k^m)$ decay.

Exercise 30.4 (Proof)**.** Prove the DT orthogonality relation $\frac{1}{N}\sum_{n=0}^{N-1} e^{j(2\pi/N)(k-l)n} = \delta_{kl}$ (for $k, l \in \{0, \ldots, N-1\}$) from Lemma 9.2, and write out the resulting DT Fourier series pair. Compare with Lesson 12's DFT to fix where the $1/N$ sits in each convention.

Deeper reading. O&W 3.2 (eigenfunctions), 3.3–3.5 (CT Fourier series, convergence, properties), 3.6–3.7 (DT Fourier series), 3.8–3.9 (LTI systems and filtering).

31 The Continuous-Time Fourier Transform

NEEDED FOR: Fourier analysis (its core object) and statistical inference (prerequisite). Not needed for ML.

31.1 Theory

Aperiodic signals have no harmonics to expand in — but an aperiodic signal is a periodic one whose period went to infinity. Carrying Theorem 30.3 through that limit (O&W 4.1 does it carefully: $Ta_k \to X(j\omega)$ at $\omega = k\omega_0$, and the Fourier sum becomes a Riemann integral) yields the transform pair that Fourier analysis lives on:

Definition 31.1 (CT Fourier transform)**.**

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt \qquad \Longleftrightarrow \qquad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)\, e^{j\omega t}\, d\omega.$$

$X(j\omega)$ is the signal's density of content at frequency ω — the continuous analogue of "coefficient = inner product with a harmonic."

Example 31.2 (The pairs to know cold)**.** Each is a direct integral:

$$\begin{aligned}
e^{-at}u(t)\ (a>0) &\longleftrightarrow \frac{1}{a+j\omega} && \text{(one-sided decay)}\\
x(t)=1,\ |t|<T_1 &\longleftrightarrow \frac{2\sin(\omega T_1)}{\omega} && \text{(rectangle} \leftrightarrow \text{sinc)}\\
\delta(t) &\longleftrightarrow 1 && \text{(all frequencies equally)}\\
1 \longleftrightarrow 2\pi\,\delta(\omega), \qquad e^{j\omega_0 t} &\longleftrightarrow 2\pi\,\delta(\omega-\omega_0) && \text{(tones are frequency impulses)}\\
\cos\omega_0 t &\longleftrightarrow \pi\big[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\big].
\end{aligned}$$

The first: $\int_0^\infty e^{-(a+j\omega)t}dt = \frac{1}{a+j\omega}$. The second: $\int_{-T_1}^{T_1} e^{-j\omega t}dt = \frac{2\sin(\omega T_1)}{\omega}$ — the same computation as the square wave's a_k (Example 30.5), now with a continuous frequency variable. The δ pairs are certified by the sifting property inside the inversion integral. Also worth knowing: a Gaussian transforms to a Gaussian ($e^{-t^2/2} \leftrightarrow \sqrt{2\pi}\, e^{-\omega^2/2}$ — Exercise 19.3's integral, completed in the exponent), the unique fixed-point shape.

Theorem 31.3 (The property toolbox)**.** *Let $x \leftrightarrow X$ and $y \leftrightarrow Y$. Then:*

(a) Linearity: $ax + by \leftrightarrow aX + bY$.

(b) Time shift: $x(t - t_0) \leftrightarrow e^{-j\omega t_0}X(j\omega)$ — *magnitude untouched, phase tilted.*

(c) Scaling: $x(at) \leftrightarrow \frac{1}{|a|}X\big(j\frac{\omega}{a}\big)$ *for $a \neq 0$ — compress time, stretch frequency, and vice versa.*

(d) Duality: *if $x(t) \leftrightarrow X(j\omega)$ then $X(jt) \leftrightarrow 2\pi\, x(-\omega)$.*

(e) Differentiation: $x'(t) \leftrightarrow j\omega\, X(j\omega)$.

(f) Parseval: $\displaystyle\int |x(t)|^2 dt = \frac{1}{2\pi}\int |X(j\omega)|^2 d\omega$.

Proof. (a) integrals are linear. (b) substitute $\tau = t - t_0$: $\int x(t-t_0)e^{-j\omega t}dt = e^{-j\omega t_0}\int x(\tau)e^{-j\omega\tau}d\tau$. (c) substitute $\tau = at$; for $a > 0$, $\int x(at)e^{-j\omega t}dt = \frac{1}{a}\int x(\tau)e^{-j(\omega/a)\tau}d\tau$; for $a < 0$ the limits flip, producing the $|a|$ — this is Lesson 14's stretch theorem in one dimension, with $1/|a|$ as the volume factor and ω/a as the contragradient frequency variable. (d) the inversion formula *is* a forward transform up to sign and 2π; write it at $-\omega$ and rename variables. (e) differentiate the inversion formula under the integral: each $e^{j\omega t}$ contributes a factor $j\omega$. (f) is Theorem 12.2(c) in integral form: the transform is (up to $\sqrt{2\pi}$) a unitary map, and lengths are preserved; the direct proof expands $|x|^2 = x\bar{x}$ via the inversion formula and swaps integrals. □

Theorem 31.4 (The convolution property — the punchline)**.**

$$y = h * x \qquad \Longleftrightarrow \qquad Y(j\omega) = H(j\omega)\, X(j\omega),$$

and dually, multiplication in time is $\frac{1}{2\pi}\times$ convolution in frequency: $x(t)p(t) \leftrightarrow \frac{1}{2\pi}(X * P)(j\omega)$.

Proof. Transform the convolution and swap the integrals:

$$Y(j\omega) = \int\Big(\int x(\tau)h(t-\tau)d\tau\Big)e^{-j\omega t}dt = \int x(\tau)e^{-j\omega\tau}\Big(\int h(t-\tau)e^{-j\omega(t-\tau)}dt\Big)d\tau = X(j\omega)H(j\omega),$$

using the shift substitution inside the inner integral. The dual statement follows by duality (or the mirror-image computation). □

This is Theorem 12.6 — "circulants are diagonalized by the DFT" — with integrals in place of sums: LTI systems are diagonal in the Fourier basis, with $H(j\omega)$ (the transform of the impulse response) as the eigenvalue at frequency ω. Solving systems, designing filters, and inverting channels all reduce to multiplication and division of transforms.

31.2 Practice: by hand

Example 31.5 (Two-sided decay)**.** $x(t) = e^{-a|t|}$, $a > 0$: split at 0 and use the first pair twice (once reversed),

$$X(j\omega) = \frac{1}{a + j\omega} + \frac{1}{a - j\omega} = \frac{2a}{a^2 + \omega^2},$$

real and even, as it must be for a real even signal. As $a \to 0$ the time signal flattens toward 1 and the transform sharpens toward $2\pi\delta(\omega)$ — scaling property intuition made visible.

Example 31.6 (RC lowpass filter)**.** The circuit equation $RC\,y'(t) + y(t) = x(t)$ has impulse response $h(t) = \frac{1}{RC}e^{-t/RC}u(t)$. Transforming with properties (e) and (a): $(1 + j\omega RC)Y = X$, so

$$H(j\omega) = \frac{1}{1 + j\omega RC}, \qquad |H(j\omega)| = \frac{1}{\sqrt{1 + (\omega RC)^2}}:$$

gain ≈ 1 for $\omega \ll 1/RC$, rolling off like $1/\omega$ above the *cutoff* $\omega_c = 1/RC$ (where $|H| = 1/\sqrt{2}$, the "-3 dB point"). One first-order system, read three ways: differential equation, impulse response, frequency response.

Fourier / statistical-inference connection. Lessons 30–31 are the first half of Fourier analysis in miniature: series, transform, properties, convolution. What Fourier analysis adds is depth (distributions to make $\delta(\omega)$ rigorous, the sampling and DFT chapters — Lessons 33 and 12 here — and applications from diffraction to CT scans). Statistical inference's stake: the power spectral density of a stationary process is the Fourier transform of its autocorrelation (Lesson 21), and filtered noise obeys $S_y(\omega) = |H(j\omega)|^2 S_x(\omega)$ — Theorem 31.4 applied to second moments. That formula is the reason statistical inference lists this material as a prerequisite.

31.3 Exercises

Exercise 31.1 (Hand)**.** Using pairs and properties only (no new integrals): find the transforms of (a) $e^{-3(t-2)}u(t-2)$; (b) $te^{-at}u(t)$ (differentiate $\frac{1}{a+j\omega}$ with respect to ω — which property is that?); (c) $\sin\omega_0 t$.

Exercise 31.2 (Hand)**.** For the RC filter of Example 31.6 with $RC = 10^{-3}$: what are the cutoff in Hz, $|H|$ at 50 Hz and at 10 kHz, and the output when $x(t) = \cos(2\pi \cdot 100\,t) + \cos(2\pi \cdot 10^4\,t)$? (Use Theorem 30.4: evaluate H at each tone.)

Exercise 31.3 (Proof, $\star$)**.** Prove the modulation property $x(t)\cos(\omega_0 t) \leftrightarrow \frac{1}{2}X(j(\omega - \omega_0)) + \frac{1}{2}X(j(\omega + \omega_0))$ from linearity and the $e^{j\omega_0 t}$ shift-in-frequency property (prove that one first, by the mirror of Theorem 31.3b). This two-line result is AM radio (O&W Ch. 8), and Lesson 33 reuses it as the sampling theorem's engine.

Exercise 31.4 (Proof)**.** Derive the transform of the triangle $\big(1 - |t|/T\big)$ on $|t| < T$ by writing it as $\frac{1}{T}(\text{rect} * \text{rect})$ and applying Theorem 31.4: the answer is a sinc^2. Confirm positivity of the transform and explain it (autocorrelation of a real signal — Lesson 21's Cauchy–Schwarz guarantees the peak is at 0).

Deeper reading. O&W 4.1–4.3 (the transform and its properties), 4.4–4.5 (convolution and multiplication properties), 4.7 (LCCDE systems); Osgood, *Lectures on the Fourier Transform and Its Applications*, chapters 2–4 tell the same story with more depth and better jokes.

32 The Discrete-Time Fourier Transform and Frequency Response

NEEDED FOR: Fourier analysis (DFT background) and statistical inference (prerequisite; DSP fluency). Not needed for ML.

32.1 Theory

Definition 32.1 (DTFT)**.** For a DT signal $x[n]$,

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \qquad \Longleftrightarrow \qquad x[n] = \frac{1}{2\pi}\int_{2\pi} X(e^{j\omega})\, e^{j\omega n}\, d\omega.$$

$X(e^{j\omega})$ is 2π-periodic — immediately from Proposition 28.3(b) — which is why one period ($\omega \in [-\pi, \pi]$) carries everything and the inversion integral runs over one period only. The notation $X(e^{j\omega})$ advertises the periodicity: the transform is a function on the unit circle, foreshadowing Lesson 34's z-plane.

Example 32.2 (The two DT pairs to know cold)**.** *Geometric decay:* for $|a| < 1$,

$$a^n u[n] \;\longleftrightarrow\; \sum_{n\geq 0} (ae^{-j\omega})^n = \frac{1}{1-ae^{-j\omega}},$$

a geometric series, convergent because $|ae^{-j\omega}| = |a| < 1$.

Rectangular window: $x[n] = 1$ for $|n| \leq N_1$:

$$X(e^{j\omega}) = \sum_{n=-N_1}^{N_1} e^{-j\omega n} = \frac{\sin\left(\omega(N_1 + \frac{1}{2})\right)}{\sin(\omega/2)},$$

by the same finite geometric sum as Lemma 9.2 — the *periodic sinc* (Dirichlet kernel), the DT sibling of rect $\leftrightarrow$ sinc.

All of Theorem 31.3 transfers (shift $\to$ phase, Parseval with $\frac{1}{2\pi}\int_{2\pi}$, etc.), and so does the punchline, by the same swap-the-sums proof as Theorem 31.4:

Theorem 32.3 (DT convolution property)**.** *$y = x * h \iff Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$, where $H(e^{j\omega}) = \sum_k h[k]e^{-j\omega k}$ is the system's* frequency response *— the eigenvalue function of Proposition 30.1 evaluated on the unit circle $z = e^{j\omega}$.*

Frequency response from a difference equation. For $\sum_k a_k y[n-k] = \sum_k b_k x[n-k]$, transform both sides with the shift property and divide:

$$H(e^{j\omega}) = \frac{\sum_k b_k e^{-j\omega k}}{\sum_k a_k e^{-j\omega k}}$$

— a ratio of polynomials in $e^{-j\omega}$, computable by inspection. The first-order smoother $y[n] - ay[n-1] = x[n]$ gives $H = 1/(1 - ae^{-j\omega})$, consistent with $h[n] = a^n u[n]$ and Example 32.2: for $0 < a < 1$, $|H|$ is largest at $\omega = 0$ and smallest at $\omega = \pi$ — a lowpass; flip the sign of a and the roles swap — a highpass.

The DFT is the sampled DTFT. For a length-N signal, Lesson 12's DFT values are exactly

$$\hat{x}_k = X(e^{j\omega})\Big|_{\omega=2\pi k/N} :$$

N equally spaced samples of the DTFT around the circle. That is why `np.fft.fft` computes "the spectrum": it evaluates this lesson's object on a grid, in $O(N \log N)$.

Why ideal filters are ideals. The perfect lowpass $H(e^{j\omega}) = 1$ for $|\omega| < \omega_c$, else 0, inverts to $h[n] = \frac{\sin(\omega_c n)}{\pi n}$ — nonzero for all $n < 0$ (noncausal) and decaying only like $1/|n|$, so $\sum |h[n]| = \infty$ (unstable by Theorem 29.3). Realizable filters can only approximate the ideal, trading passband flatness, rolloff sharpness, and delay — the design space O&W Chapter 6 maps and DSP courses inhabit.

32.2 Practice: by hand

Example 32.4 (Reading a two-point filter)**.** $y[n] = \frac{1}{2}(x[n] + x[n-1])$: $H(e^{j\omega}) = \frac{1}{2}(1 + e^{-j\omega}) = e^{-j\omega/2}\cos(\omega/2)$. So $|H(1)| = 1$ at DC and $H = 0$ at $\omega = \pi$: the two-point averager passes constants and annihilates $(-1)^n$ — Example 12.8's observation ("averaging kills the Nyquist alternation"), re-derived in one line from the frequency response. The differencer $y[n] = x[n] - x[n-1]$ has $H = 1 - e^{-j\omega} = 2je^{-j\omega/2}\sin(\omega/2)$: zero at DC, maximal at π — the complementary highpass.

Example 32.5 (First-order filter numbers)**.** $a = 0.9$: $|H(e^{j0})| = \frac{1}{1-0.9} = 10$ and $|H(e^{j\pi})| = \frac{1}{1+0.9} \approx 0.53$ — a 19:1 preference for slow signals. The half-power frequency solves $|1 - 0.9e^{-j\omega}|^2 = 2(0.1)^2$, giving $\omega_c \approx 0.105$: narrow, matching the long memory $h[n] = (0.9)^n$ (time constant $\approx 1/(1-a) = 10$ samples). Bandwidth and memory are reciprocal — the DT face of the RC filter's $\omega_c = 1/RC$.

> **Fourier / statistical-inference connection.** Fourier analysis's DFT chapter is this lesson plus Lesson 12 plus sampling; arriving with "DFT = sampled DTFT" in hand collapses it. For statistical inference (and any DSP work), the frequency response is the everyday tool: white noise into H comes out with spectrum $|H(e^{j\omega})|^2$, and the AR(1) process of Capstone IV is exactly white noise through this lesson's first-order filter — compute $|H|^2$ for $a = 0.95$ and you have that capstone's power spectral density.

32.3 Exercises

Exercise 32.1 (Hand)**.** Compute the DTFT of $x = \delta[n] - \delta[n-2]$, factor out the phase, and locate its zeros in $[-\pi, \pi]$. What tones does this filter annihilate?

Exercise 32.2 (Hand)**.** For the three-point averager $y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$: find $H(e^{j\omega})$ via the geometric sum (Example 32.2's window, shifted), locate its zeros, and evaluate $|H|$ at $\omega = 0,\ 2\pi/3,\ \pi$.

Exercise 32.3 (Proof, $\star$)**.** Prove Parseval for the DTFT, $\sum_n |x[n]|^2 = \frac{1}{2\pi}\int_{2\pi} |X(e^{j\omega})|^2 d\omega$, by inserting the definition of X, swapping sum and integral, and using $\frac{1}{2\pi}\int_{2\pi} e^{j\omega(m-n)}d\omega = \delta_{mn}$ (Lemma 30.2's DT twin — prove that too).

Exercise 32.4 (Proof)**.** Show that if $x[n]$ is real, $X(e^{-j\omega}) = \overline{X(e^{j\omega})}$, and deduce that $|H|$ of any real-coefficient filter is an even function of ω — so plotting $[0, \pi]$ suffices, which is why every DSP plot you will ever see stops at π (or at the "Nyquist frequency" in Hz).

Deeper reading. O&W 5.1–5.4 (DTFT, properties, convolution), 5.8 (LCCDE frequency responses), 6.1–6.4 (magnitude–phase, ideal vs. nonideal filters), 6.6 (first-order DT systems). For the dedicated discrete-time treatment, Oppenheim–Schafer–Buck, *Discrete-Time Signal Processing* (2nd ed.), Chs. 2–5, is the natural expansion of this lesson and Lessons 36–38.

33 Sampling

NEEDED FOR: Fourier analysis (a signature topic) and statistical inference (prerequisite; all data is sampled). Not needed for ML.

33.1 Theory

Sampling bridges the CT world of Lessons 30–31 and the DT world of Lesson 32 — and explains when the bridge loses nothing.

Lemma 33.1 (Spectrum of the impulse train)**.** *Let $p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$ and $\omega_s = 2\pi/T$. Then*

$$P(j\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s) :$$

an impulse train in time is an impulse train in frequency.

Proof. p is periodic with period T; its Fourier series coefficients are $a_k = \frac{1}{T}\int_{-T/2}^{T/2} \delta(t) e^{-jk\omega_0 t} dt = \frac{1}{T}$ for every k (sifting). So $p = \frac{1}{T}\sum_k e^{jk\omega_s t}$, and transforming term by term with the pair $e^{j\omega_0 t} \leftrightarrow 2\pi\delta(\omega - \omega_0)$ (Example 31.2) gives the claim. □

Theorem 33.2 (Spectrum replication)**.** *Let $x_p(t) = x(t)\,p(t)$ (the impulse-sampled signal, carrying exactly the sample values $x(nT)$). Then*

$$X_p(j\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X\big(j(\omega - k\omega_s)\big) :$$

sampling in time replicates the spectrum at every multiple of ω_s, scaled by $1/T$.

Proof. Multiplication in time is $\frac{1}{2\pi}\times$ convolution in frequency (Theorem 31.4), and convolving X with the frequency impulse train of Lemma 33.1 places a copy of X at each $k\omega_s$ (sifting, once per impulse):

$$X_p = \frac{1}{2\pi}\, X * P = \frac{1}{2\pi} \cdot \frac{2\pi}{T} \sum_k X\big(j(\omega - k\omega_s)\big). \qquad \square$$

Theorem 33.3 (Sampling theorem (Nyquist–Shannon))**.** *Suppose x is* band-limited*: $X(j\omega) = 0$ for $|\omega| > \omega_M$. If*

$$\omega_s > 2\omega_M$$

then the replicas in Theorem 33.2 do not overlap, and x is exactly recoverable from its samples: pass x_p through an ideal lowpass filter of gain T and cutoff $\omega_c \in (\omega_M,\ \omega_s - \omega_M)$, which selects the $k = 0$ copy. Equivalently, in the time domain,

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)\, \frac{\sin\big(\pi(t - nT)/T\big)}{\pi(t - nT)/T}$$

*(*sinc interpolation*: each sample carries a sinc, and the sincs vanish at every other sample instant). The critical rate $2\omega_M$ is the* Nyquist rate*. If $\omega_s < 2\omega_M$, the replicas overlap and add* — aliasing — *and reconstruction is impossible in general.*

Proof. Non-overlap: copy k occupies $[k\omega_s - \omega_M,\ k\omega_s + \omega_M]$; consecutive copies are disjoint iff $\omega_s - \omega_M > \omega_M$. The ideal lowpass with gain T then returns exactly $X(j\omega)$, hence x. For the interpolation formula: the filter's impulse response is $T \cdot \frac{\sin(\omega_c t)}{\pi t}$ (the inverse transform of a rect — Example 31.2 read backwards through duality), and the filter output is $x_p * h = \sum_n x(nT)\, h(t - nT)$ (convolution with an impulse train of weighted impulses; sifting once per term); with $\omega_c = \omega_s/2$ this is the displayed sum. □

Aliasing, concretely. Sample $\cos(\omega_0 t)$ with $\omega_s/2 < \omega_0 < \omega_s$: the replica arithmetic puts spectral lines at $\pm(\omega_s - \omega_0)$ inside the base band, so the reconstruction is $\cos\big((\omega_s - \omega_0)t\big)$ — a *lower* frequency impersonating the original. Wagon wheels spinning backwards in films, and the requirement of an analog *anti-aliasing* lowpass filter in front of every real sampler, are both this computation.

DT processing of CT signals. The practical payoff (O&W 7.4): sample above Nyquist, apply a DT filter $H_d(e^{j\omega})$, reconstruct. Within the band, the overall CT system is LTI with frequency response $H_d(e^{j\omega T})$ — so software can implement any (band-limited) analog filter, which is why signal processing moved into code.

33.2 Practice: by hand

Example 33.4 (Alias arithmetic)**.** $x(t) = \cos(2\pi \cdot 60\, t)$ sampled at $f_s = 100\,\text{Hz}$ (below Nyquist $= 120\,\text{Hz}$): the reconstructed tone appears at $100 - 60 = 40\,\text{Hz}$. At $f_s = 150\,\text{Hz}$ (above Nyquist): $60\,\text{Hz}$ survives exactly. Checking with samples: at $f_s = 100$, $x(n/100) = \cos(2\pi \cdot 0.6\, n) = \cos(2\pi \cdot 0.6n - 2\pi n) = \cos(2\pi \cdot (-0.4)n) = \cos(2\pi \cdot 0.4\, n)$ — the samples of a $40\,\text{Hz}$ tone, indistinguishable (Proposition 28.3(b) again: DT frequencies live mod 2π).

Example 33.5 (Budgeting a sampler)**.** Speech is intelligible below $\sim$4 kHz: telephony samples at 8 kHz. Hearing tops out near 20 kHz: CDs sample at 44.1 kHz — just above Nyquist, leaving a narrow $[20, 22.05]$ kHz guard band for the anti-aliasing filter to roll off in, which is why audiophile debates about that filter exist.

Fourier / statistical-inference connection. The sampling theorem is a Fourier-analysis centerpiece — Osgood devotes a chapter to it and its interplay with the DFT — and this

proof (replicate, then window) is the one taught there. For statistical inference, sampling is the standing assumption: every "random process" met in data is samples of something, sampled band-limited noise stays faithfully represented, and undersampled noise *folds* its out-of-band power into the base band (aliased noise), a standard exam trap. And the whole story is Proposition 28.3's asymmetry cashing out: DT cannot resolve frequencies mod ω_s, so you must not put content there.

33.3 Exercises

Exercise 33.1 (Hand)**.** For each, give the Nyquist rate and what appears at the output if sampled at $\omega_s = 2\pi \cdot 1000$: (a) $\cos(2\pi \cdot 300\, t)$; (b) $\cos(2\pi \cdot 700\, t)$; (c) their sum. (For (c): can any post-processing undo the damage? Why not?)

Exercise 33.2 (Hand)**.** A signal has spectrum supported on $[-2\pi \cdot 5\,\text{kHz},\ 2\pi \cdot 5\,\text{kHz}]$. It is sampled at 8 kHz *after* an anti-aliasing filter with cutoff 4 kHz. What band survives end-to-end? What would the spectrum near 3.5 kHz look like *without* the filter (which input frequencies fold there)?

Exercise 33.3 (Proof, $\star$)**.** Prove the samples-orthogonality behind the interpolation formula: the shifted sincs $\phi_n(t) = \operatorname{sinc}\big((t-nT)/T\big)$ satisfy $\phi_n(mT) = \delta_{mn}$ (immediate) *and* are orthogonal in L^2: $\int \phi_n \phi_m\, dt = T\,\delta_{mn}$ (transform each sinc to a rect via Example 31.2/duality, and integrate the product in the frequency domain via Parseval — phases from the shifts give Lemma 30.2's integral). So band-limited reconstruction is one more orthonormal expansion, and Theorem 6.3's geometry applies verbatim.

Exercise 33.4 (Proof)**.** Suppose x is band-limited to ω_M and you sample at exactly the Nyquist rate $\omega_s = 2\omega_M$. Exhibit a nonzero band-limited signal all of whose samples vanish (consider $\sin(\omega_M t)$), concluding that the strict inequality in Theorem 33.3 matters.

Deeper reading. O&W 7.1–7.3 (sampling theorem, reconstruction, aliasing), 7.4 (DT processing of CT signals); the modulation view of sampling is O&W 8.5–8.6 (pulse-amplitude modulation).

34 Laplace and z-Transforms, a Working Minimum

NEEDED FOR: Fourier analysis (background) and statistical inference (prerequisite; system stability fluency). Not needed for ML.

34.1 Theory

The Fourier transform evaluates a system's eigenvalue function $H(s)$ (Proposition 30.1) only on the axis $s = j\omega$. The Laplace transform keeps the whole complex plane — which is what lets it handle unstable systems, transients, and initial conditions.

Definition 34.1 (Laplace transform, ROC)**.** $X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt$ for those $s \in \mathbb{C}$ where the integral converges absolutely — the *region of convergence* (ROC), always a vertical strip determined by $\operatorname{Re} s$. On $s = j\omega$ (when the ROC contains it), $X(j\omega)$ is the Fourier transform.

Example 34.2 (Same formula, different signals)**.**

$$e^{-at}u(t) \;\longleftrightarrow\; \frac{1}{s+a}, \quad \operatorname{Re} s > -a; \qquad -e^{-at}u(-t) \;\longleftrightarrow\; \frac{1}{s+a}, \quad \operatorname{Re} s < -a.$$

(Both are the same integral computed over the half-line where the signal lives.) Identical algebra, different ROCs, completely different signals — one right-sided, one left-sided. *The ROC is part of the transform*; quoting $X(s)$ without it determines nothing.

Proposition 34.3 (ROC facts for rational transforms)**.** *Let $X(s)$ be rational with poles $p_1, \dots, p_r$ (roots of the denominator — Lesson 14's territory). Then:*

(a) the ROC is a vertical strip containing no pole, bounded by poles (or extending to $\pm\infty$);

(b) x right-sided (e.g. causal) $\iff$ ROC is the half-plane to the right of the rightmost pole;

(c) the system with impulse response x is BIBO stable $\iff$ the ROC contains the $j\omega$-axis;

(d) hence: causal and stable $\iff$ all poles satisfy $\operatorname{Re} p_i < 0$ *(left half-plane).*

Proof sketch. (a) absolute convergence depends only on $\operatorname{Re} s$, and it fails at a pole. (b) for a right-sided x, convergence at s_0 implies convergence for all $\operatorname{Re} s > \operatorname{Re} s_0$ (the factor $e^{-(s-s_0)t}$ only helps as $t \to +\infty$); Example 34.2 is the template. (c) stability is $\int |h| < \infty$ (Theorem 29.3), which is exactly absolute convergence of the transform at $s = j\omega$. (d) combine (b) and (c): the right-of-rightmost-pole half-plane contains the axis iff every pole is strictly to its left. The instance $h = e^{-at}u(t)$: pole at $-a$, causal always, stable iff $a > 0$ — matching the direct computation $\int_0^\infty e^{-at}dt < \infty$. □

Inversion by partial fractions. Rational transforms invert by table lookup after a partial-fraction expansion (Lesson 5's linear algebra: matching coefficients is solving a linear system). Worked once, causal case:

$$H(s) = \frac{1}{(s+1)(s+2)} = \frac{1}{s+1} - \frac{1}{s+2} \quad\Longrightarrow\quad h(t) = \big(e^{-t} - e^{-2t}\big)u(t),$$

with ROC $\operatorname{Re} s > -1$; poles at $-1, -2$ in the left half-plane: stable, as h's decay confirms. Differential equations transform to algebra the same way as in Lesson 32: $\sum_k a_k y^{(k)} = \sum_k b_k x^{(k)}$ gives $H(s) = \big(\sum b_k s^k\big)/\big(\sum a_k s^k\big)$, by the differentiation property $x' \leftrightarrow sX$ (same proof as Theorem 31.3(e)).

Definition 34.4 (z-transform)**.** The DT twin: $X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$ for $z \in \mathbb{C}$ in the ROC — always an annulus $r_1 < |z| < r_2$. On the unit circle $z = e^{j\omega}$ (when the ROC contains it), this is the DTFT — which is why Lesson 32 wrote $X(e^{j\omega})$.

The dictionary translates line by line ($e^{st} \to z^n$; vertical strips $\to$ annuli; left half-plane $\to$ inside the unit circle):

$$a^n u[n] \;\longleftrightarrow\; \frac{1}{1-az^{-1}}, \;\; |z| > |a|; \qquad \text{causal + stable} \iff \text{all poles satisfy } |p_i| < 1.$$

The first is Example 32.2's geometric series with $e^{j\omega}$ promoted to z; the second is Proposition 34.3(d) with the geometry renamed — and it finally unifies every "$|a| < 1$" in this booklet:

Lesson 10's decaying eigenvalues, Lesson 26's spectral gap, Lesson 29's stable recursion, Lesson 32's lowpass smoother. They are all "poles inside the unit circle."

Difference equations give rational $H(z) = \big(\sum_k b_k z^{-k}\big)/\big(\sum_k a_k z^{-k}\big)$ by the shift property $x[n-1] \leftrightarrow z^{-1}X(z)$, and $|H(e^{j\omega})|$ can be read geometrically from the pole–zero plot: magnitude response = (product of distances from $e^{j\omega}$ to zeros) / (product of distances to poles) — peaks near poles, notches near zeros (Exercise 32.5 built exactly such a notch).

34.2 Practice: by hand

Example 34.5 (A second-order system, end to end)**.** $H(z) = \dfrac{1}{(1-0.9z^{-1})(1+0.5z^{-1})}$, causal. Poles at 0.9 and -0.5: both inside the unit circle — stable. Partial fractions:

$$H(z) = \frac{9/14}{1-0.9z^{-1}} + \frac{5/14}{1+0.5z^{-1}} \implies h[n] = \Big(\tfrac{9}{14}(0.9)^n + \tfrac{5}{14}(-0.5)^n\Big)u[n]\,:$$

a slow decay (pole near the circle) plus a fast alternating one. Frequency response by geometry: the pole at 0.9 sits near $z = 1$, so $|H|$ peaks at $\omega = 0$; the pole at -0.5 mildly boosts $\omega = \pi$. The long-run behavior is owned by the pole of largest modulus — Lesson 10's dominant eigenvalue, verbatim.

Fourier / statistical-inference connection. Fourier analysis mostly stays on the $j\omega$-axis, but its treatment of one-sided signals and of the Laplace–Fourier relationship assumes this fluency; statistical inference uses pole locations as the standing language for stability of estimators and filters (a Kalman filter's error dynamics are stable because certain poles are inside the unit circle). If you can go from a difference equation to $H(z)$ to poles to a stability verdict to $h[n]$ without notes, this lesson has done its job.

34.3 Exercises

Exercise 34.1 (Hand)**.** Invert $X(s) = \dfrac{s+3}{(s+1)(s+2)}$ (causal): partial fractions, $x(t)$, stability verdict. Then re-answer for the ROC $\operatorname{Re} s < -2$ — which signal is it now, and is the Fourier transform defined?

Exercise 34.2 (Hand)**.** For $y[n] = 0.8y[n-1] + x[n] - x[n-1]$: find $H(z)$, its pole and zero, sketch $|H(e^{j\omega})|$ from the geometry (zero at $z = 1$!), and classify the filter (lowpass / highpass / notch). Compute $h[n]$ by partial fractions or long division.

Exercise 34.3 (Proof, $\star$)**.** Prove Proposition 34.3(c) in the DT case: the system with impulse response h is BIBO stable iff the ROC of $H(z)$ contains the unit circle. (Both directions are Theorem 29.3 plus the definition of absolute convergence at $|z| = 1$.)

Exercise 34.4 (Proof)**.** Prove the final-value flavor of Lesson 10 in transform language: if $H(z)$ is causal with all poles strictly inside the unit circle, then $h[n] \to 0$ geometrically, at rate $\max_i |p_i|$. (Partial fractions: each term is $c_i\, p_i^n u[n]$, possibly times a polynomial in n for repeated poles — bound it.)

Deeper reading. O&W 9.1–9.3, 9.5, 9.7 (Laplace, ROC, inversion, properties, LTI analysis), 10.1–10.5, 10.7 (the z-transform mirror); 9.4/10.4 (geometric evaluation from pole–zero plots) rewards the extra hour.

35 Capstone V: Filtering and Sampling in NumPy

NEEDED FOR: Fourier analysis and statistical inference

Forty-five minutes; Part V at once. A complete signal chain: synthesize, corrupt, sample safely, filter, and verify every step against this part's theorems. Type it.

```
import numpy as np
rng = np.random.default_rng(0)

# --- 1. A "continuous" scene: two tones + wideband noise ---
fs_fine = 8000.0
t = np.arange(0, 2, 1/fs_fine)
clean = np.sin(2*np.pi*40*t) + 0.6*np.sin(2*np.pi*300*t)
x = clean + 0.5*rng.standard_normal(t.size)

# --- 2. Sample at 1 kHz: Nyquist check first (Lesson 33) ---
M = 8                                   # 8000/8 = 1000 Hz
fs = fs_fine / M
# anti-alias with a moving average before decimating (Lesson 29):
w_ma = 9
xa = np.convolve(x, np.ones(w_ma)/w_ma, mode='same')
xs = xa[::M]
ns = np.arange(xs.size)

# --- 3. Design: keep 40 Hz, kill 300 Hz (Lessons 32, 34) ---
w0 = 2*np.pi*300/fs                     # notch target on the DT circle
h_notch = np.array([1., -2*np.cos(w0), 1.])
h_notch /= h_notch.sum()                # unit DC gain
y = np.convolve(xs, h_notch, mode='same')

# --- 4. Certify in frequency (Lessons 12, 31): FFT before/after ---
f = np.fft.rfftfreq(xs.size, 1/fs)
for name, sig in (("in", xs), ("out", y)):
    S = np.abs(np.fft.rfft(sig))
    print(name, [round(S[np.argmin(np.abs(f-ff))]) for ff in (40, 300)])
# -> 300 Hz amplitude collapses; 40 Hz survives

# --- 5. Convolution property, certified (Thm 32.2 / Lesson 12) ---
N = xs.size + h_notch.size - 1
lhs = np.fft.fft(np.convolve(xs, h_notch), N)
rhs = np.fft.fft(xs, N) * np.fft.fft(h_notch, N)
assert np.allclose(lhs, rhs, atol=1e-6)

# --- 6. Reconstruction (Lesson 33): sinc-interpolate a stretch ---
tt = np.arange(0.5, 0.55, 1/fs_fine)
xr = np.array([(y * np.sinc((ttt - ns/fs)*fs)).sum() for ttt in tt])
ref = np.sin(2*np.pi*40*tt)
print("recon err:", np.abs(xr - ref).std())    # small: mostly leftover noise
```

Checkpoints (do all of these).

1. Step 2's order matters: decimate *without* the anti-aliasing average (`xs = x[::M]`) and rerun step 4 — broadband noise from above 500 Hz folds into the band and every spectral floor rises (Lesson 33's aliasing, done to noise).

2. Step 3 is Exercise 32.5's notch: zeros at $e^{\pm j\omega_0}$, placed straight from Lesson 14's roots-of-a-polynomial picture. Verify $H(e^{j\omega_0}) \approx 0$ and $H(1) = 1$ by direct evaluation.

3. Step 5 fails without the zero-padding to length N — circular vs. linear convolution, Lessons 12 vs. 29. Break it on purpose and see the wrap-around error.

4. Step 6's reconstruction error is dominated by the noise the notch didn't remove, not by the sampling: halve the noise amplitude and watch the error halve, then move the 40 Hz tone to 480 Hz and watch sinc reconstruction start failing as Nyquist approaches.

5. Every operation in the chain was linear and (except the decimator) time-invariant — classify the decimator with Definition 28.5 (it is linear but *not* TI), which is why multirate DSP needs care beyond this booklet.

Final self-check list for Part V

From memory: classify a system's five properties with counterexamples; state and prove the convolution representation of LTI systems and the h-tests for causality and stability; hand-convolve short sequences; prove the eigenfunction property and the Fourier series coefficient formula from orthogonality; compute the square wave's coefficients and explain the smoothness–decay law and Gibbs; state the CTFT pairs and properties, prove the convolution property, and analyze an RC filter; define the DTFT, get $H(e^{j\omega})$ from a difference equation, and connect the DFT as its samples; prove spectrum replication and the sampling theorem, and do alias arithmetic; use ROCs, partial fractions, and pole locations to decide causality and stability in s and z. That is O&W Chapters 1–7 and 9–10 at working strength — the linear-systems layer — and with it, every undergraduate prerequisite for machine learning, Fourier analysis, and statistical inference in this booklet is in place.

Part VI: Applied Signal Processing — DSP, Communications, Images, Audio, and the Bench

Parts I–V built the mathematics; this part spends it, and its scope rule is strict: *if no Course 2 lab touches a topic, it is not here.* It abridges the Course 1 (diiv.io) Phase 4–6 booklets down to the design recipes, the formulas an engineer actually reaches for, and one worked example per idea, citing the full booklets (*DiivCourse1DSPMathFoundations*) for every derivation it drops. The sources it abridges are the real ones: Lyons, *Understanding Digital Signal Processing*; Kuo–Lee, *Real-Time Digital Signal Processing*; Hayes, *Statistical Digital Signal Processing and Modeling*; Grewal–Andrews, *Kalman Filtering*; Richards, *Fundamentals of Radar Signal Processing*; Proakis–Salehi, *Digital Communications*; Gonzalez–Woods, *Digital Image Processing* (4th ed.); Zölzer, *Digital Audio Signal Processing* (3rd ed.); and Scherz–Monk, *Practical Electronics for Inventors* (4th ed.).

Each of the nine lessons ends by naming the Course 2 labs it unlocks: the bench modules (Lesson 44), converters and aliasing (38), real-time FIR/IIR filters and FFT analyzers (36–37), noise floors, Kalman filters, and adaptive equalizers (39–40), radar detection and CFAR (41), and the audio and image pipelines of the edge-ML modules (42–43). The same nine lessons double as the abridged on-ramp to the applied subjects: 36–39 for digital signal processing, 39–41 for digital communications, 42 for image processing and the floor under computer vision. Prerequisites are Parts II–V — especially Lessons 12 (DFT), 22–23 (LMS estimation, Gaussian vectors), 29 (convolution), and 32–34 (DTFT, sampling, z-transforms). Part VII, the rigor layer, certifies this part's analysis moves and can be read before, after, or never.

There is no separate capstone: the Course 2 labs *are* the capstone.

36 The DFT in Practice: Resolution, Leakage, Windows, and the FFT

NEEDED FOR: Digital signal processing; Course 2 Labs 6.3, 6.5

Lesson 12 built the DFT as a unitary change of basis; Lesson 32 identified it as the DTFT of a finite record sampled at N equally spaced frequencies. This lesson is the practical layer: bins in hertz, leakage, windows, and the FFT and Goertzel algorithms.

36.1 Theory

Bins, resolution, and amplitude calibration. For a record $x[n]$, $n = 0, \dots, N-1$, sampled at rate f_s, the DFT

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi nk/N}, \qquad k = 0, 1, \dots, N-1,$$

correlates the data against a sinusoid with exactly k complete cycles over the N samples. Bin k therefore answers for the analysis frequency $f_k = kf_s/N$: the bin spacing is f_s/N, $k = N/2$ sits at the folding frequency $f_s/2$, and $X[0]$ is the sum of the samples (N times the mean). For real input, $X[N-k] = X[k]^*$ (Lesson 12's conjugate symmetry), so only the $N/2+1$ bins $k = 0, \dots, N/2$ are independent. Amplitudes scale with N: a bin-centered real sinusoid of peak

A_0 produces $|X[k]| = A_0N/2$ (a complex sinusoid gives A_0N; a DC bias D_0 gives $X[0] = D_0N$): read amplitudes by dividing by $N/2$ and the window's loss factor, or plot the normalized dB spectrum $20\log_{10}(|X[k]|/|X|_{\max})$, where all scale factors cancel. Resolution is bought only with record length:

$$\text{bin spacing } \tfrac{f_s}{N} = F_{\text{res}} \iff N = \tfrac{f_s}{F_{\text{res}}} \iff 1/F_{\text{res}} \text{ seconds of data.}$$

Leakage. The DFT is exact only for tones sitting precisely at bin centers. A real sinusoid of amplitude A_0 completing a *non-integral* k_0 cycles over the record spreads across every bin, with approximate bin response

$$X[k] \;\approx\; \frac{A_0N}{2}\,\frac{\sin\big[\pi(k_0-k)\big]}{\pi(k_0-k)}.$$

This sinc's zero crossings land exactly on the other bins when k_0 is an integer (no leakage) and on none of them otherwise (leakage everywhere). The shape is the spectrum of the implicit *rectangular window*: any finite-length DFT has multiplied the signal by a box, whose transform (the Dirichlet kernel) has its first sidelobe only -13 dB down. The spectrum is N-periodic (bin -1 is bin $N-1$), so leakage also *wraps around* $k = 0$ and $k = N/2$, skewing the skirts.

Example 36.1. $k_0 = 3.4$, $N = 64$, unit amplitude: the formula predicts

$$|X[3]| \approx 32\,\frac{\sin(0.4\pi)}{0.4\pi} = 24.2;$$

the computed DFT gives 25.1 — the excess is wraparound leakage.

Windows. Windowing tapers the record's ends smoothly toward a common value before transforming, trading main-lobe width for sidelobe suppression:

Window	Main lobe	Sidelobes
Rectangular	narrowest	first at -13 dB; strong leakage
Triangular	double width	lower than rectangular
Hanning $\frac{1}{2} - \frac{1}{2}\cos\frac{2\pi n}{N}$	double width	lower still; *fast* roll-off
Hamming	double width	lowest *first* sidelobe; slow roll-off

So Hanning wins far from a strong tone, Hamming close in (the adjustable Kaiser window tunes the trade continuously). The payoff is *detection*: in Lyons's two-tone demo a 0.1-amplitude tone two bins from a full-scale leaky tone is invisible under the rectangular window and unmistakable under Hanning. Related is *scalloping*: between bin centers the unwindowed DFT's response dips to 0.637 (-3.92 dB, the picket-fence ripple); Hanning's wider lobes fill the valleys to about -1.45 dB.

Zero padding is not resolution. Appending zeros to L data samples and taking a longer N-point DFT samples the record's DTFT (Lesson 32) on the finer grid kf_s/N: it interpolates the spectrum's *display* and cures scalloping, but buys *zero* resolution — the main lobe's width in Hz is set by the L nonzero samples alone (the sinc's nulls refuse to move). Two traps: the $A_0N/2$ amplitude rules use L, not N, under padding; and the window must be applied to the L data samples only, never across the appended zeros. The field checklist, in order: sample at

2.5–4× the signal bandwidth and mistrust components near $f_s/2$; subtract the record's mean (else the DC bin's window-smeared skirt buries low-frequency neighbors); *then* window; *then* zero-pad up to a power of two (never discard data); normalize and plot in dB. A DFT bin is also a bandpass filter of width $\sim f_s/N$: doubling N buys 3 dB of output SNR for a bin-centered tone in white noise ($10\log_{10}$ of the length ratio — *processing gain*); averaging multiple DFTs (Welch) buys further integration gain.

The radix-2 FFT. The Cooley–Tukey FFT is *not an approximation* — it computes exactly the DFT above, merely eliminating redundant arithmetic. Write the DFT with the twiddle factor $W_N = e^{-j2\pi/N}$ and split $x[n]$ into even- and odd-indexed halves; because $W_N^2 = W_{N/2}$, each half is an $N/2$-point DFT — $A[k]$ (even samples) and $B[k]$ (odd) — and

$$X[k] = A[k] + W_N^k\, B[k], \qquad X\big[k+\tfrac{N}{2}\big] = A[k] - W_N^k\, B[k], \qquad k = 0, \ldots, \tfrac{N}{2}-1,$$

since $W_N^{k+N/2} = -W_N^k$: the top half of the spectrum costs only sign flips. Recursing down to 2-point "butterflies" gives $\log_2 N$ stages of $N/2$ one-multiply butterflies, hence about

$$\frac{N}{2}\log_2 N \quad \text{complex multiplies versus the direct DFT's } N^2,$$

under the constraint $N = 2^k$. At $N = 512$ that is 114× fewer; at $N = 8192$, 1260×. The even/odd (decimation-in-time) splitting scrambles the input into *bit-reversed* order ($N = 8$: $0, 4, 2, 6, 1, 5, 3, 7$); twiddles are precomputed into tables, and fixed-point implementations scale between stages (block floating point) because a butterfly can more than double magnitudes.

The Goertzel algorithm: one bin, cheap. When only a few bins matter — DTMF tone detection needs eight — a full FFT is waste. Since $W_N^{-kN} = 1$, the DFT sum can be read as a filter: $X[k]$ is the output at time $N-1$ of the single-pole system $H_k(z) = 1/(1 - W_N^{-k} z^{-1})$, pole *on* the unit circle at the bin frequency. Multiplying through by $(1 - W_N^k z^{-1})$ pairs the conjugate poles into an all-real recursion: for $n = 0, \ldots, N-1$,

$$w_k[n] = x[n] + 2\cos\!\big(\tfrac{2\pi k}{N}\big)\, w_k[n-1] - w_k[n-2],$$

then once, at the frame's end,

$$|X[k]|^2 = w_k^2[N-1] + w_k^2[N-2] - 2\cos\!\big(\tfrac{2\pi k}{N}\big)\, w_k[N-1]\, w_k[N-2]$$

— one real coefficient and one multiply-add per sample per tone, no complex arithmetic when only power is needed. Choose N so each target frequency lands near a bin, $f_k/f_s \approx k/N$ (telephony: $f_s = 8$ kHz, $N = 205$). The on-circle pole is safe because the recursion runs a finite N samples and restarts each frame.

Applications / Course 2. This lesson is DSP's bread and butter. Course 2's Lab 6.3 (FFT spectrum analyzer) is the checklist verbatim: CMSIS-DSP's `arm_rfft_q15` is a fixed-point, bit-reversed, table-twiddled radix-2 engine, and the lab's "why is my amplitude half what I expected?" moment is the $N/2$ factor. The same FFT engine powers Lab 6.4. Lab 6.5's DTMF detector is the Goertzel recursion — one MAC per sample per tone — plus validation tests (twist, talk-off) on the raw bin powers.

36.2 Practice: by hand

P1. A 1024-point DFT of real data is taken at $f_s = 25$ kHz. Give the bin spacing, the frequency of bin $k = 300$, the number of independent bins, and the bin holding 10 kHz energy.

P2. For $N = 1024$: how many complex multiplies does the direct DFT need, how many does the radix-2 FFT need, and what is the ratio?

P3. For $f_s = 8$ kHz, $N = 205$, compute the Goertzel bin $k = \text{round}(f_0 N/f_s)$ and bin frequency kf_s/N for each DTMF low-group tone (697, 770, 852, 941 Hz); check each error against the $\pm 1.5\%$ accept spec.

36.3 Exercises

Exercise 36.1 ([Hand)**.**] (Lyons 3.2) A systems engineer specifies a DFT spectrum analyzer with input sample rate $f_s = 1000$ Hz and frequency-domain sample spacing of exactly 45 Hz. (a) How many input samples N does a single DFT need? (b) What do you tell the systems engineer about the specification?

Exercise 36.2 ([Hand)**.**] (Lyons 4.3) A 3800-sample sequence $x[n]$ represents 2 seconds of collected signal. (a) How many zero-valued samples must be appended to implement a radix-2 FFT? (b) After the FFT, what is the spacing in Hz between frequency-domain samples? (c) For lowpass sampling, what is the highest analog frequency permitted in $x(t)$ without aliasing?

Exercise 36.3 ([Hand)**.**] (Lyons 3.20) A SETI search collects 10^6 time samples and performs a 10^6-point DFT, looking for spectral components that exceed the background noise. Roughly what processing-gain improvement, in dB, does this give over a 100-point DFT in pulling a weak tone above the galactic noise?

Deeper reading. Lyons Chs. 3–4 are this lesson at full length (3.13 derives the Dirichlet kernel; 4.4–4.6 cover bit reversal and butterfly structures); the definitive window catalog is harris's 1978 *Proc. IEEE* survey. Kuo 9.3 develops Goertzel-based DTMF detection through the six validation tests of a compliant decoder; Lyons 13.17 adds threshold-setting advice; Lyons 13.10 covers FFT convolution. The full treatment condensed here is the Phase 4 booklet, Lessons 6, 7, and 9. For the rigorous account of the same material, Oppenheim–Schafer–Buck, *Discrete-Time Signal Processing* (2nd ed.), Chs. 8–10: the DFT's algebra (8), Goertzel and the FFT factorizations (9, Goertzel in 9.2), and exactly this lesson's leakage/window/spectrogram analysis (10).

37 Digital Filters: FIR and IIR Design, and Finite Word Lengths

NEEDED FOR: Digital signal processing; Course 2 Labs 6.1 (FIR on real ADC data) and 6.2 (IIR biquad-cascade stability and coefficient quantization).

37.1 Theory

FIR filters and linear phase. A *finite impulse response* (FIR) filter computes each output as a weighted sum of the current and past $N-1$ inputs,

$$y[n] = \sum_{k=0}^{N-1} h[k]\, x[n-k],$$

a *tapped delay line* of unit delays, multipliers $h[k]$, and one adder. Feeding $\delta[n]$ shows the coefficients *are* the impulse response — finite by construction, so FIR filters are always BIBO stable. This is Lesson 29's convolution, and by Lesson 32 the filter's action on any input is multiplication by the frequency response: $Y(e^{j\omega}) = H(e^{j\omega})\, X(e^{j\omega})$.

Theorem 37.1 (Linear phase of symmetric FIR filters)**.** *If $h[k] = h[N-1-k]$, the frequency response factors as a real amplitude function times a pure delay,*

$$H(e^{j\omega}) = A(\omega)\, e^{-j\omega(N-1)/2},$$

so every frequency is delayed by the same group delay *of $(N-1)/2$ samples — no phase distortion, the marquee virtue of FIR filters. (Pair terms symmetrically and use Euler; proof in the Phase 4 booklet, Lesson 3. Antisymmetric coefficients also give linear phase plus a 90° offset.)*

Daily-use facts: the DC gain is the coefficient sum $H(e^{j0}) = \sum_k h[k]$; an N-tap filter costs N multiplies per output (halved by folded symmetric structures); transition width scales like $1/N$ (harris's rule: $N \approx \text{atten}_{\text{dB}}/[22\,\Delta f/f_s]$).

Window design method (recipe). To design a lowpass FIR filter with cutoff f_c (as a fraction of f_s):

1. The ideal brick-wall response has the infinite sinc impulse response $h_\infty[k] = 2f_c\,\text{sinc}(2f_c k)$, k measured from the center tap ($\text{sinc}(x) = \sin(\pi x)/(\pi x)$).
2. Truncate to N taps centered on $k = 0$. Plain truncation is a rectangular window: its spectrum's large sidelobes put ripple in the passband and slow stopband decay — the Gibbs phenomenon as an engineering defect.
3. Multiply by a tapered window (Hamming, Blackman, ...), trading *lower sidelobes for a wider transition band*; more taps buy the width back. Optionally rescale so $\sum_k h[k] = 1$ (unity DC gain).

Variations: multiplying a lowpass prototype by a cosine at f_0 shifts it to bandpass; multiplying by $(-1)^k$ makes it highpass. The optimal alternative is the *Parks–McClellan* (Remez exchange) equiripple design — specify band edges and ripple weights, iterate to the unique equioscillating solution.

IIR filters and stability. An *infinite impulse response* (IIR) filter feeds outputs back:

$$y[n] = \sum_{k=0}^{M} b_k\, x[n-k] - \sum_{k=1}^{N} a_k\, y[n-k] \quad \Longleftrightarrow \quad H(z) = \frac{\sum_k b_k z^{-k}}{1 + \sum_k a_k z^{-k}},$$

by Lesson 34. A few coefficients buy a sharp response, at the price of possible instability and nonlinear phase. The frequency response is $H(z)$ on the unit circle, $H(e^{j\omega})$.

Theorem 37.2 (Stability from pole locations)**.** *A causal IIR filter is BIBO stable if and only if all poles of $H(z)$ lie strictly inside the unit circle (the ROC then contains the circle, Lesson 34). A pole on the circle gives an oscillator (marginally stable); outside, growth without bound. Zeros may lie anywhere; a zero on the unit circle forces an exact null at that frequency.*

Design by bilinear transform (recipe). Classic IIR design transplants an analog prototype $H_a(s)$ (Butterworth, Chebyshev, elliptic) via the substitution

$$s = \frac{2}{t_s}\,\frac{1 - z^{-1}}{1 + z^{-1}},$$

a one-to-one map of the whole $j\omega$ axis onto one trip around the unit circle — no aliasing ever (unlike impulse invariance, which periodizes the analog response) — at the price of *frequency warping*:

$$\omega_a = \frac{2}{t_s}\tan\!\Big(\frac{\omega_d}{2}\Big).$$

Procedure: (1) *prewarp* each critical digital frequency through the tangent; (2) design the analog prototype at the warped frequencies; (3) substitute for s. The digital filter then hits the intended band edges exactly.

Biquads and cascading. Direct Form I, Direct Form II, and their transposes are identical in infinite precision. In finite precision they differ sharply, and the standing practice is to factor a high-order $H(z)$ into *cascaded second-order sections* (biquads),

$$H(z) = \prod_i \frac{b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}},$$

so that coefficient quantization perturbs each conjugate pole pair independently instead of scrambling all roots of one high-order polynomial — Wilkinson's root-sensitivity phenomenon (Part VII, Lesson 51) is exactly the villain. The DC gain of a biquad is $H(z)$ at $z = 1$.

Q-formats. Fixed-point words are two's complement with an implied binary point: an "$m.n$" word has m integer and n fraction bits, a pure programmer's convention — the hardware adds and multiplies the same bit patterns regardless. The workhorse is the all-fraction *1.15 format, alias Q15*:

$$x = -b_0 + \sum_{m=1}^{15} b_m 2^{-m}, \qquad -1 \le x \le 1 - 2^{-15}.$$

Multiplying two Q15 numbers gives a Q30 product that is again a fraction — multiplication can never overflow in Q15, which is why DSP chips adopted it. To convert a decimal coefficient: multiply by $2^{15} = 32768$, round, render in binary/hex (e.g. $0.625 \to 20480 = 0\text{x}5000$).

Quantization noise model. An ideal b-bit converter spanning $\pm V_p$ has quantization level $q = 2V_p/2^b$, and its error is modeled as uniform on $[-q/2, q/2]$, stationary and uncorrelated with the input: mean zero and variance

$$\sigma_e^2 = \frac{q^2}{12} = \frac{2^{-2b}}{3}\,V_p^2.$$

With loading factor $LF = \sigma_x/V_p$, the output signal-to-noise ratio is $\mathrm{SNR} = 6.02\,b + 4.77 + 20\log_{10}(LF)$ dB; a full-scale sinusoid ($LF = 1/\sqrt{2}$) gives the marquee number

$$\mathrm{SNR}_{\max} = 6.02\,b + 1.76 \text{ dB}$$

— about 49.9 dB for 8 bits, 98 dB for 16. Internal arithmetic obeys the same model: rounding a result to level q injects zero-mean noise of variance $q^2/12$, while *truncation* adds a DC bias of $-q/2$ — poison in feedback loops. Standing rule: accumulate at full width (Q30) and round *once* at the end, not per tap.

Coefficient quantization. Quantizing designed coefficients to B bits moves the realized poles and zeros. Poles clustered near the unit circle suffer most — a quantized pole can land on or outside the circle and destabilize the filter — and the sensitivity grows with direct-form order, which is why the biquad-cascade practice above exists.

Overflow and saturation. The sum of m b-bit words can need $b+\log_2 m$ bits, and a feedback accumulator must hold the DC gain $G = H(1)$ times the input — width grows by $\lceil \log_2 G \rceil$ bits. Two's complement wraparound is forgiving: intermediate overflows are harmless if the final sum fits the word. *Saturation* arithmetic clamps to ±full scale instead of wrapping — a clipping nonlinearity that creates harmonics, so it is a guardrail, not a design. The real fix is *scaling*: pre-scale the input by $\beta < 1$ so no internal node overflows (for an FIR filter $|y[n]| \le \beta M_x \sum_l |b_l|$), at a cost of $20\log_{10}\beta$ dB of SNR — about one bit per halving.

DSP / Course 2 connection. This lesson is DSP's filter-design core and the numerical contract of Course 2's firmware. Lab 6.1 runs the windowed-sinc recipe on real ADC data and measures the $(N-1)/2$ group delay; Lab 6.2 builds the bilinear-designed biquad cascade (CMSIS-DSP's `arm_biquad_cascade`, coefficients in Kuo's Q15 as `q15_t`) and checks Theorem 37.2 *after* coefficient quantization. Every ADC datasheet's SNR spec is the $6.02b+1.76$ ideal.

37.2 Practice: by hand

Example 37.3 (A 5-tap windowed-sinc lowpass, $f_c = 0.2f_s$)**.** $h_\infty[k] = 0.4\,\mathrm{sinc}(0.4k)$ gives, for $k = -2, \ldots, 2$,

$$h = [\,0.0935,\ 0.3027,\ 0.4,\ 0.3027,\ 0.0935\,]$$

(rectangular window). Symmetric, so linear phase with group delay $(5-1)/2 = 2$ samples. DC gain $= \sum_k h[k] = 1.192$; dividing every tap by 1.192 normalizes it to one. Multiplying by $(-1)^k$ instead would make it a highpass with cutoff $0.3f_s$.

Example 37.4 (Bilinear transform of the RC lowpass)**.** Want a digital −3 dB point at exactly $0.1f_s$, with $t_s = 1$. Prewarp: $\omega_c = 2\tan(\pi \cdot 0.1) = 0.6498$. Prototype $H_a(s) = \omega_c/(s+\omega_c)$; substituting $s = 2(1-z^{-1})/(1+z^{-1})$ gives

$$H(z) = \frac{0.2452\,(1+z^{-1})}{1-0.5095\,z^{-1}}:$$

one pole at $z = 0.5095$ (inside the circle: stable), one zero at $z = -1$ (exact null at $f_s/2$), DC gain $2(0.2452)/(1-0.5095) = 1$ exactly.

Example 37.5 (Q15 and converter SNR)**.** $1/\sqrt{2} \approx 0.70711$: $0.70711 \times 32768 = 23170.5 \to 23170 = 0\text{x}5\text{A}82$, representation error about half an lsb. A 12-bit converter spanning ± 2.5 V has $q = 5/4096 = 1.22$ mV, $\sigma_e = q/\sqrt{12} = 0.35$ mV, and full-scale-sinusoid SNR $= 6.02(12) + 1.76 = 74.0$ dB.

37.3 Exercises

Exercise 37.1 ([Hand] Lyons 5.14)**.** A linear-phase lowpass FIR filter has coefficients $h_1[k] = [-0.8,\ 1.6,\ 25.5,\ 47,\ 25.5,\ 1.6,\ -0.8]$ and DC gain $H_1(e^{j0}) = 99.6$. Changing the coefficients to $h_2[k] = [-0.8,\ 1.6,\ Q,\ 47,\ Q,\ 1.6,\ -0.8]$ gives a new DC gain of 103.6. What is Q?

Exercise 37.2 ([Hand] Lyons 6.22)**.** Knowing a filter's DC gain is critical when implementing it in binary arithmetic (it sizes the accumulator). For the standard biquad $H(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2})/(1 + a_1 z^{-1} + a_2 z^{-2})$, what is the DC gain in terms of the coefficients?

Exercise 37.3 ([Hand] Lyons 12.22)**.** An ideal 12-bit A/D converter operates over ± 5 volts. (a) What is its quantization level q? (b) What are its maximum positive and negative quantization errors? (c) For a 7-volt peak–peak sinusoidal input, what output signal-to-quantization-noise ratio $\text{SNR}_{A/D}$ in dB should we expect? Show your work.

Deeper reading. The Phase 4 booklet, Lessons 3–5, carries the full treatment this lesson abridges: Parks–McClellan and half-band FIR design, impulse-invariance worked examples (watch aliasing corrupt a design), Direct Form structures, and the floating-point/block-floating-point formats cut here — sourced from Lyons Chs. 5, 6, 12 and Kuo Chs. 3, 5, 6. Oppenheim–Schafer–Buck (2nd ed.) is the theory underneath: filter structures and the coefficient-quantization and round-off-noise analyses this lesson could only summarize (Ch. 6, especially 6.6–6.7, plus limit cycles in 6.8), and the complete design theory — impulse invariance, bilinear, Kaiser windows, optimum equiripple — in Ch. 7.

38 Multirate Systems, Oversampling, and Sigma–Delta Conversion

NEEDED FOR: Digital signal processing; Course 2 Labs 3.2–3.5, 5.4

38.1 Theory

Real systems rarely agree on one sample rate: CD audio lives at 44.1 kHz, professional audio and USB interfaces at 48 kHz, and every conversion between them must happen *digitally*, without returning to analog. The ratio is the awkward rational number $48/44.1 = 160/147$, so "resampling" cannot mean "keep every few samples." This lesson builds the machinery: changing sample rate by integers (M down, L up), by rationals L/M, doing it cheaply (polyphase), and then running the same ideas in reverse order inside modern AD/DA converters (oversampling and sigma–delta), where they buy resolution instead of rate.

Decimation: filter, then downsample. *Downsampling* by integer M keeps every Mth sample, $x_{\text{new}}[m] = x_{\text{old}}[Mm]$, so $f_{s,\text{new}} = f_{s,\text{old}}/M$. By the sampling picture of Lesson 33, the spectrum's periodic replications are now only $f_{s,\text{new}}$ apart: unless the signal band B satisfies $f_{s,\text{new}} > 2B$, the replications overlap and alias. *Decimation* is therefore the two-step recipe *lowpass filter first, then downsample*. To preserve a band of width B', the filter (a Lesson 37 lowpass) must have its stopband start no later than

$$f_{\text{stop}} = f_{s,\text{new}} - B',$$

so that spectral folding lands nothing inside the kept band. Three facts to internalize: downsampling is *not* time-invariant (shift the input by one sample and a different subsequence emerges); it changes no time-domain amplitudes; but DFT magnitudes drop by M (fewer samples in the transform). For large M, two-stage decimation $M = M_1 M_2$ with the biggest factor first slashes filter cost — Lyons's $M = 100$ example drops from about 2727 taps in one stage to about 250 in two ($M_1 = 25$, $M_2 = 4$); a passband-ripple budget R is split $R/2$ per cascaded filter.

Interpolation: upsample, then filter. *Interpolation* by integer L is the dual recipe: *upsample* by inserting $L - 1$ zeros between samples ("zero stuffing"), *then* lowpass filter. Upsampling raises the rate to $Lf_{s,\text{old}}$ and exposes the old spectral replications as *images* centered at multiples of $f_{s,\text{old}}$; the interpolation filter's whole job is to remove them, so the interpolation's fidelity *is* the filter's stopband attenuation. Two standing facts: zero stuffing loses a factor L of amplitude, so the filter needs DC gain L (equivalently $\sum_k h[k] = L$); and repeating samples instead of stuffing zeros rolls the passband off by $\sin(x)/x$ — avoid it.

Rational conversion by L/M. Cascade interpolate-by-L then decimate-by-M, sharing *one* combined lowpass that serves both as anti-image and anti-alias filter; its passband must end at $\min\big(f_{s,\text{old}}/2,\ (f_{s,\text{old}}/2)(L/M)\big)$ (in radian terms, cutoff $\min(\pi/L, \pi/M)$ at the high internal rate). The motivating example: CD 44.1 kHz to DAT 48 kHz is $L/M = 160/147$; to 96 kHz it is $320/147$.

The polyphase idea. The naive cascades are wasteful twice over: zero stuffing makes most filter products zero, and downsampling discards most filter outputs. *Polyphase decomposition* eliminates both wastes. Split an N-tap prototype $h[k]$ by index residue mod Q:

$$H(z) = H_0(z^Q) + z^{-1}H_1(z^Q) + \cdots + z^{-(Q-1)}H_{Q-1}(z^Q),$$

giving Q subfilters of N/Q taps each. The *noble identities* — downsampling commutes with $H(z^Q) \to H(z)$, and dually for upsampling — move the rate change to the cheap side of the subfilters, and a commutating switch replaces the delay-and-resample scaffolding. The payoff: for decimation by M, the switch deals M consecutive inputs to the M subfilters, whose summed outputs give one result — nothing computed is discarded and *all arithmetic runs at the low rate*, $1/M$ the multiplies per unit time; for interpolation by L, each input fires the L subfilters in rotation with no multiply-by-zero. Rational L/M resampling reduces to a modulo-L counter selecting which subfilter (coefficient set) produces each output. We state the structure without the block-diagram derivation; the phase-4 booklet (Lesson 8 there) draws it in full.

Oversampling converters. Now run multirate thinking at the analog boundary. Plain Nyquist-rate conversion samples at $f_S = 2f_B$ with a w-bit converter and demands a brutal analog brick-wall anti-aliasing filter before the ADC. *Oversampling* by L instead spreads the fixed quantization-noise power $Q^2/12$ over an L-times wider band, so the noise *in the signal band* drops:

$$\mathrm{SNR} = 6.02\,w + 10\log_{10} L \quad [\mathrm{dB}],$$

i.e. 3 dB — half a bit — per doubling of L. Just as valuable: the analog anti-alias filter's transition band stretches from f_B out to nearly Lf_S, so it can be gentle and cheap, with a sharp *digital* lowpass and $\downarrow L$ decimation (this lesson's first half) doing the real work. DA conversion mirrors the chain: $\uparrow L$ zero-stuffing, a digital image-killing filter, the DAC running at Lf_S, and a relaxed analog reconstruction filter.

Noise shaping and sigma–delta. Three dB per octave is slow. A *sigma–delta* (delta–sigma) modulator wraps a feedback loop around a coarse quantizer — an integrator ahead of the quantizer, the quantized output fed back — so that

$$Y(z) \approx X(z) + \frac{1}{1+H(z)}\,E(z)\,:$$

the signal passes through, but the quantization error E is *shaped* away from DC. The first-order discrete model is $y[n] = x[n{-}1] + e[n] - e[n{-}1]$: error weighted by $|1 - e^{-j\Omega}|^2 = 4\sin^2(\Omega/2)$, small in-band, large near $f_S/2$ where the decimation filter removes it. In-band noise power becomes

$$N_B^2 \simeq \frac{Q^2}{12}\,\frac{\pi^2}{3}\Big(\frac{1}{L}\Big)^3$$

— 9 dB (1.5 bits) per doubling of L. A second-order loop shapes by $(1 - z^{-1})^2$ and gains 15 dB (2.5 bits) per doubling; multistage (MASH) cascades of first-order loops reach third-order shaping while staying unconditionally stable. This is how a *1-bit* quantizer clocked at $64f_S$ delivers 16–20-bit audio: trade amplitude resolution for rate, then filter the shaped noise away with exactly the decimation machinery above. The 3/9/15 dB-per-octave ladder (plain oversampling / first / second order) is the single most useful number set in converter selection.

DSP and Course 2. DSP's multirate weeks are this lesson; the labs put real silicon behind it. The converters on the Course 2 bench — the MCP4725 DAC and ADS1115 ADC of Labs 3.2–3.5 and the STM32's on-chip ADC — are oversampling converters: the ADS1115 is a delta-sigma part whose data-rate register trades L for effective bits exactly per the 3/9/15 dB ladder, and the STM32's hardware-oversampling registers literally implement $\downarrow L$ accumulation. The anti-alias and reconstruction filters flanking them are Lab 4.4's Sallen–Key stages — gentle analog filters that oversampling makes sufficient — and Lab 5.4's aliasing validation is what happens when the decimation recipe's "filter first" step is skipped.

38.2 Practice: by hand

P1 (decimation design). A 48 kHz stream must drop to 8 kHz keeping the band $B' = 3.2$ kHz. Then $M = 6$; the filter passes 0–3.2 kHz and its stopband must start by $f_{\mathrm{stop}} = 8 - 3.2 = 4.8$ kHz; after decimation the spectral replications sit 8 kHz apart.

P2 (polyphase bookkeeping). Decompose a 9-tap $h[k]$ for decimation by 3: $H_0 = \{h_0, h_3, h_6\}$, $H_1 = \{h_1, h_4, h_7\}$, $H_2 = \{h_2, h_5, h_8\}$. Naive filter-then-downsample runs 9 multiplies per *input* sample (27 per surviving output); polyphase runs 3 subfilters of 3 taps once per *output*: 9 multiplies, all at the low rate — the promised factor-M saving.

P3 (why noise shaping). To gain 8 bits (48 dB) beyond a converter's native resolution: plain oversampling at 3 dB/octave needs 16 octaves, $L = 2^{16} = 65536$ — hopeless. First-order shaping at 9 dB/octave needs $48/9 \approx 5.3$ octaves, $L \approx 40$; second-order at 15 dB/octave needs 3.2 octaves, $L \approx 9$. Noise shaping is what makes oversampled converters practical.

38.3 Exercises

Exercise 38.1 ([Hand)**.**] (Lyons 10.1, 10.8.) (a) To decimate $x[n]$ by four, should the lowpass filter act before or after discarding samples? Give the ideal filter's cutoff in Hz (in terms of the input rate f_s) and in radians/sample, and its required DC gain. (b) Answer the same three questions for interpolation by three: does the zero-stuffing happen before or after the filter, where is the cutoff, and what DC gain avoids amplitude loss?

Exercise 38.2 ([Hand)**.**] (Lyons 10.14.) A CD signal at $f_{s,\mathrm{CD}} = 44.1$ kHz must become a DAT signal at 48 kHz. If we interpolate by $L = 160$, by what factor M must we decimate to land exactly on 48 kHz?

Exercise 38.3 ([Hand)**.**] (Zölzer 3.1 practice.) (a) A 12-bit SAR ADC is oversampled by $L = 16$: compute the SNR from $\mathrm{SNR} = 6.02\,w + 10\log_{10} L$ and the number of effective bits. (b) For a 1-bit first-order sigma–delta modulator (levels ± 1, so $Q = 2$) at $L = 64$, compute the in-band noise power N_B^2 from the formula in this lesson and the SNR for a full-scale sine ($\sigma_X^2 = 1/2$).

Deeper reading. The phase-4 booklet's multirate lesson carries what we cut: the two-stage decimation design workflow, half-band filters, and the multiplier-free CIC (cascaded integrator–comb) filters that digest sigma–delta bit streams in hardware; its sources are Lyons Ch. 10, Crochiere & Rabiner's tutorial, and harris's *Multirate Signal Processing for Communication Systems*. The phase-6 booklet's AD/DA lesson derives the MASH cascade and surveys converter architectures (SAR, flash, R-2R, sigma–delta) — the vocabulary of every converter datasheet — and its sample-rate-conversion lesson treats the harder *asynchronous* case of free-running clocks (Zölzer Chs. 3 and 8). The theory chapter behind all of it is Oppenheim–Schafer–Buck (2nd ed.) Ch. 4's second half: rate changing and polyphase implementations (4.6–4.7) and the oversampling and noise-shaping A/D–D/A analysis (4.8–4.9) that makes $\Sigma\Delta$'s bits-for-bandwidth trade a computation rather than a slogan.

39 Discrete-Time Random Signals and Spectrum Estimation

NEEDED FOR: DSP and communications (noise models), statistical-inference overlap; Course 2 Lab 6.4 (noise floor and PSD estimation)

39.1 Theory

Random processes and stationarity. A discrete-time random process is an indexed family of random variables $x[n]$: each finite collection of samples is just a random vector (Part IV). The process is *Gaussian* if every such vector is jointly Gaussian (Lesson 23) — then means and covariances tell the whole story. Full "strict-sense" stationarity (all joint densities shift-invariant) is more than we can ever check; the workhorse assumption fixes only the first two moments.

Definition 39.1 (Wide-sense stationary)**.** $x[n]$ is *WSS* if (1) the mean $m_x = \mathbf{E}[x[n]]$ is constant in n, (2) the autocorrelation $r_x[k] = \mathbf{E}\big[x[n+k]\,x^*[n]\big]$ depends only on the lag k, and (3) the variance is finite. For Gaussian processes WSS is equivalent to strict-sense stationarity.

Autocorrelation properties. Three quick facts: *conjugate symmetry* $r_x[-k] = r_x^*[k]$; *mean-square value* $r_x[0] = \mathbf{E}[|x[n]|^2] \geq 0$; and *maximum at zero*, $|r_x[k]| \leq r_x[0]$: expand $\mathbf{E}[|x[n+k] - a\,x[n]|^2] \geq 0$ with $a = e^{j \arg r_x[k]}$. Stacking $p+1$ consecutive samples gives Lesson 23's covariance matrix $\mathbf{R}_x = \mathbf{E}[\mathbf{x}\mathbf{x}^H]$, here Hermitian *Toeplitz*. Running example: the random-phase sinusoid $x[n] = A\sin(n\omega_0 + \phi)$, $\phi \sim U[-\pi, \pi]$, is WSS with $r_x[k] = \frac{1}{2}A^2\cos(k\omega_0)$.

White noise. $v[n]$ is *white* with variance σ_v^2 if it is zero-mean and uncorrelated sample to sample: $r_v[k] = \sigma_v^2\,\delta[k]$. Any uncorrelated sequence qualifies (Gaussian, Bernoulli ± 1, ...): whiteness is a second-moment property, not a distribution.

Power spectral density. For a WSS process the *power spectral density* (PSD) is the DTFT (Lesson 32) of the autocorrelation:

$$P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} r_x[k]\,e^{-jk\omega}, \qquad r_x[k] = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})\,e^{jk\omega}\,d\omega.$$

It is real (by conjugate symmetry of r_x), nonnegative, and integrates to the total power: $r_x[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x$. Keep three examples at hand: white noise $\to$ flat σ_v^2; the random-phase sinusoid $\to$ impulses of weight $\frac{\pi}{2}A^2$ at $\pm\omega_0$; and $r_x[k] = \alpha^{|k|} \to (1-\alpha^2)/(1 - 2\alpha\cos\omega + \alpha^2)$, worked below.

Theorem 39.2 (Wiener–Khinchin)**.** *For a WSS process with absolutely summable autocorrelation, the PSD equals the limit of the expected, normalized magnitude-squared DTFT of the data:*

$$P_x(e^{j\omega}) = \lim_{N\to\infty} \mathbf{E}\Big[\frac{1}{N}\Big|\sum_{n=0}^{N-1} x[n]\,e^{-jn\omega}\Big|^2\Big].$$

The proof is a swap of expectation, sum, and limit — Part VII (Lessons 45–46) supplies the rigor. The theorem is the license to call P_x a "density" of power over frequency, and the seed of estimation: the bracketed quantity at finite N is exactly the periodogram below.

Filtering a WSS process: the $|H|^2$ law. This one derivation is worth keeping. Feed WSS $x[n]$ through a stable LTI filter (Lessons 29, 32), $y[n] = \sum_m h[m]\,x[n-m]$. The output is WSS with mean $m_y = m_x H(e^{j0})$, and

$$r_y[k] = \mathbf{E}\big[y[n+k]\,y^*[n]\big] = \sum_m \sum_l h[m]\,h^*[l]\,r_x[k-m+l] = r_x[k] * h[k] * h^*[-k].$$

Taking DTFTs (convolution theorem, Lesson 32; $h^*[-k]$ transforms to $H^*(e^{j\omega})$):

$$\boxed{P_y(e^{j\omega}) = P_x(e^{j\omega})\,|H(e^{j\omega})|^2.}$$

Phase is invisible to second-order statistics. White noise through H has PSD $\sigma_v^2|H(e^{j\omega})|^2$ — the noise-shaping identity behind every filtered-ADC-noise measurement.

Estimating the PSD: the periodogram and its disease. Given N samples of one realization, the *periodogram* is the finite-N Wiener–Khinchin quantity without the expectation:

$$\hat{P}_{\text{per}}(e^{j\omega}) = \frac{1}{N}\Big|\sum_{n=0}^{N-1} x[n]\,e^{-jn\omega}\Big|^2$$

— one FFT and a squaring. Its *bias* is spectral smoothing: $\mathbf{E}[\hat{P}_{\text{per}}] = \frac{1}{2\pi}P_x * W_B$, the true PSD convolved with the DTFT of a triangular (Bartlett) window, so it is asymptotically unbiased with resolution $\Delta\omega \approx 0.89 \cdot 2\pi/N$ (rule of thumb). The disease is the *variance*: for large N, $\text{var}(\hat{P}_{\text{per}}(e^{j\omega})) \approx P_x^2(e^{j\omega})$, *independent of* N — each frequency bin acts as a bandpass filter of width $\sim 2\pi/N$ whose output power is estimated from *one* sample; more data narrows the bins but never calms them. The periodogram is *inconsistent*: a single FFT of noise looks as ragged at $N = 10^6$ as at $N = 64$. Windowing the data first (the *modified periodogram*) trades main-lobe width for sidelobe level; it cures *masking* of weak tones, not variance.

Averaging: Bartlett and Welch. Consistency comes from averaging independent estimates. **Bartlett**: chop $N = KL$ into K nonoverlapping length-L segments and average their periodograms: variance drops by the factor K, resolution degrades to $0.89 \cdot 2\pi K/N$ — a dial trading variance against resolution. **Welch**: overlap the segments (typically 50%) and window each; overlap recovers some of the averaging windowing throws away. With 50% overlap and triangular windows, $\text{var} \approx \frac{9}{16}\frac{L}{N}P_x^2$ — for fixed resolution, nearly twice Bartlett's effective averaging, and the standard bench estimator. The punchline: with variability $\mathcal{V} = \text{var}/\mathbf{E}^2$ and figure of merit $\mathcal{M} = \mathcal{V}\,\Delta\omega$, every nonparametric method lands at $\mathcal{M} \approx c \cdot 2\pi/N$, $c \in [0.43, 0.89]$: overall performance is bought only with data; the methods just spend the budget differently.

Parametric methods, in one sentence. If the process is well modeled as white noise through a rational filter (an AR model), fitting the few filter coefficients from measured correlations (Yule–Walker) and reading the PSD off the model gives far sharper spectra from short records *when the model fits* — the full Phase 4 booklet (Lessons 10–14) develops this track.

Applications / Course 2. Statistical inference supplies the ensemble language (this lesson is its random-processes chapter in discrete time); DSP uses $P_y = |H|^2 P_x$ for quantization-noise shaping; digital communications' channel models start from white Gaussian noise and its filtered variants. Course 2 Lab 6.4 is this lesson on the bench: the noise-floor measurement is a Welch PSD, the $\varepsilon = 1/\sqrt{n_d}$ error bar is the variance-over-K result rearranged, and the lab averages FFTs because the periodogram is inconsistent.

39.2 Practice: by hand

P1 (PSD of a geometric autocorrelation). Let $r_x[k] = \alpha^{|k|}$, $|\alpha| < 1$. Split the DTFT into $k = 0$ and two geometric tails:

$$P_x(e^{j\omega}) = 1 + \sum_{k=1}^{\infty} \alpha^k e^{-jk\omega} + \sum_{k=1}^{\infty} \alpha^k e^{jk\omega} = 1 + \frac{\alpha e^{-j\omega}}{1 - \alpha e^{-j\omega}} + \frac{\alpha e^{j\omega}}{1 - \alpha e^{j\omega}} = \frac{1 - \alpha^2}{1 - 2\alpha\cos\omega + \alpha^2},$$

real and positive, as a PSD must be — e.g. $\alpha = 0.8$ sweeps from 9 at $\omega = 0$ down to $1/9$ at $\omega = \pi$.

P2 (a measurement budget). $N = 4096$ samples at $f_s = 48$ kHz. Periodogram resolution $\Delta f \approx 0.89\, f_s/N \approx 10.4$ Hz, but variability $\mathcal{V} \approx 1$: error bars as big as the estimate. Welch with $L = 512$, 50% overlap: $K = 2N/L - 1 = 15$ segments, variance $\approx \frac{9}{16}\frac{L}{N} P_x^2 \approx P_x^2/14$ — the noise floor steadies fourteenfold while resolution coarsens to $\approx 0.89\, f_s/L \approx 83$ Hz (wider once segments are windowed).

39.3 Exercises

Exercise 39.1 ([Hand].) (Hayes 3.5.) Find the power spectrum of the WSS process with each of the following autocorrelation sequences: (a) $r_x[k] = 2\delta[k] + j\delta[k-1] - j\delta[k+1]$; (b) $r_x[k] = \delta[k] + 2(0.5)^{|k|}$; (c) $r_x[k] = 2\delta[k] + \cos(\pi k/4)$; (d) $r_x[k] = 10 - |k|$ for $|k| < 10$, and 0 otherwise.

Exercise 39.2 ([Hand].) (Hayes 8.3.) Bartlett's method is used to estimate the power spectrum of a process from $N = 2000$ samples. (a) What is the minimum segment length L that gives resolution $\Delta f = 0.005$ (cycles/sample)? (b) Why is it not advantageous to increase L beyond that value? (c) With quality factor $Q = 1/\mathcal{V}$, find the minimum N giving $\Delta f = 0.005$ and Q five times the periodogram's.

Exercise 39.3 ([Proof].) (Hayes 3.16.) Show that the cross-correlation $r_{xy}[k] = \mathbf{E}[x[n+k]\, y^*[n]]$ of jointly WSS processes satisfies (a) $|r_{xy}[k]| \le \big[r_x[0]\, r_y[0]\big]^{1/2}$ and (b) $|r_{xy}[k]| \le \frac{1}{2}\big[r_x[0] + r_y[0]\big]$.

Deeper reading. The Phase 4 booklet's Lessons 10 and 14 carry the full story: ergodic theorems, eigenvalue bounds, spectral factorization, Blackman–Tukey, minimum variance, maximum entropy, MUSIC. Sources: Hayes, *Statistical Digital Signal Processing and Modeling*, Ch. 3 and 8; Lyons, *Understanding Digital Signal Processing*, Ch. 11 (averaging in bench units); Oppenheim–Schafer–Buck (2nd ed.) 10.5–10.6 for the careful bias/variance analysis of the periodogram and its windowed and averaged repairs.

40 Optimal and Adaptive Filters: Wiener, LMS, and the Kalman Filter

NEEDED FOR: DSP, statistical inference (estimation), communications (equalization uses LMS); Course 2 Labs 6.6 (real-time Kalman) and 6.8 (LMS adaptive equalizer)

40.1 Theory

The FIR Wiener filter. Estimate a process $d[n]$ from a jointly WSS related process $x[n]$ (Lesson 39 supplies the vocabulary) using a p-tap FIR filter,

$$\hat{d}[n] = \sum_{l=0}^{p-1} w(l)\, x[n-l], \qquad \xi = \mathbf{E}\big[(d[n]-\hat{d}[n])^2\big] \to \min.$$

This is exactly Lesson 22's LMS estimation problem restricted to a linear combination of the last p samples, so the same orthogonality principle decides it: at the optimum, the error $e[n] = d[n]-\hat{d}[n]$ is orthogonal to every observation used, $\mathbf{E}\big[e[n]\,x[n-k]\big] = 0$ for $k = 0,\dots,p-1$. Expanding gives the *Wiener–Hopf (normal) equations* and minimum error

$$\boxed{\mathbf{R}_x\mathbf{w} = \mathbf{r}_{dx}} \qquad \sum_{l=0}^{p-1} w(l)\, r_x(k-l) = r_{dx}(k), \qquad \xi_{\min} = r_d(0) - \mathbf{r}_{dx}^{\mathsf{T}}\mathbf{w},$$

where $\mathbf{R}_x$ is the symmetric Toeplitz autocorrelation matrix of Lesson 39 and $r_{dx}(k) = \mathbf{E}[d[n]\,x[n-k]]$ is the cross-correlation. Everything is in the wiring of d to x: *smoothing* takes $x[n] = d[n]+v[n]$ with noise uncorrelated with signal, so $\mathbf{R}_x = \mathbf{R}_d + \mathbf{R}_v$ and $\mathbf{r}_{dx} = \mathbf{r}_d$; *linear prediction* takes $d[n] = x[n+1]$, making $\mathbf{r}_{dx}$ a shifted autocorrelation; *noise cancellation* estimates the primary noise from a correlated secondary sensor and subtracts it — all three are the same $p \times p$ solve with different right-hand sides. The Practice section works the classic smoothing example by hand; the optimal filter comes out lowpass, because that is where the SNR lives.

From steepest descent to LMS. When $\mathbf{R}_x$ and $\mathbf{r}_{dx}$ are unknown (or drifting), descend the quadratic error surface instead of solving: steepest descent iterates $\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\,(\mathbf{r}_{dx} - \mathbf{R}_x\mathbf{w}_n)$. The *LMS algorithm* replaces the gradient by its one-sample unbiased estimate $e[n]\,\mathbf{x}[n]$, with $\mathbf{x}[n] = [x[n],\dots,x[n-p+1]]^{\mathsf{T}}$:

$$\boxed{\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\, e[n]\,\mathbf{x}[n], \qquad e[n] = d[n] - \mathbf{w}_n^{\mathsf{T}}\mathbf{x}[n].}$$

Averaging the recursion (under the standard independence assumption) shows $\mathbf{E}[\mathbf{w}_n]$ obeys exactly the steepest-descent iteration; rotating into the eigenbasis of $\mathbf{R}_x$ (Lesson 10) decouples the modes into factors $(1-\mu\lambda_k)^n$, so the mean weights converge if and only if

$$0 < \mu < \frac{2}{\lambda_{\max}}, \qquad \text{conservatively} \quad \mu < \frac{2}{\operatorname{tr}(\mathbf{R}_x)} = \frac{2}{p\,\mathbf{E}[x^2[n]]},$$

since $\lambda_{\max} \le \operatorname{tr}(\mathbf{R}_x)$ — a bound you can estimate from signal power alone. The noisy gradient never lets the filter settle exactly: the steady-state error exceeds $\xi_{\min}$ by an excess whose relative size, the *misadjustment* $\mathcal{M} \approx \frac{1}{2}\mu\operatorname{tr}(\mathbf{R}_x)$, is the design dial trading convergence speed against steady-state noise. Convergence speed itself is hostage to the eigenvalue spread of $\mathbf{R}_x$, and the scale-free *normalized LMS* step $\mu[n] = \beta/(\mathbf{x}^{\mathsf{T}}[n]\mathbf{x}[n] + \epsilon)$, $0 < \beta < 2$, is the form that survives input-level changes in practice.

Recursive least squares, in one paragraph. RLS abandons ensemble statistics and *exactly* minimizes the weighted data cost $\sum_{i=0}^{n} \lambda^{n-i} e^2(i)$ with forgetting factor $0 < \lambda \le 1$: the resulting normal equations update by rank-one recursions, and Woodbury's identity turns the inverse

correlation matrix into a recursion of its own, so no inversion is ever performed. What it buys: convergence in roughly $2p$ samples *independent of the eigenvalue spread* that throttles LMS; what it costs: $O(p^2)$ multiplies per sample against LMS's $O(p)$, plus care with numerics (the recursed inverse can lose symmetry in fixed point). Choosing $\lambda < 1$ forgets old data over an effective window $\sim 1/(1-\lambda)$, the price of tracking drift. Structurally, RLS is the Kalman filter below with the state frozen — the gain and inverse correlation recursions are predict/correct in disguise.

The Kalman filter. The Wiener filter assumes stationarity and an infinite past. The Kalman filter drops both by modeling the world in *state space*:

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\,\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \qquad \mathbf{z}_k = \mathbf{H}_k\,\mathbf{x}_k + \mathbf{v}_k,$$

with white, mutually uncorrelated process noise $\mathbf{w}_k \sim (\mathbf{0}, \mathbf{Q}_k)$ and measurement noise $\mathbf{v}_k \sim (\mathbf{0}, \mathbf{R}_k)$. Here $\mathbf{\Phi}$ propagates the state one sample ahead (for a continuous plant $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x}$ sampled every Δt, $\mathbf{\Phi} = e^{\mathbf{F}\Delta t}$ — where firmware sample-period constants come from), and $\mathbf{H}$ says what the sensors see; an AR(p) process fits with a companion $\mathbf{\Phi}$ and $\mathbf{H} = [1, 0, \dots]$. Writing $\hat{\mathbf{x}}_k(-), \mathbf{P}_k(-)$ for the estimate and its error covariance *before* seeing $\mathbf{z}_k$ and $(+)$ for after, the filter alternates two moves:

$$\begin{aligned}
&\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_{k-1}\,\hat{\mathbf{x}}_{k-1}(+), && \mathbf{P}_k(-) = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}(+)\mathbf{\Phi}_{k-1}^{\mathsf{T}} + \mathbf{Q}_{k-1} && \text{(predict)}\\
&\quad \mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^{\mathsf{T}}\big[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^{\mathsf{T}} + \mathbf{R}_k\big]^{-1} && && \text{(gain)}\\
&\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k\big[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-)\big], && \mathbf{P}_k(+) = \big[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\big]\mathbf{P}_k(-) && \text{(update)}
\end{aligned}$$

initialized at $\hat{\mathbf{x}}_0(+) = \mathbf{E}[\mathbf{x}_0]$, $\mathbf{P}_0(+) = \operatorname{cov}(\mathbf{x}_0)$. Read it as a blending machine. The bracket $\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-)$ is the *innovation*, the part of the new measurement the model could not predict; the gain $\mathbf{K}_k$ decides how much of it to believe, comparing the filter's own uncertainty $\mathbf{P}_k(-)$ against the sensor's $\mathbf{R}_k$. Worthless measurements ($\mathbf{R}_k \to \infty$) drive $\mathbf{K} \to \mathbf{0}$ and the estimate coasts on the model; a huge prior uncertainty makes the filter swallow the measurement whole. Estimating a constant with no prior gives exactly $K_k = 1/k$ — the recursive sample mean. The covariance $\mathbf{P}$ is the filter's self-assessed confidence, and (for a linear model) gains and covariances never touch the data, so they can be computed offline. The derivation is Lesson 22's orthogonality principle once more — demand the error be orthogonal to every measurement — and lives in full in the Phase 4 booklet (Hayes Ch. 7; Grewal & Andrews Ch. 4); one payoff worth quoting: for a stationary model the gain settles to a constant, and the steady-state Kalman filter *is* the causal Wiener filter. In Hayes' AR(1)-in-noise example the gain sequence marches $0.5, 0.4048, 0.3824, \cdots \to 0.375$, the causal Wiener gain exactly.

Beyond linear: the EKF, and the craft. Real plants and sensors are rarely linear, and the workhorse fix is the *extended Kalman filter*: propagate the state through the full nonlinear model, but run the covariance and gain equations on Jacobians evaluated at the current estimate — optimality is surrendered, usefulness is not (the Phase 4 booklet's final lesson develops it). In practice the equations are the easy part; the craft is *tuning* $\mathbf{Q}$ and $\mathbf{R}$ — translating datasheet noise specs, inflating $\mathbf{Q}$ to absorb modeling debt — and then *checking* the filter: keep $\mathbf{P}$ symmetric and positive definite every cycle (the Joseph form of the covariance update is the robust way), and test claimed against actual performance by comparing $\mathbf{P}$ with the observed

innovation statistics. When a state is structurally unobservable, $\mathbf{P}$ grows without bound no matter the tuning, and the remedy is hardware (another sensor), not software.

> The orthogonality-principle estimation here is statistical inference material; DSP covers the adaptive-filter thread, and digital communications' channel equalizers are LMS/NLMS filters with the training sequence playing the role of $d[n]$ — exactly Course 2 Lab 6.8's adaptive equalizer, where the misadjustment formula is the datasheet for picking μ. Course 2 Lab 6.6 puts the predict/update loop on the bench in real time: five lines of linear algebra per sample, $\mathbf{Q}/\mathbf{R}$ from sensor datasheets, and the symmetry check on $\mathbf{P}$ as a firmware assertion.

40.2 Practice: by hand

P1 (two-tap Wiener smoother). Let $d[n]$ be AR(1) with $r_d(k) = 0.8^{|k|}$, observed in unit white noise: $x[n] = d[n] + v[n]$. With $p = 2$, $\mathbf{R}_x = \mathbf{R}_d + \mathbf{I}$ and $\mathbf{r}_{dx} = [1, 0.8]^{\mathsf{T}}$:

$$\begin{bmatrix} 2 & 0.8 \\ 0.8 & 2 \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix} \Rightarrow \mathbf{w} = \begin{bmatrix} 0.4048 \\ 0.2381 \end{bmatrix}, \quad \xi_{\min} = 1 - 0.4048 - 0.8(0.2381) = 0.4048,$$

a gain of about +2.3 dB in SNR. Check the orthogonality principle directly: $\mathbf{E}[e[n]x[n]] = r_{dx}(0) - 2w(0) - 0.8w(1) = 0$.

P2 (scalar Kalman filter; G&A Example 4.1). Take $\Phi = H = 1$, $Q = 1$, $R = 2$, $P_0(+) = 10$. Step 1: $P_1(-) = 10 + 1 = 11$, $K_1 = 11/13 = 0.846$, $P_1(+) = (1 - K_1)\,11 = 1.692$. Step 2: $P_2(-) = 2.692$, $K_2 = 0.574$, $P_2(+) = 1.148$. The fixed point satisfies $P = 2(P+1)/(P+3)$, i.e. $P^2 + P - 2 = 0$, so $P_\infty(+) = 1$ and $K_\infty = 2/4 = \frac{1}{2}$: the wild initial uncertainty is forgotten geometrically, and the steady filter splits the difference between model and measurement.

40.3 Exercises

Exercise 40.1 ([Hand].) (Hayes 7.5.) A signal $d[n]$ satisfying $d[n] = 0.5\,d[n-1] + v[n]$, with $v[n]$ white of variance $\sigma_v^2 = 1$, is observed in uncorrelated white noise: $x[n] = d[n] + w[n]$ with $r_w(k) = 0.5\,\delta(k)$. Design a first-order FIR linear predictor $W(z) = w(0) + w(1)z^{-1}$ operating on $x[n]$ to estimate $d[n+1]$, and find the mean-square prediction error $\xi = \mathbf{E}[(d[n+1] - \hat{d}[n+1])^2]$.

Exercise 40.2 ([Hand].) (Hayes 9.9.) The first three autocorrelations of a process $x[n]$ are $r_x(0) = 1$, $r_x(1) = 0.5$, $r_x(2) = 0.5$. Design a two-coefficient LMS adaptive linear predictor for $x[n]$ with misadjustment $\mathcal{M} = 0.05$ (i.e. choose μ), find the optimal weights, and compute the steady-state mean-square error $\xi(\infty)$.

Exercise 40.3 ([Hand].) (Hayes 7.18.) An AR(1) process $x[n] = 0.5\,x[n-1] + w[n]$, with $\sigma_w^2 = 0.64$, is observed as $y[n] = x[n] + v[n]$ with $\sigma_v^2 = 1$. (a) Write the scalar Kalman filter equations for $\hat{x}(n|n)$ given $\hat{x}(0|0) = 0$ and error variance $P(0|0) = 1$. (b) Find the steady-state Kalman gain and the limiting form of the estimation recursion.

Deeper reading. The Phase 4 booklet is the uncut version: Hayes Ch. 7 adds the IIR Wiener filters via spectral factorization, Hayes Ch. 9 the full LMS/NLMS/RLS analysis, and Grewal & Andrews Chs. 2–5 and 7 the state-space modeling toolkit, the Riccati equation, the EKF, and the engineering lore (divergence diagnosis, innovation gating, square-root filters).

41 Detection and Digital Communication

NEEDED FOR: Digital communications and statistical inference; Course 2 Labs 6.7–6.9

41.1 Theory

The matched filter maximizes SNR. A known waveform $x(t)$ arrives in white noise of PSD $N_0/2$; which LTI receiver $h(t)$ maximizes the output SNR at a chosen instant T_M? The output signal at T_M is the pairing $\frac{1}{2\pi}\int X(\Omega)H(\Omega)e^{j\Omega T_M}\,d\Omega$ and the output noise power is proportional to $\int |H(\Omega)|^2\,d\Omega$, so the question is: which H maximizes a fixed pairing against X per unit of its own energy? Cauchy–Schwarz (Lesson 2's inequality, in integral form; the pairing view returns as Part VII Lesson 50's dual pairing) answers in one line: the ratio is maximized, with equality, iff

$$H(\Omega) = \alpha\, X^*(\Omega)\, e^{-j\Omega T_M} \qquad \Longleftrightarrow \qquad h(t) = \alpha\, x^*(T_M - t),$$

the *matched filter*: conjugate the spectrum, i.e. time-reverse and conjugate the waveform, so the receiver is a *correlator* with the transmitted waveform as reference. The maximum SNR is

$$\chi_{\max} = \frac{2E}{N_0},$$

which depends *only on waveform energy* E, not on modulation: two equal-energy waveforms through their own matched filters detect equally well. Modulation buys *resolution*, not SNR. In colored interference, whiten first, then match. For a simple pulse of width τ the matched-filter output is the 2τ-wide triangle of Lesson 29's convolution, peaking at αE.

Pulse compression. A simple pulse gives range resolution $\Delta R = c\tau/2$, so long pulses (energy) and fine range resolution conflict directly. *Linear-FM pulse compression* resolves it: sweep β Hz of frequency across the τ-second pulse, and the matched-filter output compresses to width $\approx 1/\beta$, giving $\Delta R = c/2\beta$ — a factor $\beta\tau$ (the *time-bandwidth product*) narrower than the uncompressed pulse: the energy of the long pulse with the resolution of the short one. The price is *range–Doppler coupling*: a Doppler shift F_D slides the LFM peak in delay by $\Delta t = -F_D\tau/\beta$, so a moving target appears displaced in range with only a gentle amplitude loss. (Ambiguity functions and the multipulse range/Doppler trade are in the Phase 4 Supplement, Lessons 1 and 3.)

Detection is binary hypothesis testing. Decide H_0 (interference only) vs H_1 (target + interference); the error currencies are the false-alarm probability P_{FA} and the detection probability P_D. Radar cannot price the two errors, so it uses the *Neyman–Pearson rule*: maximize P_D subject to $P_{FA} \le \alpha$ — whose solution is always the *likelihood ratio test*: compare $\Lambda(\mathbf{x}) = p(\mathbf{x} \mid H_1)/p(\mathbf{x} \mid H_0)$ (or its log) to a threshold set by the P_{FA} budget. Sweeping the threshold traces the *ROC curve*, P_D against P_{FA}: each threshold is one operating point, and the whole curve is the detector's quality. For a known signal in Gaussian noise (Lesson 23's log-likelihoods) the LRT *is* the matched filter followed by a threshold; with unknown phase it becomes the envelope detector. For square-law-detected noise (exponential PDF) the threshold for a desired false-alarm rate is $T = -\sigma^2 \ln P_{FA}$, and a Swerling-1 fluctuating target obeys the tidy $P_D = P_{FA}^{1/(1+\bar{\chi})}$ at average SNR $\bar{\chi}$.

Cell-averaging CFAR. The fixed threshold needs σ^2, which in the field varies by tens of dB (clutter, jamming, weather). *CFAR processing* estimates it from the data. The cell-averaging recipe: (1) slide a window over the range(–Doppler) map; (2) for each *cell under test*, average N *reference cells*, excluding *guard cells* around the CUT so a straddling target does not pollute the estimate — the sample mean is exactly the ML estimate of the exponential parameter; (3) threshold at $\hat{T} = \alpha\,\hat{\sigma}^2$ with

$$\alpha = N\left(P_{FA}^{-1/N} - 1\right) \qquad \Longleftrightarrow \qquad P_{FA} = \left[1 + \frac{\alpha}{N}\right]^{-N},$$

so the achieved false-alarm rate depends only on N and α — *not* on the interference power. The finite-N estimate costs a *CFAR loss* (a threshold above the ideal, shrinking as N grows), and the two assumptions — homogeneous interference, target-free reference cells — fail instructively: clutter edges inflate the estimate and neighboring targets mask each other; greatest-of/smallest-of and order-statistic variants are the standard fixes.

Signal space and the optimum AWGN receiver. Digital modulation maps bit groups to waveforms; the organizing idea is *signal space*: expand the M candidate waveforms over an orthonormal basis $\phi_n(t)$ and each becomes a point $\mathbf{s}_m$ — geometry replaces calculus. The families: *PAM* (amplitudes on a line, average energy growing as $(M^2-1)/3$), *PSK* (points on a circle), *QAM* (grid or ring constellations, more energy-efficient than PSK at the same M), with *Gray coding* labeling neighbors by single-bit differences so the dominant nearest-neighbor errors cost one bit. The receiver projects $r(t) = s_m(t) + n(t)$ onto the basis — a correlator bank, or equivalently matched filters sampled at the symbol rate — and this finite vector $\mathbf{r}$ is a *sufficient statistic*: nothing else in the waveform matters. Optimum detection is MAP — for equiprobable messages, maximum likelihood — and in AWGN the likelihood is monotone in distance, so the rule is *nearest neighbor*: choose the constellation point closest to $\mathbf{r}$, computable through the correlation metrics $C(\mathbf{r}, \mathbf{s}_m) = 2\sum_n r_n s_{mn} - \sum_n s_{mn}^2$. Error rates follow from the geometry and are stated, not derived, in terms of the Gaussian tail $Q(x)$: binary *antipodal* signaling ($\mathbf{s}_2 = -\mathbf{s}_1$, e.g. BPSK) achieves

$$P_e = Q\left(\sqrt{\tfrac{2E_b}{N_0}}\right),$$

the best any binary scheme can do at bit energy E_b; binary *orthogonal* signaling pays 3 dB, $P_e = Q(\sqrt{E_b/N_0})$; and M-ary PSK/QAM curves come from union bounds over neighbor pairs, $P_e \approx N_{\min}\,Q\big(d_{\min}/\sqrt{2N_0}\big)$ — more bits per symbol trade E_b/N_0 margin for rate. This is the matched filter deciding *which* of M signals rather than *whether* one is present.

ISI and the adaptive equalizer. Band-limited channels smear symbols into each other — *intersymbol interference*. Two design criteria for a linear equalizer: *zero forcing* (invert the folded channel spectrum — kills ISI but amplifies noise exactly where the channel is weak) and *MMSE* (minimize $\mathbf{E}|I_k - \hat{I}_k|^2$ — the Wiener/LMS-estimation solution of Lesson 22, trading residual ISI against noise enhancement). When the channel is unknown or drifting, adapt: run Lesson 40's *LMS algorithm*, $\mathbf{C}_{k+1} = \mathbf{C}_k + \Delta\,\varepsilon_k \mathbf{V}_k^*$, first on a known *training sequence*, then *decision-directed* on its own detected symbols. Stability requires $0 < \Delta < 2/\lambda_{\max}$ of the input covariance Γ; the step size trades convergence speed against excess steady-state MSE (self-noise); and convergence speed is hostage to the eigenvalue spread $\lambda_{\max}/\lambda_{\min}$ — channels with deep spectral nulls equalize slowly (RLS exists for exactly this).

Communications / Course 2. This lesson is digital communications' core chain — constellation, optimum AWGN receiver, equalizer — with statistical inference supplying the hypothesis-testing and Gaussian machinery underneath; the matched filter, the LRT, and the nearest-neighbor receiver are one theorem in three costumes. Course 2 builds each block: Lab 6.7 transmits a chirp and matched-filters by FFT to read range off the compressed peak (pulse compression on the bench), Lab 6.9 sets CFAR thresholds and traces measured ROC curves against the false-alarm budget, and Lab 6.8 trains an LMS equalizer and watches the eye reopen — Lesson 40's adaptive filter with a modem's job.

41.2 Practice: by hand

P1 (compression). A 10 μs simple pulse: $\Delta R = c\tau/2 = 1500$ m. The same pulse with a 50 MHz LFM sweep: $\Delta R = c/2\beta = 3$ m, compression ratio $\beta\tau = 500$.

P2 (thresholds and CFAR loss). Square-law detector, known σ^2: $P_{FA} = 10^{-4}$ needs $T/\sigma^2 = -\ln 10^{-4} = 9.21$ (9.6 dB); 10^{-6} needs 13.8 (11.4 dB). Cell-averaging CFAR with $N = 16$ at $P_{FA} = 10^{-4}$: $\alpha = 16(10^{1/4} - 1) = 12.45$, against the ideal 9.21 — a CFAR loss of $10\log_{10}(12.45/9.21) \approx 1.3$ dB.

P3 (error rates). BPSK at $E_b/N_0 = 9.6$ dB: $P_e = Q(\sqrt{2 \times 9.12}) = Q(4.27) \approx 10^{-5}$. Square 16-QAM with spacing $2d$: average symbol energy $10d^2$, so $E_b = 2.5d^2$ over 4 bits; QPSK at the same $d_{\min} = 2d$ has $E_b = d^2$ — 16-QAM pays $10\log_{10} 2.5 \approx 4$ dB per bit for twice the rate.

P4 (equalizer step size). Input-covariance eigenvalues $\{0.2, 1.0, 2.6\}$: LMS is stable for $\Delta < 2/2.6 \approx 0.77$, and the slowest mode converges roughly $\lambda_{\max}/\lambda_{\min} = 13$ times slower than the fastest.

41.3 Exercises

Exercise 41.1 ([Hand)**.**] (Richards Ch. 4, prob. 5.) A linear FM waveform sweeps from 9.5 to 10.5 GHz over a pulse length of 20 μs. What is the bandwidth β? The time-bandwidth product? The Rayleigh (peak-to-first-null) range resolution of the matched filter output in meters? And what would be the Rayleigh resolution in meters of a square pulse of the same energy (assuming both have the same amplitude)?

Exercise 41.2 ([Hand)**.**] (Richards Ch. 6, prob. 1.) Under H_0 the observation has PDF $p_x(x \mid H_0) = \alpha e^{-x/\alpha}$, $0 \le x < \infty$; under H_1, $p_x(x \mid H_1) = \beta e^{-x/\beta}$, with $\beta > \alpha$. Find the likelihood ratio and the log likelihood ratio, and show the LRT reduces to comparing x itself to a threshold.

Exercise 41.3 ([Hand)**.**] (P&S 10.5.) The tap-leakage LMS algorithm is $\mathbf{C}_N(n+1) = w\,\mathbf{C}_N(n) + \Delta\,\varepsilon(n)\,\mathbf{V}_N^*(n)$ with $0 < w < 1$ and step size Δ. Determine the condition for convergence of the mean of $\mathbf{C}_N(n)$.

Deeper reading. The Phase 4 Supplement carries the full derivations and problem sets: ambiguity functions, stretch processing, and phase codes (Richards Ch. 4); Albersheim's equation, Swerling models, and binary integration (Ch. 6); the P&S signal-space and receiver problem chains (Chs. 3–4); and DFE, RLS, and blind equalization (Ch. 10). Its Doppler/MTI/tracking lesson (Richards Chs. 5, 7) is the natural companion we skipped here. Kay's *Detection Theory* and Proakis & Salehi Ch. 9 (Nyquist pulses, MLSE) are the graduate-level continuations.

42 Images: Convolution, Filtering, Restoration, Edges, and Motion

NEEDED FOR: Image processing and computer vision; Course 2 Labs 9.4–9.6, 8.3

42.1 Theory

Images as arrays. A continuous scene $f(s,t)$ becomes a digital image by sampling and quantization — Lesson 33's story, run once per axis. The result is an $M \times N$ array $f(x,y)$ with the origin at the *top left* and intensities on $L = 2^k$ integer levels. Aliasing is also Lesson 33 twice over: a band-limited image is recoverable iff the sampling rate exceeds twice the bandwidth per axis, jaggies and moiré are the 2-D aliases, and the rules are the same — anti-alias *before* sampling (optical blur; there is no software cure afterward), and resampling is sampling again, so smooth before shrinking.

Spatial filtering: correlation and convolution. A linear spatial filter replaces each pixel by a sum of products of an $m \times n$ kernel $w(s,t)$ ($m = 2a+1$, $n = 2b+1$, odd) with the neighborhood:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s,\, y+t).$$

This is *correlation*; *convolution* pre-rotates the kernel by 180°. The litmus test: convolving with a unit impulse yields an exact copy of the kernel (Lesson 29's sifting property, twice over); correlating yields it rotated. The two coincide only for symmetric kernels — which is why the distinction is invisible until it bites. Borders need padding (a rows, b columns minimum; zero, replicate, and mirror padding trade artifacts), and convolution's algebra — commutative, associative, distributive — is what the transform theory below requires.

Smoothing and sharpening kernels. *Box* kernels (all ones, normalized) are cheapest and worst: not isotropic (their transform is a 2-D sinc with directional lobes). *Gaussian* kernels $G(s,t) = Ke^{-(s^2+t^2)/2\sigma^2}$ are the only circularly symmetric separable kernels; size $\lceil 6\sigma \rceil$ (odd) captures essentially all the mass — bigger is waste, smaller truncates. Cascaded Gaussians obey $\sigma^2 = \sigma_1^2 + \sigma_2^2$, and repeated box filtering tends Gaussian — the central limit theorem, now with kernels. Sharpening is differentiation: the digital Laplacian

$$\nabla^2 f = f(x{+}1,y) + f(x{-}1,y) + f(x,y{+}1) + f(x,y{-}1) - 4f(x,y),$$

kernel [0 1 0; 1 −4 1; 0 1 0], gives $g = f - \nabla^2 f$; *unsharp masking* ($g = f + k(f - \bar{f})$, subtract a blur, add the difference back) is the same idea in disguise.

Separability. A rank-1 kernel $w = \mathbf{v}\mathbf{w}^{\mathsf{T}}$ filters in two 1-D passes: $(m+n)$ multiplies per pixel instead of mn — for a 15×15 kernel, a 7.5× saving. Box, Gaussian, and Sobel kernels all qualify, and the two-pass result equals the one-pass result to machine precision. On an embedded target this is the difference between a frame rate and a slide show (Lab 9.4).

The 2-D DFT and frequency-domain filtering.

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(ux/M+vy/N)}$$

— Lesson 12's DFT run on rows then columns (the kernel is separable), so the FFT gives $O(MN \log MN)$. Translation changes phase only; real images give conjugate-symmetric transforms; $F(0,0) = MN\bar{f}$; magnitude says how much of each frequency, phase says *where*. Products of DFTs implement *circular* convolution, so linear filtering demands padding (typically to $2M \times 2N$): unpadded, a blur wraps around and borrows from the opposite edge. Filtering is $g = \text{IDFT}[HF]$ with a real, centered, symmetric H; with $D(u,v)$ the distance from the center of the (padded) frequency rectangle, the ideal lowpass ($H = 1$ for $D \le D_0$, else 0) *rings* — its spatial kernel is a 2-D sinc, so this is Gibbs — while the Gaussian $H = e^{-D^2/2D_0^2}$ is ring-free, and highpass filters come free as $1 - H_{LP}$.

Restoration: why the inverse filter explodes, and Wiener. Restoration models the degradation: $g = h * f + \eta$, i.e. $G = HF + N$. Direct inversion $\hat{F} = G/H = F + N/H$ blows up wherever H is small — and real blurs guarantee such places: uniform linear motion gives a sinc-shaped H with periodic *zeros*, and the streaky "curtain of noise" in inverse-filtered images is exactly N/H erupting there. The principled fix is Lesson 40's Wiener filter with $D(u,v)$ in place of ω: minimizing $\mathbf{E}[(f - \hat{f})^2]$ gives

$$\hat{F} = \left[\frac{1}{H}\,\frac{|H|^2}{|H|^2 + S_\eta/S_f}\right] G \;\approx\; \left[\frac{1}{H}\,\frac{|H|^2}{|H|^2 + K}\right] G,$$

inverse filtering tempered by the noise-to-signal ratio: $\to 1/H$ where signal dominates, $\to 0$ where noise does, with the constant-K form tuned interactively when the spectra are unknown. (Tikhonov/constrained least squares replaces S_η/S_f by $\gamma|P(u,v)|^2$ with P the Laplacian's transform.) For impulse (salt-and-pepper) noise no linear filter wins: the *median* of the window discards impulses outright — probability's tool, not Fourier's.

Edges: Sobel and the Canny recipe. Edge strength and direction come from the gradient $\nabla f = (g_x, g_y)$: magnitude $M(x,y) = \sqrt{g_x^2 + g_y^2}$, angle $\alpha = \tan^{-1}(g_y/g_x)$, edge direction *orthogonal* to the gradient. Noise wrecks bare derivatives, so every serious detector smooths first. The standard kernels are Sobel's,

$$g_x: \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \qquad g_y: \text{ its transpose}$$

— separable into a difference pass $[-1\ 0\ 1]$ and a smoothing pass $[1\ 2\ 1]$, so they smooth crossways while differencing. Canny's optimal detector is the four-step recipe: (1) smooth with a Gaussian; (2) compute gradient magnitude and angle; (3) *nonmaxima suppression* — zero any pixel beaten by either neighbor along its edge normal, collapsing ridges to single-pixel width; (4) *hysteresis* — with T_H:T_L between 2:1 and 3:1, keep pixels above T_H and adopt those above T_L only where 8-connected to a kept one. For segmenting what the edges enclose, Otsu picks the threshold from the histogram alone by maximizing the between-class variance $\sigma_B^2(k)$ — one sweep over the histogram.

Corners, descriptors, and motion: the bridge onward. The Harris–Stephens detector linearizes a patch's shift-SSD to $C = [x\ y]M[x\ y]^{\mathsf{T}}$ with $M = \sum w \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$: both eigenvalues small = flat, one large = edge, both large = corner, and $R = \det M - k\,\mathrm{trace}^2 M$ skips the eigenvalue cost. SIFT adds scale invariance (difference-of-Gaussian extrema across a scale pyramid, a 128-number orientation-histogram descriptor); this detector–descriptor–match pattern is the entry point to computer vision's geometry. Video's temporal cousin is *block-matching motion estimation*: find, for each block, the displacement into the previous frame that best predicts it — Lab 9.6's search loop.

Everything here is Part V promoted to two indices: the filter sum is Lesson 29's convolution with two subscripts, the 2-D DFT is Lesson 12's unitary transform on rows then columns, and Wiener deconvolution is Lesson 40's orthogonality principle (Lesson 22) in the frequency plane. Course 2: Lab 9.4 implements exactly this lesson's 2-D convolution on the embedded target (separability and fixed-point kernels decide whether the frame loop closes in time); Lab 9.5 runs per-frame Sobel / thresholding; Lab 9.6's motion estimation is the block-matching paragraph; and Lab 8.3's learned denoiser is benchmarked against the Wiener baseline above. For computer vision this is the classical prerequisite layer its geometry consumes.

42.2 Practice: by hand

P1. Apply the -4-center Laplacian kernel to the center of the patch [10 10 10; 10 20 10; 10 10 10] and interpret the sign of the response.

P2. A 3×3 neighborhood has rows $(0, 0, 10)$, $(0, 10, 10)$, $(10, 10, 10)$. Compute g_x and g_y with the Sobel kernels, then M and α, and give the edge direction.

P3. At a frequency where $|H| = 0.1$, compare the inverse filter's gain with the Wiener gain for $K = 0.01$; then show the constant-K Wiener formula reduces to $1/H$ as $K \to 0$.

42.3 Exercises

Exercise 42.1 ([Hand].] (G&W 3.27) An image is filtered four times with a 3×3 Gaussian kernel of standard deviation 1.0. By associativity, the same result comes from a single Gaussian kernel. (a) What is its size? (b) What is its standard deviation?

Exercise 42.2 ([Hand].] (G&W 4.47) A 3×3 kernel averages the four closest neighbors of a point (x, y) but excludes the point itself from the average. (a) Find the equivalent frequency-domain transfer function $H(u, v)$. (b) Show that it is a lowpass filter transfer function.

Exercise 42.3 ([Hand].] (G&W 5.30) A motion-blurred image corrupted by additive Gaussian noise is inverse-filtered. The blurring itself is corrected, but the restored image shows a strong streak pattern that is not apparent in the blurred image. Explain how this pattern originated.

Deeper reading. The Phase 5 booklet carries the full versions across its Lessons 1–11: histogram processing, the Butterworth family and notch filters, the noise zoo and adaptive medians, constrained least squares, LoG/zero crossings, the Hough transform, region-based segmentation, moment invariants, texture, and MSER/SIFT in detail — with proofs and the complete G&W exercise sets. Tekalp's motion-estimation chapters are the video extension Lab 9.6 draws on.

43 Audio Signal Processing: Quantization, Effects, Coding, and Learned DSP

NEEDED FOR: Digital signal processing; Course 2 Labs 8.2–8.4, 9.2–9.3

What quantizing a signal costs (and how dither and noise shaping spend that cost wisely), how EQ, dynamics, and coding are designed, and the on-ramp to *learned* DSP. Assumes Lessons 37 (biquads, bilinear) and 38 (sigma-delta, STFT).

43.1 Theory

Uniform quantization and the 6-dB-per-bit law. A word-length-w uniform quantizer over $\pm x_{\max}$ has step size $Q = 2x_{\max}/2^w$ and output $x_Q[n] = Q \cdot \mathrm{round}(x[n]/Q)$. The *classical model* treats the error $e[n] = x_Q[n] - x[n]$ as additive noise, uniform on $(-Q/2, Q/2]$, white, and independent of the signal: $\mathbf{E}[e] = 0$, $\sigma_E^2 = Q^2/12$. Writing the signal power through the *peak factor* $P_F = x_{\max}/\sigma_X$,

$$\mathrm{SNR} = 10\log_{10}\frac{\sigma_X^2}{\sigma_E^2} = 10\log_{10}\frac{x_{\max}^2/P_F^2}{(2x_{\max}2^{-w})^2/12} = 6.02\,w \;-\; 10\log_{10}(P_F^2/3) \quad [\mathrm{dB}].$$

Special cases: full-scale sine ($P_F = \sqrt{2}$) gives the famous $6.02w + 1.76$ dB; a uniform PDF ($P_F = \sqrt{3}$) gives $6.02w$; near-Gaussian real audio, run at $P_F = 4.61$ so overload probability stays below 10^{-5}, gives $6.02w - 8.5$ dB — headroom is paid for in SNR. The model holds for wide-dynamic-range inputs exercising many steps; for *small* signals (a few Q of amplitude) the error is a deterministic, periodic function of the signal and its spectrum is harmonic distortion, not noise. (The deeper story, Widrow's quantization theorem, treats quantization as *sampling of the amplitude PDF* — Shannon's theorem restated in amplitude — with the classical model as its baseband shadow.)

Dither. *Dither* rescues the small-signal case: add a random sequence $d[n]$ *before* requantizing. Uniform (RECT) dither on $(-Q/2, Q/2]$ linearizes the staircase — the ensemble-mean output becomes a fine-stepped ramp instead of a coarse one — but the error variance still swells and shrinks with the input level (audible *noise modulation*). Triangular (TPDF) dither, the sum of two independent uniforms, makes the error variance constant as well: linearized *and* modulation-free, at a cost of about 4.77 dB of noise floor (dither power adds to quantization power). Higher-order dither buys nothing more for audio; TPDF dithering is the last step of every mastering chain.

Noise shaping. Dither fixes the error's *statistics*; noise shaping moves its *spectrum*. Isolate the requantization error and feed it back through $H(z)$ ahead of the quantizer:

$$Y(z) = X(z) + E(z)\,[1 - H(z)]$$

— the signal passes untouched while the error is filtered by the noise transfer function $1 - H(z)$. The canonical $H(z) = z^{-1}$ gives first-order highpass weighting $|1 - e^{-j\Omega}|^2$; $H(z) = z^{-1}(-2 + z^{-1})$ gives second-order $(1 - z^{-1})^2$. Total error power *rises*, but in-band power falls (the curves cross near $f_S/6$): quiet where the ear is, loud near $f_S/2$. Combined with TPDF dither this is how 16-bit media carry roughly 19-bit-quality mid-band audio — and the error-feedback loop is exactly the engine inside Lesson 38's sigma-delta converters, which run it at high oversampling so the shaped noise lands mostly out of band.

Equalizers: shelving and peak biquads. Audio EQ is built from three shapes, designed as analog prototypes and discretized by Lesson 37's bilinear transform. The recipe: (i) pick the gain $V_0 = 10^{G_{\text{dB}}/20}$ and the cutoff/center frequency ω_c, and prewarp $K = 1/\tan(\omega_c/2)$; (ii) choose the prototype — low-frequency shelving boost $H(s) = (s + V_0)/(s + 1)$, high-frequency shelving boost $H(s) = (sV_0 + 1)/(s + 1)$, and for cut *invert* the transfer function (e.g. $(s + 1)/(s + V_0)$) rather than setting $V_0 < 1$, so boost and cut curves mirror about 0 dB at the same cutoff; the *peak* filter (boost/cut at an arbitrary center) is $1 + H_{BP}(s)$:

$$H(s) = \frac{s^2 + \dfrac{V_0}{Q_\infty}\, s + 1}{s^2 + \dfrac{1}{Q_\infty}\, s + 1},$$

center gain V_0, relative bandwidth $1/Q_\infty$, response geometrically symmetric about the center; (iii) substitute $s \to K(1 - z^{-1})/(1 + z^{-1})$ and normalize — out come biquad (or first-order) coefficients ready for Lesson 37's direct forms. Production notes: *parametric* (allpass-plus-direct-path) refactorings decouple the gain, frequency, and bandwidth knobs (one coefficient each), and coefficient quantization bites worst at low cutoffs — why firmware treats bass with care.

Dynamic range control. A compressor/limiter maps input level to output level through a *static curve* in dB–dB axes: noise gate below NT, unity in the middle, compressor slope $1/R$ above threshold CT (ratio R), limiter clamp above LT. The recipe: (i) *measure level* — RMS through a one-pole, $x_{\text{rms}}[n] = (1 - \text{TAV})\, x_{\text{rms}}[n{-}1] + \text{TAV}\, x^2[n]$, or PEAK tracking of $|x[n]|$ for the limiter; (ii) work in the *log domain*: subtract the threshold from the level in dB, multiply the excess by the segment slope ($\text{CS} = 1 - 1/R$ for a compressor), antilog back to a linear control factor $f[n]$; (iii) *smooth* $f[n]$ through another one-pole whose coefficient is AT while the gain is falling (attack) and RT while rising (release); (iv) multiply the smoothed gain onto a *delayed* input $x[n - D]$, so the gain anticipates transients (look-ahead limiting). Defining attack time as the smoother's 10%–90% rise, $t_a = 2.2\tau$, the pole and coefficient are

$$z_\infty = e^{-2.2\, T_S/t_a}, \qquad k = 1 - z_\infty$$

(same formula for RT and TAV) — the one formula that turns every "attack: 5 ms" knob into a filter coefficient. Stereo processors must *link* the channels' detectors (one common gain), or the image wanders toward whichever channel is quieter.

Perceptual audio coding (stated, not derived). Information theory sets the floors. Lossless: a memoryless source can be coded at any rate above its entropy $H(X) = -\sum_i p_i \log_2 p_i$ and at no rate below it; Huffman coding achieves $\bar{L} < H + 1$, and source memory lowers the floor to a conditional entropy. Lossy: for a Gaussian source under mean-square distortion, $R(D) = \frac{1}{2}\log_2(\sigma^2/D)$ — 6.02 dB per bit again, now as a law of nature; plain uniform quantization sits about 0.25 bit above it. Perceptual coders beat these floors for *perceived* quality by spending bits where the ear listens: the cochlea analyzes in roughly 25 *critical bands*, and a strong signal *masks* its neighborhood — spreading functions applied to the band powers combine into a global masking threshold $T_m(i)$. The actionable per-band number is the *signal-to-mask ratio*, $\text{SMR}_i = L_S(i) - L_{T_m}(i)$: noise in band i is inaudible while below the mask, so the band needs only about $\lceil \text{SMR}_i/6.02 \rceil$ bits. MPEG-1 industrializes this: a 32-band polyphase (pseudo-QMF) filter bank splits the signal, a parallel FFT feeds the psychoacoustic model that computes

the SMRs, and *dynamic bit allocation* plus scalefactors pack each frame; Layer III (MP3) further splits each subband by an 18-point MDCT (a lapped transform, with window switching to catch transients) for extra coding gain.

Learned DSP: networks as trained filter banks. A multilayer feedforward network stacks layers $\mathbf{a}^{(\ell)} = h(W^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{b}^{(\ell)})$ with a differentiable activation h (sigmoid, with $h' = h(1-h)$, or ReLU). *Backpropagation* in three sentences: define $\boldsymbol{\delta}^{(\ell)} = \partial E/\partial \mathbf{z}^{(\ell)}$ for the squared error $E = \frac{1}{2}\|\mathbf{r} - \mathbf{a}^{(L)}\|^2$, so at the output $\boldsymbol{\delta}^{(L)} = h'(\mathbf{z}^{(L)}) \odot (\mathbf{a}^{(L)} - \mathbf{r})$. Each earlier layer's delta follows by the chain rule, $\boldsymbol{\delta}^{(\ell)} = h'(\mathbf{z}^{(\ell)}) \odot (W^{(\ell+1)\mathsf{T}}\boldsymbol{\delta}^{(\ell+1)})$ — the error propagates backward through the transposed weights. Every parameter then steps by $-\alpha \times \delta\times$ (its input activation), which is the exact gradient of E. A *convolutional* network replaces the dense layers' weights with small learnable kernels slid across the input: each feature map is convolution + bias + activation, i.e. a bank of FIR filters (Lesson 29) whose coefficients are *learned* rather than designed, and weight sharing encodes the shift-invariance prior while collapsing the parameter count. Pooling (neighborhood max) buys translation tolerance; each feature-map row has $N = (V+2P-F)/S+1$ neurons for input width V, padding P, kernel width F, stride S; the final maps are vectorized into an ordinary classifier. Backprop extends naturally — deltas flow backward through the same convolutions with flipped kernels, and shared weights accumulate gradient over every position. This is why Course 2's edge-ML labs run small CNNs over spectrograms (Lesson 38's STFT front end): the forward pass is a multiply-accumulate chain that fits a Cortex-M budget, while backprop trains offline on the host.

> Course 2 Labs 8.2–8.4 are this lesson's last paragraph deployed: spectrogram features feeding small CNNs for keyword classification, learned denoising, and anomaly detection. Labs 9.2–9.3 implement the classical half in real time: biquad EQ and compressor blocks (the shelving/peak recipe and the $z_\infty = e^{-2.2T_S/t_a}$ formula, straight from codec datasheets) and STFT overlap-add effects. DSP covers quantization, dither, and noise shaping as coursework; the perceptual coder is behind every stream the bench decodes.

43.2 Practice: by hand

P1 (SNR law). A full-scale 16-bit sine: $6.02\cdot 16 + 1.76 = 98.1$ dB. The same sine at -20 dBFS: 78.1 dB (the amplitude drops; Q does not). Word length for a Gaussian source at $P_F = 4.61$ to reach 96 dB: $6.02w - 8.5 \geq 96 \Rightarrow w \geq 17.4$, so 18 bits.

P2 (shaping weight). First-order shaping multiplies the error PSD by $|1 - e^{-j\Omega}|^2 = 2 - 2\cos\Omega$: 0 at $\Omega = 0$, 2 at $\pi/2$, 4 at π. It starts *amplifying* noise where the weight crosses 1, at $\Omega = \pi/3$, i.e. $f_S/6$.

P3 (shelving design). LF shelf, +12 dB ($V_0 = 4$), cutoff $0.1f_S$ ($\omega_c = 0.2\pi$): $K = 1/\tan(0.1\pi) = 3.078$. Substituting $s \to K(1-z^{-1})/(1+z^{-1})$ into $(s+V_0)/(s+1)$ and dividing by $K+1$: $b_0 = (K+V_0)/(K+1) = 1.736$, $b_1 = (-K+V_0)/(K+1) = 0.226$, $a_1 = (1-K)/(1+K) = -0.510$. Check: at $z = 1$ the gain is $(b_0+b_1)/(1+a_1) = 4.00$ (12.0 dB); at $z = -1$ it is 1 (0 dB).

P4 (compressor). Ratio $R = 4$ above CT $= -20$ dBFS: slope CS $= 1 - 1/R = 0.75$, gain $= -\text{CS}\times$ excess. Input -12 dBFS (8 dB over) $\to -18$ dBFS; -4 dBFS (16 over) $\to -16$ dBFS. Coefficients at $f_S = 48$ kHz: $t_a = 5$ ms gives AT $= 1 - e^{-2.2/240} = 9.13 \times 10^{-3}$; $t_r = 130$ ms gives RT $= 3.53 \times 10^{-4}$.

P5 (masked bit allocation). A band holds signal at 50 dB; the mask spread into it from

a neighboring 60 dB tone is 42 dB. SMR = 8 dB, so $\lceil 8/6.02 \rceil = 2$ bits keep the band's noise under the mask.

43.3 Exercises

Exercise 43.1 ([Hand].] (P&S 6.19.) A discrete memoryless source has an alphabet of eight letters x_i, $i = 1, \ldots, 8$, with probabilities 0.25, 0.20, 0.15, 0.12, 0.10, 0.08, 0.05, 0.05. (1) Use the Huffman procedure to determine a binary code for the source. (2) Determine the average number $\bar{R}$ of binary digits per source letter. (3) Determine the entropy of the source and compare it with $\bar{R}$.

Exercise 43.2 ([Hand].] (G&W 12.16.) Derive the derivatives of the following activation functions: (a) the sigmoid $h(z) = 1/(1 + e^{-z})$; (b) the hyperbolic tangent; (c) the ReLU $h(z) = \max(0, z)$.

Exercise 43.3 ([Hand].] (G&W 12.30.) A CNN takes 512×512 RGB images and has two convolutional layers, no padding. (a) The first layer's 12 feature maps are 504×504; with square, odd-sized kernels, what are the kernel dimensions? (b) With 2×2 pooling, what are the pooled feature-map dimensions? (c) How many pooled maps are there? (d) The second layer's kernels are 3×3: what are its feature-map sizes? (e) With 6 feature maps in the second layer and 2×2 pooling again, what is the length of the vectorized last layer?

Deeper reading. The Phase 6 booklet expands every paragraph here into a full lesson: Widrow's theorem via characteristic functions, dither ensemble analysis, parametric allpass EQ structures, the combination compressor/limiter system and stereo linking, the MPEG-1 polyphase bank, and the backprop/CNN chapters with gradient-check demos. Sources: Zölzer, *Digital Audio Signal Processing* (3e Ch. 11 for machine learning on audio); Proakis & Salehi Ch. 6 (entropy, Huffman, rate-distortion); Gonzalez & Woods Ch. 12 (backpropagation and deep CNNs).

44 The Electronics Bench: a Working Minimum

NEEDED FOR: Course 2 Labs 0.1–4.4

A bench lesson, not a theory lesson: the working rules, formulas, and numbers needed for the Course 2 labs, distilled from *Practical Electronics for Inventors* (PEfI).

44.1 Theory

DC circuits

Ohm's law and power. $V = IR$, and the power a component absorbs is $P = VI$; for a resistor (all power becomes heat) $P = V^2/R = I^2R$. A 100 Ω resistor across 12 V draws 120 mA and dissipates 1.44 W. Working rule: pick a power rating at least *twice* the worst-case dissipation (here a 3 W part); garden-variety resistors are rated $\frac{1}{8}$–1 W, so recompute I^2R whenever a resistance drops.

Series and parallel. Series: same current through all, resistances add, $R_{\text{tot}} = R_1 + R_2 + \cdots$. Parallel: same voltage across all, $1/R_{\text{tot}} = 1/R_1 + 1/R_2 + \cdots$; for two, $R_1 \| R_2 = R_1 R_2/(R_1 + R_2)$, always below the smaller. Branch currents V/R_k obey Kirchhoff's current law $I_{\text{in}} = I_1 + I_2 + \cdots$.

Voltage divider. Two resistors in series across V_{in} give

$$V_{\text{out}} \;=\; V_{\text{in}}\,\frac{R_2}{R_1+R_2} \qquad (R_2 \text{ is the resistor } V_{\text{out}} \text{ is taken across}).$$

The single most used formula on the bench. Caveats: the load lands in parallel with R_2, so the formula holds only when $R_{\text{load}} \gg R_2$; and a divider is *unregulated* — never use one as a power supply for a variable load.

Grounds. "Ground" means three things: earth ground (safety), chassis ground (the case, usually tied to earth), and circuit ground (the common 0 V return all node voltages are measured against). Keep analog and digital circuit grounds separate and join them at one point near the supply, or digital switching currents will pollute analog measurements.

Open and short circuits. The two standard faults: an open (broken wire, burnt part) stops current entirely; a short (solder splash, crossed leads) draws excessive current, limited only by the source's internal resistance, and blows fuses or parts. Diagnose shorts by heat and smell, opens with the ohmmeter or by walking the voltage along the circuit.

Measuring V, I, R. Voltmeter: in *parallel* with the element; ideal input resistance infinite (real DMMs $\sim 10\,\text{M}\Omega$). Ammeter: break the circuit and insert in *series*; ideally zero resistance. Ohmmeter: only on an *unpowered* circuit. Loading errors grow as the circuit's Thevenin resistance approaches the meter's; a voltmeter should present at least $\sim 20\times$ the resistance it measures across.

AC and RC

Describing an AC waveform. A sinusoid is fixed by amplitude, frequency, and phase: $v(t) = V_P \sin(2\pi f t)$, with $f = 1/T$ (Hz) and phase differences quoted in degrees (90° = quarter period). US mains: 60 Hz, $T = 16.7$ ms.

RMS. The plain average of a sine is zero, so AC voltages are quoted as root-mean-square values, the DC-equivalent for heating a resistor:

$$V_{\text{rms}} = \frac{V_P}{\sqrt{2}} = 0.707\,V_P, \qquad I_{\text{rms}} = \frac{I_P}{\sqrt{2}}, \qquad P = I_{\text{rms}}V_{\text{rms}} = \frac{V_{\text{rms}}^2}{R}.$$

"120 VAC" mains is RMS: the peak is 170 V. Unlabeled AC ratings are RMS by convention, and Ohm's law works unchanged with RMS values.

Capacitors. $C = Q/V$ (farads); practical parts run 1 pF–4700 μF. Combination rules are the *reverse* of resistors: parallel capacitances add, series combine by reciprocals. Stored energy: $E = \frac{1}{2}CV^2$.

RC time constant. Charging a capacitor through a resistor from a step V_S:

$$V_C(t) = V_S\big(1 - e^{-t/RC}\big), \qquad \tau = RC.$$

After one time constant V_C reaches 63.2% of V_S; after 3τ, 95%; after 5τ it is "fully charged" (99.3%). Discharge is the mirror image, $V_C = V_S e^{-t/RC}$. This one curve runs every timer, debouncer, and reset circuit you will build.

Reactance. A capacitor passes AC in proportion to frequency; its opposition is

$$X_C = \frac{1}{2\pi f C} \quad (\Omega).$$

At DC it is an open circuit; as $f \to \infty$ it is a short; no power is dissipated in reactance. This frequency-dependent "resistance" is the whole basis of RC filtering and supply decoupling.

Real components

Wire. Gauge number: *smaller* gauge = fatter wire = more current. Use 22-gauge *solid-core* hookup wire on breadboards; stranded for anything that flexes.

Switches. Classified by poles (independent circuits switched) and throws (positions per pole): SPST is on/off, SPDT selects one of two contacts, DPDT switches two circuits at once; pushbuttons are momentary-contact. Mechanical contacts bounce — expect it when a switch feeds a digital input.

Real resistors. Axial parts use color bands: digit–digit–multiplier–tolerance (no fourth band means 20%). Carbon film runs 1–5% tolerance, metal film $\sim$ 1% and stabler; surface-mount parts carry 3- or 4-digit codes. A "100 Ω, 10%" part is anywhere from 90 to 110 Ω — check tolerance before blaming your divider.

Real capacitors. Ceramics: small, non-polarized, poor tolerance, low series inductance — right for high-frequency decoupling. Electrolytics (aluminum, tantalum): large capacitance per volume, but *polarized* (reverse them and they fail) and leaky, with significant equivalent series resistance (ESR); a real capacitor behaves like C + ESR + series inductance, and looks inductive above self-resonance. Films: precise and stable, for filters and audio. Respect the working-voltage rating.

Decoupling and ripple. Place a 0.01–0.1 μF ceramic directly at each IC's supply pin (shortest leads) plus one bulk electrolytic per rail: the ceramic bypasses high-frequency transients that the electrolytic, with its ESR and inductance, cannot. In a rectifier supply a reservoir capacitor smooths the pulsating DC into DC plus sawtooth ripple; bigger C and lighter load mean less ripple (PEfI: 1000 μF full-wave into 100 Ω leaves 0.34 V RMS; half-wave doubles it).

Instruments and practice

Safety. Current, not voltage, does the damage: above $\sim$ 10 mA muscle control goes, and 100 mA–1 A across the chest is the deadliest range. Keep one hand in your pocket when probing anything mains-connected; power down before rewiring; discharge large capacitors through a resistor (then verify with a voltmeter) — big electrolytics hold charge for days.

ESD. Ordinary motion puts hundreds to thousands of volts of static on your body — enough to punch through a MOSFET gate oxide invisibly. MOS transistors, MOS ICs, and JFETs are most vulnerable: use a grounded wrist strap and antistatic mat, keep parts in their antistatic bags, and touch grounded metal before handling a CMOS chip.

Building and troubleshooting. Prototype on a solderless breadboard (0.1-inch socket grid, power buses along the edges); commit to perfboard or a PCB later. Solder with a 25–40 W iron and rosin-core solder, keep the tip tinned, heat the joint — not the solder — and use desoldering braid for rework. Troubleshoot in order: visual inspection, then supply rails at every IC, then walk the signal stage by stage with the scope.

Multimeter. Placement rules as above; remember that a 10 MΩ voltmeter across a 1 MΩ divider node visibly drags the reading, and that cheap meters report "RMS" correct only for sinusoids (a true-RMS meter handles other waveforms).

Oscilloscope. The scope measures *voltage only*, versus time, into a 1 MΩ $\|\sim$ 20 pF input. The standard 10× passive probe adds 9 MΩ in series, a 10 : 1 divider: the circuit sees a gentle 10 MΩ load and the scope multiplies readings by 10. *Compensate* the probe before trusting it: clip to the scope's square-wave reference and trim the probe capacitor until the corners are square (overshoot = overcompensated, rounded = under). Keep the ground lead short — a long

loop rings.

Bench kit. An adjustable current-limited DC supply (outputs usually float), a two-channel scope, and a function generator (sine/square/triangle, DC offset). Trap: the BNC sleeves of scope and generator are typically tied to earth ground — clipping the scope's ground lead to a non-ground node of an earth-referenced circuit is a short.

Op-amps

What it is. A high-gain differential amplifier: $V_{\text{out}} = A_o\,(V_+ - V_-)$ with open-loop gain $A_o \sim 10^4$–10^6. With no feedback the output slams to a rail — that is the comparator.

Golden rules (valid with negative feedback): (I) the inputs draw no current; (II) the output does whatever forces $V_+ = V_-$. Every closed-loop formula below is a two-line consequence.

The three feedback amplifiers.

$$\text{follower: } V_{\text{out}} = V_{\text{in}}; \qquad \text{non-inverting: } \frac{V_{\text{out}}}{V_{\text{in}}} = 1 + \frac{R_2}{R_1}; \qquad \text{inverting: } \frac{V_{\text{out}}}{V_{\text{in}}} = -\frac{R_F}{R_{\text{in}}}.$$

The follower is a buffer: huge input impedance, tiny output impedance — put one between a high-impedance source (sensor, divider tap) and whatever loads it. In the inverting amplifier the feedback holds V_- at 0 V (a *virtual ground*), so its input impedance is just R_{in}.

Real op-amps. Finite gain–bandwidth product (closed-loop gain × bandwidth ≈ constant), finite slew rate (V/μs), input offset voltage, and input bias currents (bipolar inputs leak most, FET-input parts least). The output cannot swing past the rails, and the *inputs* must never exceed the rails either, or the part can latch up. Bypass both supply pins with 0.1 μF ceramics to ground.

Single-supply operation. With only $+V_S$ and ground (the MCU-board situation), bias the noninverting input at $V_S/2$ with an equal-resistor divider and AC-couple the signal in and out: the op-amp then amplifies about a mid-rail "virtual zero" with maximum symmetric swing. Stay inside the rated common-mode input range.

Filters

First-order RC. Series R, then C to ground, output across C: a low-pass with cutoff

$$f_c = \frac{1}{2\pi RC}, \qquad \left|\frac{V_{\text{out}}}{V_{\text{in}}}\right| = \frac{1}{\sqrt{1 + (f/f_c)^2}}.$$

At f_c the output is -3 dB (half power, 0.707×); beyond it the response falls 6 dB/octave (20 dB/decade): the Bode picture is a flat shelf meeting a -20 dB/dec ramp at f_c. Swap R and C for the matching high-pass.

Sharper filters. One pole is shallow. For a spec "-3 dB at $f_{3\text{dB}}$, at least $-A$ dB at f_s," compute the steepness factor $A_s = f_s/f_{3\text{dB}}$ and read the required order n off normalized response curves — Butterworth (maximally flat), Chebyshev (steeper, passband ripple), or Bessel (best pulse behavior) — then scale normalized (1 rad/s, 1 Ω) tables in frequency and impedance: recipe, not algebra.

Sallen–Key active low-pass. An op-amp follower with two equal resistors and two capacitors makes a unity-gain *two-pole* section (a three-pole variant adds one RC); cascading sections builds any order without inductors, and the op-amp buffers the load. The normalized

Butterworth two-pole section uses $C_1 = 1.414\,\mathrm{F}$, $C_2 = 0.7071\,\mathrm{F}$ with $1\,\Omega$ resistors; at a practical impedance level Z (e.g. $10\,\mathrm{k}\Omega$ resistors) the real capacitors are $C_{\text{actual}} = C_{\text{table}}/(2\pi f_{3\mathrm{dB}} Z)$. Above roughly 100 kHz the op-amp's own bandwidth intrudes; go passive there.

Every item above is a Course 2 lab. Lab 1.1: compensate your 10× scope probe on the calibrator square wave before any measurement. Lab 1.3: build the passive RC low-pass and verify $f_c = 1/(2\pi RC)$ with a function-generator Bode sweep. Modules 0–3 live on the divider, decoupling, DMM, and breadboard rules (mixed-signal boards fail from missing $0.1\,\mu\mathrm{F}$ ceramics more than anything else). Module 4 uses the MCP6002, a single-supply op-amp: Labs 4.1–4.3 build the follower and the non-inverting amplifier $(1 + R_2/R_1)$ biased at $V_S/2$, and Lab 4.4 puts a Sallen–Key low-pass before the ADC — the anti-aliasing filter Lesson 33 demands.

44.2 Practice: by hand

Example 44.1 (Divider for a 5 V reference)**.** A chip input needs 5 V from a 9 V supply and draws essentially no current. Choose $R_2 = 10\,\mathrm{k}\Omega$; solving the divider for R_1: $R_1 = R_2\,(V_{\text{in}} - V_{\text{out}})/V_{\text{out}} = 10\,\mathrm{k}\Omega \cdot 4/5 = 8\,\mathrm{k}\Omega$. The divider's standing (bleeder) current is $V_{\text{out}}/R_2 = 5\,\mathrm{V}/10\,\mathrm{k}\Omega = 0.5\,\mathrm{mA}$ — small enough to waste, large enough to swamp the chip's input current.

Example 44.2 (RC timing network)**.** A timer IC triggers when its input, charged through R from a 5 V supply into $C = 10\,\mu\mathrm{F}$, reaches 3.4 V. For a 5 s delay, invert the charging law $t/RC = -\ln\big((V_S - V_C)/V_S\big)$:

$$R = \frac{5.0\,\mathrm{s}}{-\ln\!\left(\frac{5-3.4}{5}\right)\cdot 10^{-5}\,\mathrm{F}} = \frac{5.0}{1.139\cdot 10^{-5}} \approx 4.4\times 10^{5}\,\Omega \quad (\text{use } 439\,\mathrm{k}\Omega).$$

Example 44.3 (Reactance and RMS numbers)**.** (a) A 220 pF capacitor at 10 MHz: $X_C = 1/(2\pi\cdot 10^7 \cdot 2.2\times 10^{-10}) = 72.3\,\Omega$. (b) US mains "120 VAC" is RMS: $V_P = 120\sqrt{2} = 170\,\mathrm{V}$, and a $100\,\Omega$ load dissipates $P = V_{\text{rms}}^2/R = 144\,\mathrm{W}$ — RMS plugs straight into the DC formulas.

44.3 Exercises

Exercise 44.1 ([Hand)**.**] A $1\,\mathrm{k}\Omega$ and a $3\,\mathrm{k}\Omega$ resistor sit in parallel across a 12 V battery. Find the equivalent resistance, the current through each resistor, the total current, and the total power dissipated. Check Kirchhoff's current law and check that $P_{\text{tot}} = V^2/R_{\text{tot}}$.

Exercise 44.2 ([Hand)**.**] Find the reactance of a 470 pF capacitor at 7.5 MHz and at 15 MHz. What happens to X_C when the frequency doubles, and why does that make a shunt capacitor act as a low-pass element?

Exercise 44.3 ([Hand)**.**] An anti-aliasing filter must be $-3\,\mathrm{dB}$ at 100 Hz and at least $-60\,\mathrm{dB}$ at 400 Hz. (a) Compute the steepness factor $A_s = f_s/f_{3\mathrm{dB}}$; the Butterworth curves then demand a five-pole filter — how would you realize it from Sallen–Key sections? (b) Scale a normalized unity-gain two-pole Butterworth section ($C_1 = 1.414\,\mathrm{F}$, $C_2 = 0.7071\,\mathrm{F}$, $1\,\Omega$ resistors) to $f_{3\mathrm{dB}} = 1\,\mathrm{kHz}$ with $Z = 10\,\mathrm{k}\Omega$, using $C_{\text{actual}} = C_{\text{table}}/(2\pi f_{3\mathrm{dB}} Z)$.

Deeper reading. Condensed from Scherz & Monk, *Practical Electronics for Inventors*, 4th ed.: Ch. 2 (circuit theory, RC, reactance), Ch. 3 (real components), Ch. 7 (safety, construction, instruments), Ch. 8 (op-amps), Ch. 9 (filter tables and Sallen–Key). The frequency-domain view of these filters is Lessons 31–34; the sampling argument behind Lab 4.4's anti-aliasing filter is Lesson 33.

Part VII: The Analysis Behind the Transforms

Part V computed freely: it differentiated a square wave, integrated δ's, swapped $\int$ with $\sum$ and $\int$ with $\int$, evaluated transforms by table and partial fractions, and promised each time that the move "can be certified." This part is the certificate. It answers, with the bare minimum of real analysis, measure theory, functional analysis, complex analysis, distribution theory, and numerical linear algebra, the six questions Part V kept deferring:

1. When may a limit pass through a sum or an integral? (Lesson 45 — and why the Gibbs overshoot is a theorem, not a numerical artifact.)
2. What integral makes "energy," "almost everywhere," and every double-integral swap honest? (Lesson 46.)
3. What space do signals live in, and in what sense does a Fourier series actually converge? (Lesson 47.)
4. Why do poles, ROCs, partial fractions, and inversion integrals work? (Lesson 48.)
5. What *is* $\delta(t)$, and why may we differentiate discontinuous signals and Fourier-transform constants and impulse trains? (Lesson 49.)
6. What is a derivative with respect to a *function*, and where do matched filters and regularized deconvolution come from? (Lesson 50.)

Lesson 51 adds the compute layer — conditioning, stability, and the structured-matrix facts that make transform mathematics runnable — and the capstone exercises all of it numerically. Sources, one per lesson, all self-contained here at working depth: Ross, *Elementary Analysis* (2nd ed.); Axler, *Measure, Integration & Real Analysis* (MIRA); Bak–Newman, *Complex Analysis* (3rd ed.); Strichartz, *A Guide to Distribution Theory and Fourier Transforms*; Bowers–Kalton, *An Introductory Course in Functional Analysis*; Trefethen–Bau, *Numerical Linear Algebra*. The ground rule throughout is *bare minimum, honestly connected*: a proof appears when it teaches a mechanism you will reuse; a theorem is stated with a pointer when its proof is a course of its own. *Nothing here is needed for machine learning*, and Fourier analysis and statistical inference can formally be survived without it — but graduate Fourier analysis develops exactly Lesson 49, statistical inference's "almost surely" is exactly Lesson 46's "almost everywhere," and graduate DSP (estimation, adaptive filters, wavelets, multirate) speaks the language of Lessons 38–42 as its native tongue. Part VII keeps Part V's convention j for the imaginary unit, even inside complex analysis.

45 Limits Done Right: Sequences, Series, and When Swapping Is Legal

NEEDED FOR: Optional rigor layer for Fourier analysis and statistical inference; foundation for graduate DSP. Not needed for ML.

45.1 Theory

Everything in analysis rests on one axiom that distinguishes $\mathbb{R}$ from the rationals.

Definition 45.1 (Completeness axiom)**.** Every nonempty set $S \subseteq \mathbb{R}$ that is bounded above has a *least* upper bound $\sup S \in \mathbb{R}$.

Definition 45.2 (Limit of a sequence)**.** $s_n \to s$ means: for every $\varepsilon > 0$ there is N such that $|s_n - s| < \varepsilon$ for all $n > N$. A sequence is *Cauchy* if for every $\varepsilon > 0$ there is N with $|s_n - s_m| < \varepsilon$ for all $n, m > N$.

Theorem 45.3 (Monotone and Cauchy convergence)**.** *(a) Every bounded monotone sequence converges. (b) A sequence of reals converges iff it is Cauchy.*

Proof. (a) Say $s_1 \le s_2 \le \cdots \le B$. Let $s = \sup_n s_n$ (completeness). Given $\varepsilon > 0$, $s - \varepsilon$ is not an upper bound, so some $s_N > s - \varepsilon$; by monotonicity $s - \varepsilon < s_n \le s$ for all $n > N$. (b) Convergent $\Rightarrow$ Cauchy is the triangle inequality. Conversely a Cauchy sequence is bounded, so $t_n = \sup_{m \ge n} s_m$ is a bounded decreasing sequence; by (a) it converges, say to s, and the Cauchy condition squeezes $s_n \to s$ as well (Ross §10 fills in the two-line estimate). □

Part (b) is the tool that proves convergence *without knowing the limit* — the only way to handle series whose sums have no closed form.

Definition 45.4 (Series)**.** $\sum_{k=1}^{\infty} a_k = \lim_{n\to\infty} \sum_{k=1}^{n} a_k$ when the limit of partial sums exists. The series *converges absolutely* if $\sum |a_k| < \infty$.

Proposition 45.5 (The three workhorses)**.** *(a) (Geometric) $\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}$ for $|r| < 1$, and diverges for $|r| \ge 1$.*

(b) (Comparison) If $|a_k| \le b_k$ and $\sum b_k < \infty$, then $\sum a_k$ converges, absolutely.

(c) Absolute convergence implies convergence.

Proof. (a) $(1 - r)\sum_{k=0}^{n} r^k = 1 - r^{n+1}$ and $r^{n+1} \to 0$ iff $|r| < 1$. (c) Partial sums of $\sum a_k$ are Cauchy because $\left|\sum_{k=m}^{n} a_k\right| \le \sum_{k=m}^{n} |a_k|$, and the right side is the Cauchy tail of a convergent series; apply Theorem 45.3(b). (b) The same estimate, with b_k as majorant, plus (c). □

Theorem 45.6 (Order matters, unless it doesn't)**.** *If $\sum a_k$ converges absolutely, every rearrangement converges to the same sum. If it converges only conditionally (like $1 - \frac{1}{2} + \frac{1}{3} - \cdots = \ln 2$), then for* any *target $t \in \mathbb{R} \cup \{\pm\infty\}$ some rearrangement converges to t (Riemann). (Proof: Ross §22 / Rudin; the mechanism — greedily overshoot with positive terms, undershoot with negative — is worth one minute of thought.)*

This is why Part V's hypotheses are what they are. Theorem 29.3 demanded $\sum_k |h[k]| < \infty$, and Proposition 29.2 freely reindexed the double sum $\sum_k \sum_m x[k]h_1[m]h_2[n-k-m]$: absolute convergence is precisely the license to reorder and regroup infinite sums as if they were finite. Without it, Theorem 45.6 says the "sum" is not even well defined.

From numbers to functions. A signal-processing limit is almost never a limit of numbers; it is a limit of *functions* (partial Fourier sums, truncated impulse responses, mollified pulses). Two inequivalent meanings:

Definition 45.7 (Pointwise and uniform convergence)**.** $f_n \to f$ *pointwise* on S if $f_n(t) \to f(t)$ for each fixed $t \in S$; *uniformly* if

$$\|f_n - f\|_\infty = \sup_{t \in S} |f_n(t) - f(t)| \longrightarrow 0,$$

i.e. one N works for every t simultaneously. (Note the sup-norm: uniform convergence is convergence in a norm on function space — Lesson 47 makes that view official.)

Example 45.8. $f_n(t) = t^n$ on $[0,1]$ converges pointwise to $f(t) = 0$ for $t < 1$, $f(1) = 1$ — but not uniformly: $\sup_t |f_n(t) - f(t)| = 1$ for every n (take t just below 1). The limit of these continuous functions is discontinuous.

Theorem 45.9 (Uniform limits preserve continuity)**.** *If each f_n is continuous on S and $f_n \to f$ uniformly, then f is continuous.*

Proof. The $\varepsilon/3$ argument. Fix t_0 and $\varepsilon > 0$. Choose n with $\|f_n - f\|_\infty < \varepsilon/3$, then $\eta > 0$ with $|f_n(t) - f_n(t_0)| < \varepsilon/3$ for $|t - t_0| < \eta$ (continuity of f_n). Then

$$|f(t) - f(t_0)| \le \underbrace{|f(t) - f_n(t)|}_{<\varepsilon/3} + \underbrace{|f_n(t) - f_n(t_0)|}_{<\varepsilon/3} + \underbrace{|f_n(t_0) - f(t_0)|}_{<\varepsilon/3} < \varepsilon. \qquad \square$$

Corollary 45.10 (Gibbs is inevitable)**.** *The partial Fourier sums of the square wave are continuous (finite sums of cosines); the square wave is not. Therefore the convergence* cannot *be uniform: by Theorem 45.9,* $\sup_t |S_N(t) - x(t)| \not\to 0$*. The overshoot of Lesson 30's partial sums, stalled near 9% of the jump, is this corollary made visible (its exact value,* $\frac{1}{\pi}\int_0^\pi \frac{\sin u}{u}\,du - \frac{1}{2} \approx 0.0895$*, is computed in Lesson 52). No amount of terms, and no amount of floating-point care, removes it.*

Theorem 45.11 (Swapping limit and integral: the uniform case)**.** *If $f_n \to f$ uniformly on a* bounded *interval $[a,b]$ and each f_n is integrable, then f is integrable and $\int_a^b f_n \to \int_a^b f$.*

Proof. $\left|\int_a^b f_n - \int_a^b f\right| \le \int_a^b |f_n - f| \le (b-a)\,\|f_n - f\|_\infty \to 0$. (Integrability of f: Ross §25.) $\square$

Example 45.12 (Both hypotheses are load-bearing)**.** *Bounded interval fails:* $f_n = \mathbf{1}_{[n,\,n+1]}$ (a bump marching to $+\infty$) converges to 0 pointwise, and even "locally uniformly," yet $\int_{\mathbb{R}} f_n = 1 \not\to 0$. *Uniformity fails:* $f_n = n \cdot \mathbf{1}_{[0,\,1/n]}$ converges to 0 pointwise on $[0,1]$ but $\int_0^1 f_n = 1$. Mass escaping to infinity or into a spike is exactly what the hypotheses forbid — keep both pictures; Lesson 46 tames them with domination instead of uniformity, and Lesson 49 will make the second one *converge after all*, to δ, in a weaker sense.

Theorem 45.13 (Weierstrass M-test)**.** *If $\|g_k\|_\infty \le M_k$ on S and $\sum_k M_k < \infty$, then $\sum_k g_k$ converges uniformly on S (and absolutely at each point).*

Proof. The partial sums are uniformly Cauchy: $\|\sum_{k=m}^n g_k\|_\infty \le \sum_{k=m}^n M_k$, a tail of a convergent series. Apply Theorem 45.3(b) at each t, and note the resulting pointwise limit is approached at the uniform rate $\sum_{k>n} M_k$. $\square$

Corollary 45.14 (Absolutely summable Fourier series behave perfectly)**.** *If $\sum_k |a_k| < \infty$, then $x(t) = \sum_k a_k e^{jk\omega_0 t}$ converges uniformly, x is continuous, and term-by-term integration against anything bounded is legal (Theorem 45.11) — in particular the coefficient formula $a_l = \frac{1}{T}\int_T x(t)e^{-jl\omega_0 t}dt$ of Theorem 30.3 is an honest computation, not a formal one: swap $\int_T \sum_k$ to $\sum_k \int_T$ and invoke Lemma 30.2.*

By the smoothness–decay law (Exercise 30.3), $\sum |a_k| < \infty$ holds whenever x is continuously differentiable ($|a_k| = O(1/k^2)$); the square wave, with its $|a_k| \sim 1/k$, sits precisely on the wrong side of this test — which is Gibbs again, from the coefficient side.

Theorem 45.15 (Term-by-term differentiation)**.** *If $\sum_k g_k$ converges at one point and the* differentiated *series $\sum_k g_k'$ converges uniformly on an interval, then $\sum g_k$ converges uniformly there, its sum is differentiable, and $\big(\sum g_k\big)' = \sum g_k'$. (Proof: Ross §26.)*

Note where the hypothesis sits: on the *derivatives*. $\sum \sin(kt)/k^2$ converges uniformly (M-test, $M_k = 1/k^2$) but its differentiated series $\sum \cos(kt)/k$ does not converge absolutely — one uniform-convergence hypothesis per derivative you take. That is why each degree of smoothness of x costs one power of decay of a_k, and it is the honest content of "differentiate the Fourier series of the triangle wave to get the square wave." (For genuinely discontinuous signals, differentiation must wait for Lesson 49, where it becomes *always* legal — at the price of the answer being a distribution.)

45.2 Practice: by hand

Example 45.16 (The four-example zoo, worth memorizing)**.** On the stated sets, with $f = \lim f_n$ pointwise:

f_n	set	uniform?	moral
t^n	$[0,1]$	$\times$	uniform limit would be continuous
$\frac{\sin(nt)}{\sqrt{n}}$	$\mathbb{R}$	✓	$f_n'(0) = \sqrt{n} \to \infty$: derivatives need their own test
$n\,\mathbf{1}_{[0,1/n]}$	$[0,1]$	$\times$	spike: $\int f_n = 1 \neq 0 = \int f$
$\mathbf{1}_{[n,n+1]}$	$\mathbb{R}$	$\times$	escape to ∞: same integral failure

The second row is the sharpest: $\|f_n - 0\|_\infty = 1/\sqrt{n} \to 0$, uniform convergence at its best, yet the derivatives diverge — convergence of f_n never controls f_n', exactly as Theorem 45.15's hypothesis warns.

Example 45.17 (A stability margin from a geometric tail)**.** For $h[n] = a^n u[n]$, $|a| < 1$, the truncation error of the impulse response is $\sum_{k>N} |a|^k = \frac{|a|^{N+1}}{1-|a|}$ (geometric tail). So an FIR truncation of this IIR filter is within ε of the true system — in the operator sense of Lesson 47 — once $N > \log\big(\varepsilon(1-|a|)\big)/\log|a|$. For $a = 0.9$, $\varepsilon = 10^{-6}$: $N \geq 153$. This one-line tail estimate is the entire theory behind "truncate the impulse response" in filter design.

Fourier / statistical-inference / grad DSP connection. Fourier analysis states its convergence theorems in exactly this vocabulary (pointwise vs. uniform vs. the L^2 sense of Lesson 47), and Gibbs (Corollary 45.10) is a lecture there. Statistical inference's limit theorems (Lesson 24) juggle three modes of convergence of random variables; the pointwise/uniform/norm trichotomy here is the deterministic warm-up — and Theorem 45.3 is

what "the empirical average converges" silently uses. In graduate DSP, every adaptive algorithm's convergence proof is a sequence argument, and every filter-truncation error bound is a tail estimate like the one above.

45.3 Exercises

Exercise 45.1 (Hand)**.** For each family decide pointwise limit and uniformity, with a one-line sup computation: (a) $f_n(t) = \frac{t}{1+nt^2}$ on $\mathbb{R}$; (b) $f_n(t) = \frac{nt}{1+nt}$ on $[0,1]$; (c) $f_n(t) = t^n(1-t)$ on $[0,1]$. (Answers: (a) uniform, sup $= \frac{1}{2\sqrt{n}}$; (b) not uniform — limit jumps at 0; (c) uniform — maximize to get sup $\sim \frac{1}{en}$.)

Exercise 45.2 (Hand)**.** How many terms of $\sum_k 1/k^2$ guarantee the partial sum is within 10^{-3} of $\pi^2/6$? (Bound the tail by $\int_N^\infty t^{-2}dt$.) Compare with the alternating series $\sum(-1)^{k+1}/k^2$, where the error is at most the first omitted term — conditional-style cancellation converges faster, but Theorem 45.6 says you must not reorder it carelessly.

Exercise 45.3 (Proof, $\star$)**.** Prove Theorem 45.13 in full (uniformly Cauchy $\Rightarrow$ uniformly convergent included), then use it to show that $x(t) = \sum_{k=1}^{\infty} 2^{-k}\cos(3^k t)$ is continuous on all of $\mathbb{R}$. (This is Weierstrass's famous example; it is differentiable *nowhere* — continuity from Theorem 45.9 costs nothing, differentiability is a different currency, Theorem 45.15.)

Exercise 45.4 (Proof)**.** Show that if $f_n \to f$ uniformly on $\mathbb{R}$ and each f_n is bounded, then f is bounded — and conclude (contrapositive) that $\frac{1}{t}\mathbf{1}_{(0,1)}$ is not a uniform limit of bounded functions. Then verify that the convergence $\frac{t}{1+nt^2} \to 0$ *is* uniform although each function is bounded by a different constant.

Deeper reading. Ross §§4, 7–12 (completeness, sequences), 14–15 (series), 23–26 (sequences and series of functions, uniform convergence, term-by-term theorems).

46 Integration Done Right: Lebesgue, Almost Everywhere, and the Swap Theorems

NEEDED FOR: Optional rigor layer for Fourier analysis and statistical inference; the language of graduate probability and DSP. Not needed for ML.

46.1 Theory

Lesson 45 bought $\lim \int = \int \lim$ with uniform convergence on bounded intervals. Part V needs far more: transforms integrate over all of $\mathbb{R}$, and the sequences that matter (partial inversions, truncated spectra) converge anything but uniformly. The Riemann integral cannot deliver:

Example 46.1 (Riemann's ceiling)**.** Enumerate the rationals in $[0,1]$ as $r_1, r_2, \ldots$ and let $f_n = \mathbf{1}_{\{r_1,\ldots,r_n\}}$. Each f_n is Riemann integrable (finitely many spikes, integral 0), f_n increases pointwise to $f = \mathbf{1}_{\mathbb{Q}\cap[0,1]}$ — and f is not Riemann integrable at all: every subinterval contains rationals and irrationals, so all upper sums are 1 and all lower sums are 0. The class of Riemann integrable functions is not closed under the limits we care about.

Lebesgue's fix has two ingredients: measure the *sets* first, and integrate by slicing the *range* instead of the domain.

Definition 46.2 (Measure zero; almost everywhere)**.** $E \subseteq \mathbb{R}$ has *measure zero* if for every $\varepsilon > 0$ it can be covered by countably many intervals of total length $< \varepsilon$. A property holds *almost everywhere* (a.e.) if it holds off a set of measure zero. Every countable set is null (cover r_k by an interval of length $\varepsilon 2^{-k}$; total $\leq \varepsilon$).

Definition 46.3 (Lebesgue measure and the integral, in three strokes)**.** (1) The *outer measure* of $E \subseteq \mathbb{R}$ is $|E| = \inf\left\{ \sum_k \ell(I_k) : E \subseteq \bigcup_k I_k \right\}$, the cheapest countable interval cover. Restricted to the (enormous) class of *measurable* sets, $|\cdot|$ is *countably additive*: $|\bigcup_k E_k| = \sum_k |E_k|$ for disjoint measurable E_k — the property that powers everything below (MIRA ch. 2; every set you can write down is measurable). (2) A function is *measurable* if preimages of intervals are measurable sets; limits of measurable functions are measurable — the closure Riemann lacked. (3) For $f \geq 0$,

$$\int f = \sup\left\{ \textstyle\sum_i c_i\, |E_i| : \sum_i c_i \mathbf{1}_{E_i} \leq f \right\},$$

the supremum over *simple* (finite-valued) minorants: slice the range into levels c_i, measure the level sets E_i. For general f, write $f = f^+ - f^-$ and set $\int f = \int f^+ - \int f^-$ when $\int |f| < \infty$; such f are called *(Lebesgue) integrable*, written $f \in L^1$. When the Riemann integral exists, the two integrals agree (MIRA 3B) — nothing you computed in Part V changes value.

Two immediate dividends. First, $\mathbf{1}_{\mathbb{Q}\cap[0,1]}$ integrates to 0 (its level set is null): Example 46.1's ceiling is gone. Second:

Proposition 46.4 (Null sets are invisible)**.** *If $f = g$ a.e. then $\int f = \int g$ (over any set), so f and g have the same energy, the same Fourier transform, the same everything expressible by integrals. Conversely if $f \geq 0$ and $\int f = 0$ then $f = 0$ a.e.*

Proof. $f - g$ vanishes off a null set E; every simple minorant of $|f - g|$ is supported in E and so has integral $\leq \|\text{minorant}\|_\infty \cdot |E| = 0$. For the converse: the set $\{f > 1/n\}$ has measure zero (else the simple function $\frac{1}{n}\mathbf{1}_{\{f>1/n\}}$ would witness $\int f > 0$), and $\{f > 0\} = \bigcup_n \{f > 1/n\}$ is a countable union of null sets, hence null. □

This is why "the" inverse Fourier transform is only defined up to a.e. equality, and why nobody cares what a reconstructed signal does at the single point of a jump — Theorem 30.3's midpoint convention at discontinuities is a choice of representative, not a fact about the signal.

Theorem 46.5 (Monotone Convergence Theorem, MCT)**.** *If $0 \leq f_1 \leq f_2 \leq \cdots$ and $f_n \to f$ pointwise (a.e. suffices), then $\int f_n \to \int f$, finite or infinite. (Proof: MIRA 3A; one page from countable additivity.)*

Theorem 46.6 (Dominated Convergence Theorem, DCT — the workhorse)**.** *If $f_n \to f$ a.e. and there is a single $g \in L^1$ with $|f_n| \leq g$ a.e. for all n, then $f \in L^1$ and*

$$\lim_n \int f_n = \int f.$$

(Proof from MCT via Fatou's lemma: MIRA 3B.)

The hypothesis is exactly calibrated to Example 45.12: the marching bumps $\mathbf{1}_{[n,n+1]}$ have no integrable dominator (their sup-envelope is $\mathbf{1}_{[1,\infty)}$, integral ∞), and neither do the spikes $n\mathbf{1}_{[0,1/n]}$ (envelope $\sim 1/t$). One integrable roof over the whole family forbids both escape routes — no uniformity, no bounded interval required.

Corollary 46.7 (The transform-side dividends)**.** *Let $x \in L^1(\mathbb{R})$, $X(j\omega) = \int x(t)e^{-j\omega t}dt$.*

(a) X is continuous, and $|X(j\omega)| \le \|x\|_1$ everywhere.

(b) If also $t\,x(t) \in L^1$, then X is differentiable with $\frac{dX}{d\omega} = \int(-jt)x(t)e^{-j\omega t}dt$: differentiation under the integral sign — Part V's "multiply by t $\leftrightarrow$ differentiate in ω" rule (dual of Theorem 31.3), now a theorem.

(c) The same argument gives: the Laplace transform $X(s)$ of Proposition 30.1 is differentiable in s — in the complex sense of Lesson 48 — at every interior point of its ROC.

Proof. (a) For $\omega_n \to \omega$, the integrands $x(t)e^{-j\omega_n t}$ converge pointwise and are all dominated by $|x| \in L^1$: DCT. The bound is the triangle inequality for integrals. (b) Difference quotients:

$$\frac{X(j(\omega+h)) - X(j\omega)}{h} = \int x(t)e^{-j\omega t}\,\frac{e^{-jht}-1}{h}\,dt.$$

The quotient $\frac{e^{-jht}-1}{h}$ tends to $-jt$ pointwise and is bounded in modulus by $|t|$ (the chord of the unit circle is shorter than the arc), so the integrands are dominated by $|t\,x(t)| \in L^1$: DCT again. (c) Identical, with e^{-st}; the domination $|t|e^{-\sigma t}|x(t)|$ is integrable slightly inside the ROC, which is why the statement is for interior points. □

Theorem 46.8 (Tonelli–Fubini: the double-integral license)**.** *For measurable $f(t,\tau)$ on $\mathbb{R}^2$: (Tonelli) if $f \ge 0$, the iterated integrals in either order and the double integral are all equal (possibly $+\infty$), unconditionally; (Fubini) if $\iint |f| < \infty$ — checkable by Tonelli — then the same equalities hold for f itself. (MIRA ch. 5.)*

Corollary 46.9 (Convolution certified)**.** *If $x, h \in L^1(\mathbb{R})$, then $(x*h)(t) = \int x(\tau)h(t-\tau)d\tau$ is defined for a.e. t, lies in L^1 with $\|x*h\|_1 \le \|x\|_1\|h\|_1$, and the convolution property (Theorem 31.4) holds honestly: $\widehat{x*h} = X(j\omega)H(j\omega)$.*

Proof. Tonelli on $|x(\tau)||h(t-\tau)| \ge 0$:

$$\iint |x(\tau)||h(t-\tau)|\,d\tau\,dt = \int |x(\tau)|\Big(\int |h(t-\tau)|dt\Big)d\tau = \|x\|_1\|h\|_1 < \infty,$$

(inner integral is shift-invariant). Finiteness of the double integral means the inner integral $\int |x(\tau)h(t-\tau)|d\tau$ is finite for a.e. t (Proposition 46.4 applied to the t-marginal), which is the "defined a.e." claim, and the displayed bound is $\|x*h\|_1 \le \|x\|_1\|h\|_1$. Now Fubini applies to $x(\tau)h(t-\tau)e^{-j\omega t}$ (its modulus was just integrated), so the swap in Part V's proof of Theorem 31.4 — integrate over t first, substitute $u = t-\tau$, factor — is legal, line by line. □

The same two-step (Tonelli to check, Fubini to swap) certifies Parseval's derivation, the interchange $\frac{1}{T}\int_T \sum_k$ in Fourier series manipulations (sums are integrals with respect to counting measure — one theorem covers both), and every "exchange the order of integration" in statistical inference's expectation calculations.

Theorem 46.10 (Riemann–Lebesgue lemma)**.** *If $x \in L^1(\mathbb{R})$ then $X(j\omega) \to 0$ as $|\omega| \to \infty$.*

Proof sketch. For $x = \mathbf{1}_{[a,b]}$, compute: $|X(j\omega)| = \left|\frac{e^{-j\omega a} - e^{-j\omega b}}{j\omega}\right| \le \frac{2}{|\omega|} \to 0$. Finite combinations of such steps inherit the property; and step functions are dense in L^1 (MIRA 3B: simple functions from below, intervals approximate measurable sets), so for general x pick a step s with $\|x - s\|_1 < \varepsilon$ and use $|X| \le |\hat{s}| + \|x - s\|_1$ (Corollary 46.7(a)'s bound applied to $x - s$). □

Read backwards, this is a design constraint: a transform that does *not* vanish at infinity (the ideal lowpass brick wall $\mathbf{1}_{[-\omega_c,\omega_c]}$, viewed as the transform of h) cannot come from an L^1 impulse response — so the ideal filter of Lesson 32 is necessarily unstable in the BIBO sense of Theorem 29.3. The sinc's $1/t$ tail is not sloppiness; it is forced.

46.2 Practice: by hand

Example 46.11 (Fubini's hypothesis is not decoration)**.** $f(t,\tau) = \dfrac{t^2 - \tau^2}{(t^2+\tau^2)^2}$ on $(0,1)^2$. Using $\frac{\partial}{\partial \tau}\frac{\tau}{t^2+\tau^2} = f(t,\tau)$:

$$\int_0^1 \Big(\int_0^1 f \, d\tau \Big) dt = \int_0^1 \frac{dt}{1+t^2} = \frac{\pi}{4}, \qquad \int_0^1 \Big(\int_0^1 f \, dt \Big) d\tau = -\frac{\pi}{4}.$$

The two orders *disagree*, which by Fubini convicts the modulus: $\iint |f| = \infty$ (check: on the wedge $\tau < t$, $|f| \gtrsim 1/(4t^2)$ near the corner). Whenever Part V swapped a double integral, Tonelli's finiteness check was silently passing; this example is what failure looks like.

Example 46.12 (DCT in one line, twice)**.** (i) $\lim_n \int_0^\infty e^{-t}\cos^n(t)\,dt = 0$: integrand $\to 0$ a.e. (only at $t \in \pi\mathbb{Z}$ does $|\cos t| = 1$, a null set), dominated by $e^{-t} \in L^1$. (ii) $\lim_{a \downarrow 0} \int_0^\infty e^{-at}\frac{\sin t}{t^2+1}dt = \int_0^\infty \frac{\sin t}{t^2+1}dt$: dominated by $\frac{1}{t^2+1}$. This move — "evaluate a hard limit by passing it through the integral" — is used constantly in Fourier analysis to regularize divergent-looking transforms (the e^{-at} is Abel regularization, and DCT is why it gives the right answer when one exists).

Statistical-inference connection (the big one). A probability space *is* a measure space with total measure 1; $\mathbf{E}[X]$ *is* the Lebesgue integral of X; "almost surely" *is* "almost everywhere"; B&T's technical footnotes (which events have probabilities, why $\mathbf{E}$ is linear even for wild random variables, when $\mathbf{E}[\lim] = \lim \mathbf{E}$) are Definition 46.3, MCT, and DCT verbatim. The strong law of large numbers in Lesson 24 is an a.e. statement; statistical inference's convergence-of-random-variables menagerie is Lesson 45's pointwise/uniform story with measure replacing the sup. This lesson is the standard rigor layer under all graduate probability and statistical signal processing.

46.3 Exercises

Exercise 46.1 (Hand)**.** Which sets are null? (a) $\mathbb{Z}$; (b) $\mathbb{Q}$; (c) $[0,1]$; (d) the set of $t \in [0,1]$ whose decimal expansion contains no digit 7 (cover by intervals at each digit depth: total length $(9/10)^n \to 0$ — an *uncountable* null set).

Exercise 46.2 (Hand)**.** Justify or refute with DCT/MCT: (a) $\lim_n \int_0^1 \frac{n\sin(t/n)}{t(1+t^2)}dt = \int_0^1 \frac{dt}{1+t^2}$; (b) $\lim_n \int_0^n \left(1 - \frac{t}{n}\right)^n e^{t/2}dt = \int_0^\infty e^{-t/2}dt$; (c) $\lim_n \int_{\mathbb{R}} \frac{n\,dt}{1+n^2t^2} = 0$ (careful — compute it instead).

Exercise 46.3 (Proof, $\star$)**.** Prove Corollary 46.7(a) and (b) in full detail, stating exactly where DCT is invoked and what the dominating function is. Then use (b) twice to compute the transform of $t^2 e^{-|t|}$ from the known pair $e^{-|t|} \leftrightarrow \frac{2}{1+\omega^2}$ (Example "Two-sided decay," Lesson 31).

Exercise 46.4 (Proof)**.** Using Tonelli as in Corollary 46.9, prove the DT statement: if $x, h \in \ell^1$ then $x * h \in \ell^1$ with $\|x * h\|_1 \le \|x\|_1 \|h\|_1$ — i.e. the cascade-stability Exercise 29.3, which you proved by hand there, is a special case of Fubini with counting measure.

Deeper reading. MIRA ch. 1 (Riemann's limits), 2A–2D (outer measure, measurable sets), 3A–3B (integration, MCT, DCT), 5A–5B (product measures, Tonelli/Fubini). For the probability dictionary: MIRA ch. 12 or any graduate probability text's opening chapter.

47 Signal Space: Banach and Hilbert Spaces, L^2, and Orthonormal Bases

NEEDED FOR: Optional rigor layer for Fourier analysis and statistical inference; the native language of graduate DSP. Not needed for ML.

47.1 Theory

Part I did linear algebra in $\mathbb{R}^n$; Part V quietly worked in vector spaces of *signals*, where dimension is infinite and a new question appears that finite dimensions never asks: do limits stay in the space?

Definition 47.1 (Normed space, Banach space)**.** A *norm* on a vector space V is a function $\|\cdot\| : V \to [0, \infty)$ with $\|v\| = 0$ iff $v = 0$, $\|\alpha v\| = |\alpha| \|v\|$, and the triangle inequality. It makes V a metric space via $d(u, v) = \|u - v\|$; V is a *Banach space* if it is *complete*: every Cauchy sequence converges to a point of V. The standing examples:

$\ell^1,\ \ell^2,\ \ell^\infty$:	sequences with $\|x\|_1 = \sum \|x[n]\|$, $\|x\|_2 = \big(\sum \|x[n]\|^2\big)^{1/2}$, $\|x\|_\infty = \sup \|x[n]\|$;
$L^1(\mathbb{R}),\ L^2(\mathbb{R})$:	signals with $\|x\|_1 = \int \|x\|$, $\|x\|_2 = \big(\int \|x\|^2\big)^{1/2}$ (Lebesgue, Lesson 46);
$C[a, b]$:	continuous functions with $\|x\|_\infty = \sup \|x\|$.

In L^p, "$x = 0$" means $x = 0$ a.e. (Proposition 46.4) — signals equal a.e. are the same vector. All of these are Banach spaces; for L^1, L^2 this is the Riesz–Fischer theorem (MIRA 7A) and it is the payoff of Lesson 46:

Theorem 47.2 (Completeness is Lebesgue's gift)**.** *L^2 is complete. With the Riemann integral in place of Lebesgue's it would not be: the truncated-rational sequence of Example 46.1 is L^2-Cauchy with no Riemann-integrable limit. (Proof of completeness: MIRA 7A — extract a rapidly convergent subsequence, dominate, apply DCT.)*

Uniform convergence (Lesson 45) is convergence in $\|\cdot\|_\infty$; "energy convergence" is convergence in $\|\cdot\|_2$. They are genuinely different topologies on signal space, and Gibbs is the statement that partial Fourier sums converge in the second but not the first.

Definition 47.3 (Hilbert space)**.** A *Hilbert space* H is an inner product space (Lesson 2's axioms, complex form as in Lesson 9: $\langle x, y\rangle = \int x\,\bar{y}$ on L^2, $\sum x[n]\overline{y[n]}$ on ℓ^2) that is complete in the induced norm $\|x\| = \langle x, x\rangle^{1/2}$. So: ℓ^2 and L^2 are Hilbert spaces; they are the homes of DT and CT finite-energy signals. Cauchy–Schwarz (Theorem 2.5) holds verbatim — its Lesson 2 proof used only the axioms.

Which norms come from inner products? Exactly those satisfying the parallelogram law $\|u+v\|^2 + \|u-v\|^2 = 2\|u\|^2 + 2\|v\|^2$ (expand the inner products). The sup-norm fails it (Exercise 38.1), so $C[a,b]$ with $\|\cdot\|_\infty$ and ℓ^∞ are Banach but not Hilbert: no orthogonality, no projections, no Fourier geometry. The 2-norms are not a convention; they are the only norms with geometry.

Theorem 47.4 (Projection theorem — Lesson 6, infinite-dimensional)**.** *Let M be a* closed *subspace of a Hilbert space H and $x \in H$. Then there is a unique $\hat{x} \in M$ closest to x, and it is characterized by orthogonality: $x - \hat{x} \perp M$.*

Proof. Let $d = \inf_{m\in M}\|x-m\|$ and pick $m_n \in M$ with $\|x-m_n\| \to d$. The parallelogram law, applied to $u = x-m_n$, $v = x-m_m$, gives

$$\|m_n - m_m\|^2 = 2\|x-m_n\|^2 + 2\|x-m_m\|^2 - 4\left\|x - \tfrac{m_n+m_m}{2}\right\|^2 \le 2\|x-m_n\|^2 + 2\|x-m_m\|^2 - 4d^2 \to 0,$$

using $\frac{m_n+m_m}{2} \in M$. So (m_n) is Cauchy; by completeness *of H* it converges, and because M is *closed* the limit $\hat{x}$ lies in M, with $\|x-\hat{x}\| = d$. Orthogonality and uniqueness are then Lesson 6's computation (Theorem 6.3) unchanged: for $m \in M$, $\|x - \hat{x} + tm\|^2$ is minimized at $t = 0$, forcing $\langle x-\hat{x}, m\rangle = 0$. □

Every hypothesis earned its keep: completeness produced the limit, closedness kept it in M. This theorem is the deepest single fact in the lesson — least squares (Theorem 6.4), LLMS estimation (Theorem 22.3), and Fourier truncation are all instances.

Theorem 47.5 (Orthonormal sequences: Bessel, then Parseval)**.** *Let $(e_k)_{k\in\mathbb{Z}}$ be orthonormal in a Hilbert space H and $x \in H$, with "Fourier coefficients" $a_k = \langle x, e_k\rangle$.*

(a) (Bessel) $\sum_k |a_k|^2 \le \|x\|^2$; in particular the coefficient sequence lies in ℓ^2.

(b) The partial sums $S_N = \sum_{|k|\le N} a_k e_k$ are the orthogonal projections of x onto the span of $e_{-N}, \dots, e_N$, and $\sum_k a_k e_k$ always converges in H.

(c) The following are equivalent: (i) $\operatorname{span}(e_k)$ *is dense in H (the system is* complete*); (ii) $x = \sum_k a_k e_k$ for every x; (iii) Parseval, $\|x\|^2 = \sum_k |a_k|^2$, holds for every x.*

Proof. (b) first: $x - S_N \perp e_l$ for $|l| \le N$ by direct computation, so S_N is the projection (Theorem 47.4's characterization), and Pythagoras gives $\|x\|^2 = \|S_N\|^2 + \|x-S_N\|^2 = \sum_{|k|\le N}|a_k|^2 + \|x-S_N\|^2$. Letting $N \to \infty$ yields (a). Convergence of $\sum a_k e_k$: its partial sums are Cauchy because $\|\sum_{m\le|k|\le n} a_k e_k\|^2 = \sum_{m\le|k|\le n}|a_k|^2$ is a tail of the convergent series in (a) — completeness of H finishes (this is where Theorem 47.2 works for a living). (c): (i)⇒(ii): the limit $x' = \sum a_k e_k$ satisfies $x - x' \perp e_k$ for all k, hence $x - x' \perp$ a dense subspace, hence $x - x' \perp$ its closure $= H$, hence $x = x'$ (take the inner product with $x - x'$ itself). (ii)⇒(iii): continuity of the norm in the Pythagoras display above. (iii)⇒(i): if the span were not dense, Theorem 47.4 would produce $x \neq 0$ orthogonal to every e_k, violating Parseval. □

Theorem 47.6 (The harmonics are complete in $L^2[0,T]$)**.** *The normalized harmonics $e_k(t) = \frac{1}{\sqrt{T}} e^{jk\omega_0 t}$, $\omega_0 = 2\pi/T$, form a complete orthonormal system in $L^2[0,T]$. Consequently, for* every *finite-energy T-periodic signal — no differentiability, no Dirichlet conditions, jumps and all —*

$$\Big\| x - \sum_{|k|\le N} a_k e^{jk\omega_0 t} \Big\|_2 \longrightarrow 0 \qquad \textit{and} \qquad \frac{1}{T}\int_T |x|^2 = \sum_k |a_k|^2.$$

Proof sketch. Orthonormality is Lemma 30.2. Completeness reduces, by Theorem 47.5(c), to density of trigonometric polynomials in $L^2[0,T]$, which is proved in two standard steps: continuous functions are dense in L^2 (MIRA 3B, via simple and step functions), and trigonometric polynomials approximate continuous periodic functions uniformly (Fejér's theorem: the *averaged* partial sums $\frac{1}{N}\sum_{n<N} S_n$ use a nonnegative kernel and converge uniformly — Strichartz §1.2 or MIRA 11A; Lesson 52 plots exactly this Cesàro rescue of Gibbs). □

This upgrades Theorem 30.3 to its final form, and settles in what sense Fourier series "work for everything": in the energy norm, always; uniformly, only with smoothness (Corollary 45.14); pointwise a.e., true for L^2 but genuinely hard (Carleson 1966 — know the name, skip the proof). For the line instead of the circle, the same machinery yields:

Theorem 47.7 (Plancherel)**.** *The Fourier transform, defined on $L^1 \cap L^2$ and extended by continuity, is a bijection of $L^2(\mathbb{R})$ with $\int |x(t)|^2 dt = \frac{1}{2\pi}\int |X(j\omega)|^2 d\omega$ and $\langle x,\, y\rangle = \frac{1}{2\pi}\langle X,\, Y\rangle$. (MIRA ch. 11 / Strichartz §3.3; the DT sibling: the DTFT is, up to $\frac{1}{2\pi}$, the inverse of a Fourier* series*, so Theorem 47.6* is *its Parseval — Exercise 32.3 certified.)*

So the transform is a unitary change of basis of signal space — Lesson 12's Theorem 12.4, escaped from dimension N to dimension ∞. That was Part II's promise; this is the theorem that keeps it.

Definition 47.8 (Bounded operators)**.** A linear map $A : H \to H$ is *bounded* if $\|A\| := \sup_{x\neq 0}\|Ax\|/\|x\| < \infty$ — the operator norm, the infinite-dimensional largest singular value (Lesson 13). Bounded = continuous for linear maps (same ε–δ bookkeeping as always).

Theorem 47.9 (Stable LTI filters are bounded operators on ℓ^2)**.** *If $h \in \ell^1$, then $x \mapsto h * x$ maps ℓ^2 into ℓ^2 with*

$$\|h * x\|_2 \;\le\; \Big(\max_{\omega} |H(e^{j\omega})|\Big)\, \|x\|_2 \;\le\; \|h\|_1\, \|x\|_2.$$

The operator norm is exactly $\max_\omega |H(e^{j\omega})|$.

Proof. By Plancherel (DT form) and the convolution property (Theorem 32.3):

$$\|h*x\|_2^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2 |X(e^{j\omega})|^2 d\omega \le \big(\max|H|\big)^2 \frac{1}{2\pi}\int_{-\pi}^{\pi} |X|^2 = \big(\max|H|\big)^2 \|x\|_2^2,$$

and $\max|H| \le \sum |h[k]|$ by the triangle inequality. Sharpness: concentrate $|X|^2$ near the maximizing frequency. □

In frequency domain the filter is the *multiplication operator* $X \mapsto H{\cdot}X$: a "diagonal matrix" indexed by ω, with the transform as the diagonalizing unitary. That is Lesson 12's circulant story (Theorem 12.6), now exact in infinite dimensions — and it is the template for graduate DSP, where filters, decimators, channel models, and estimators are all operators on ℓ^2, compared by operator norm, and "design" means shaping $H(e^{j\omega})$ under constraints.

Statistical-inference connection. Random variables with finite variance form a Hilbert space, $L^2(\Omega, \mathbf{P})$, with $\langle X, Y\rangle = \mathbf{E}[X\bar{Y}]$ — Lesson 21 flagged that covariance behaves like an inner product (Theorem 21.3); it *is* one. The orthogonality principle of LLMS estimation (Theorem 22.3) is Theorem 47.4 verbatim, with M the span of the observations; conditional expectation $\mathbf{E}[X \mid Y]$ is the projection onto the (closed) subspace of all functions of Y. Wiener filters and Kalman filters — statistical inference's second half — are computed projections in this Hilbert space, nothing more. One theorem, learned once, running the whole estimation theory.

47.2 Practice: by hand

Example 47.10 (Best approximation, computed like Lesson 6)**.** Project $x(t) = t$ on $[-\pi, \pi]$ onto $\operatorname{span}\{1, \cos t, \sin t\}$ in L^2. The three are orthogonal with $\|1\|^2 = 2\pi$, $\|\cos\|^2 = \|\sin\|^2 = \pi$. Coefficients: $\langle t, 1\rangle = 0$, $\langle t, \cos\rangle = 0$ (odd integrand), $\langle t, \sin\rangle = \int_{-\pi}^{\pi} t \sin t\, dt = 2\pi$. So $\hat{x}(t) = \frac{2\pi}{\pi} \sin t = 2 \sin t$ — the first term of the sawtooth's Fourier series, of course: partial Fourier sums *are* these projections (Theorem 47.5(b)). Residual energy: Pythagoras, $\|x - \hat{x}\|^2 = \|x\|^2 - \|2 \sin t\|^2 = \frac{2\pi^3}{3} - 4\pi \approx 8.11$.

Example 47.11 (Why closedness matters)**.** In ℓ^2, let M be the subspace of sequences with finitely many nonzero terms — dense, not closed. The vector $x[n] = 2^{-n}$ $(n \geq 0)$ has no closest point in M: truncations get arbitrarily close but the infimum 0 is not attained. Theorem 47.4 does not fail; its hypothesis does. (In graduate DSP this is not pedantry: "the projection onto the space of causal finite-energy signals" exists because that space is closed; "onto the FIR filters of unbounded order" does not.)

47.3 Exercises

Exercise 47.1 (Hand)**.** Show the parallelogram law fails for $\|\cdot\|_\infty$ on $C[0, 1]$ (try $u(t) = 1$, $v(t) = t$) and for $\|\cdot\|_1$ on $\mathbb{R}^2$ (try $u = (1, 0)$, $v = (0, 1)$). Conclude neither norm admits an inner product — so "project onto the closest point" is not even well-posed there (find two closest points to $(1, 1)$ in $\operatorname{span}\{(1, 0)\}$ under $\|\cdot\|_\infty$... there are many).

Exercise 47.2 (Hand)**.** In $L^2[0, 1]$, apply Gram–Schmidt (Lesson 6's algorithm, verbatim, with $\langle f, g\rangle = \int_0^1 f\bar{g}$) to $1, t, t^2$. You will produce the first three (shifted) Legendre polynomials. Then compute the best line approximating e^t in energy, and compare its slope with the calculus tangent at any point — projection is a global fit, not a local one.

Exercise 47.3 (Proof, $\star$)**.** Prove that in the Pythagoras display of Theorem 47.5's proof, the error $\|x - S_N\|$ is *decreasing* in N, and deduce the "best approximation improves monotonically" claim that Lesson 30 made after Theorem 30.3. Then prove: if Parseval holds for x and y separately, the polarization identity forces $\langle x, y\rangle = \sum_k a_k \bar{b}_k$ (Parseval for inner products).

Exercise 47.4 (Proof)**.** Prove that the DT ideal lowpass operator (multiplication by $\mathbf{1}_{|\omega| \leq \omega_c}$ in frequency) is a bounded operator on ℓ^2 of norm 1, even though Lesson 46 showed its impulse response is not in ℓ^1 and Theorem 29.3 showed it is not BIBO stable. Moral: ℓ^2-boundedness and BIBO (ℓ^∞) stability are different norms on the same operator — graduate DSP tracks both.

Deeper reading. MIRA 6A–6B (Banach), 7A (completeness of L^p), 8A–8B (Hilbert spaces, orthonormal bases); Bowers–Kalton ch. 1 (normed spaces) and 7.1–7.2 (Hilbert space theory, operators). For Fejér and Carleson context: Strichartz §1.2.

48 Complex Analysis: Why Poles, ROCs, and Inversion Integrals Work

NEEDED FOR: Optional rigor layer for Fourier analysis (inversion, Gaussians) and the s/z-plane; standard equipment for graduate DSP. Not needed for ML.

48.1 Theory

Part V's transform tables, ROC rules, and partial-fraction inversions all rest on the behavior of functions of a *complex* variable. The miracle of the subject is that one hypothesis — complex differentiability on an open set — implies everything else.

Definition 48.1 (Holomorphic functions)**.** f is *holomorphic* (equivalently *analytic*) on an open set $\Omega \subseteq \mathbb{C}$ if $f'(z) = \lim_{h\to 0} \frac{f(z+h)-f(z)}{h}$ exists for every $z \in \Omega$, with h approaching 0 from *every direction in* $\mathbb{C}$. Writing $f = u + jv$ and comparing the limit along the real and imaginary axes forces the *Cauchy–Riemann equations* $u_x = v_y$, $u_y = -v_x$; conversely CR (with continuous partials) implies holomorphy (Bak–Newman ch. 3). Polynomials, e^z, $\sin z$, $\cos z$ are entire (holomorphic on $\mathbb{C}$); rational functions are holomorphic off their poles; $\bar{z}$, $|z|^2$, $\operatorname{Re} z$ are holomorphic nowhere — "anti-rotating" behavior fails the directional limit.

Definition 48.2 (Contour integrals; the ML bound)**.** For a curve $\gamma : [a, b] \to \mathbb{C}$ and continuous f, $\int_\gamma f\,dz = \int_a^b f(\gamma(t))\gamma'(t)\,dt$. The one estimate used everywhere (*ML bound*): $\left|\int_\gamma f\,dz\right| \le \max_\gamma |f| \cdot \text{length}(\gamma)$ — the triangle inequality for integrals, nothing more.

Theorem 48.3 (Cauchy's theorem and integral formula)**.** *Let f be holomorphic on and inside a simple closed curve γ (traversed counterclockwise). Then*

$$\oint_\gamma f\,dz = 0, \qquad \textit{and} \qquad f(z_0) = \frac{1}{2\pi j}\oint_\gamma \frac{f(z)}{z - z_0}\,dz \quad \textit{for every } z_0 \textit{ inside } \gamma.$$

Proof sketch. The theorem is proved first for rectangles by a bisection argument (Goursat: subdivide, zoom in on the worst quadrant, use differentiability at the limit point — Bak–Newman ch. 4), then for general curves via local primitives. The formula follows from the theorem: the integrand is holomorphic between γ and a tiny circle C_ε around z_0, so the two integrals agree; on C_ε, write $f(z) = f(z_0) + O(\varepsilon)$ and compute $\oint_{C_\varepsilon} \frac{dz}{z-z_0} = \int_0^{2\pi} \frac{j\varepsilon e^{j\theta}}{\varepsilon e^{j\theta}} d\theta = 2\pi j$, while the $O(\varepsilon)$ part dies by ML. □

The formula says a holomorphic function is determined inside a curve by its boundary values — and differentiating it under the integral sign (legal by Lesson 46!) gives $f^{(n)}(z_0) = \frac{n!}{2\pi j}\oint \frac{f(z)}{(z-z_0)^{n+1}} dz$ for every n: *one* complex derivative buys infinitely many, plus a convergent Taylor series in the largest disk avoiding singularities (Bak–Newman ch. 6). Two famous one-liners follow:

Theorem 48.4 (Liouville; Fundamental Theorem of Algebra)**.** *(a) A bounded entire function is constant: the derivative formula on a radius-R circle gives $|f'(z_0)| \le \max|f|/R \to 0$. (b) Every nonconstant polynomial p has a root in $\mathbb{C}$: else $1/p$ would be entire and bounded (it $\to 0$ at ∞), hence constant. Dividing out roots: p factors completely into linear factors.*

Part (b) is why *every* rational transform in Lesson 34 splits into partial fractions over $\mathbb{C}$ — the algebra that Part V used from a table is a consequence of complex analysis, not an accident of examples.

Definition 48.5 (Laurent series, poles, residues)**.** On an annulus $r < |z - z_0| < R$ where f is holomorphic, f has a unique expansion $f(z) = \sum_{n=-\infty}^{\infty} c_n (z-z_0)^n$ (Bak–Newman ch. 9). If only finitely many negative powers appear, z_0 is a *pole* of order m (simple if $m = 1$); the coefficient c_{-1} is the *residue* $\operatorname{Res}_{z_0} f$. For a simple pole, $\operatorname{Res}_{z_0} f = \lim_{z\to z_0}(z - z_0)f(z)$; if $f = p/q$ with $q(z_0) = 0 \neq q'(z_0)$, this is $p(z_0)/q'(z_0)$; for an order-m pole, $\operatorname{Res}_{z_0} f = \frac{1}{(m-1)!}\lim_{z\to z_0}\frac{d^{m-1}}{dz^{m-1}}\big[(z - z_0)^m f(z)\big]$.

Theorem 48.6 (Residue theorem)**.** *If f is holomorphic on and inside the simple closed curve γ except at finitely many points $z_1, \dots, z_k$ inside, then*

$$\oint_\gamma f\,dz = 2\pi j \sum_{i=1}^{k} \operatorname{Res}_{z_i} f.$$

Proof sketch. Shrink γ to tiny circles around each z_i (Cauchy's theorem on the Swiss-cheese region between); on each circle integrate the Laurent series term by term (uniform convergence on the circle, Theorem 45.11): every power $(z - z_i)^n$ integrates to 0 except $n = -1$, which gives $2\pi j\, c_{-1}$ — the $\oint \frac{dz}{z-z_0} = 2\pi j$ computation from Theorem 48.3, which is thus the only integral anyone ever actually evaluates in this subject. □

Example 48.7 (A transform pair, certified by contour)**.** Claim (Lesson 31's two-sided decay pair, read backwards):

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{2a}{a^2 + \omega^2}\, e^{j\omega t}\, d\omega = e^{-a|t|}, \qquad a > 0.$$

Take $t > 0$. Close the contour with a large upper semicircle $|\omega| = R$: on it, $|e^{j\omega t}| = e^{-t\,\mathrm{Im}\,\omega} \le 1$ and the rational factor is $O(1/R^2)$, so by ML the semicircle contributes $O(1/R) \to 0$ (for slower $1/R$ decay one invokes *Jordan's lemma*, Bak–Newman ch. 11 — the $e^{j\omega t}$ decay rescues the estimate). The integrand's poles are at $\omega = \pm ja$; only $\omega = +ja$ is inside. Residue (p/q' rule): $\frac{2a\,e^{j(ja)t}}{2(ja)} = \frac{e^{-at}}{j}$. Hence the integral is $\frac{1}{2\pi}\cdot 2\pi j \cdot \frac{e^{-at}}{j} = e^{-at}$. For $t < 0$ close *downward* (that is where $e^{j\omega t}$ decays), pick up $\omega = -ja$ with a sign flip from orientation: e^{at}. Together: $e^{-a|t|}$. ✓ The "close the contour on the side where the exponential dies" rule is the entire mechanics of transform inversion.

Now the s- and z-planes make sense. Three facts, all direct consequences:

1. *Transforms are holomorphic on their ROCs.* Corollary 46.7(c) lets us differentiate $X(s) = \int x(t)e^{-st}dt$ in s, under the integral sign: X is holomorphic on the open ROC strip. Same for $X(z)$ on its annulus.

2. *The ROC is pole-free and pole-bounded.* Holomorphy fails exactly at poles, so the ROC cannot contain one; and for rational transforms the Taylor/Laurent expansion converges precisely up to the nearest singularity, so the ROC of Proposition 34.3 extends exactly to the nearest pole line/circle — Part V's "ROC bounded by poles" rule, now a theorem about radii of convergence.

3. *The z-transform* is *a Laurent series* — $X(z) = \sum_n x[n]z^{-n}$, in the variable z^{-1} — and Laurent coefficients on a given annulus are *unique*, recoverable by the contour integral $x[n] = \frac{1}{2\pi j}\oint_\gamma X(z)z^{n-1}dz$ (γ any circle in the ROC). That is why "$X(z)$ *plus its ROC*" determines $x[n]$ (Example 34.2's moral), why the same formula with different annuli gives different signals, and why the inversion tables of Lesson 34 have unique answers. The Laplace inversion (Bromwich) integral $x(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st}ds$ is the same statement on a vertical line in the ROC, evaluated exactly as in Example 48.7: close left for $t > 0$ (picking up the poles left of the ROC $\to$ causal modes $e^{p_i t}$), close right for $t < 0$.

Pole locations are therefore not bookkeeping; they are the singularity structure of a holomorphic function, and "causal and stable iff all poles in the open left half-plane / unit disk with the ROC outside them" is complex analysis wearing an engineering hat.

48.2 Practice: by hand

Example 48.8 (z-inversion by Laurent expansion, both ROCs)**.** $X(z) = \dfrac{1}{1-az^{-1}}$. On $|z| > |a|$: expand in powers of az^{-1} (geometric, converges there): $X(z) = \sum_{n\geq 0} a^n z^{-n}$, so $x[n] = a^n u[n]$. On $|z| < |a|$: rewrite $X = \frac{-a^{-1}z}{1-a^{-1}z}$ and expand in powers of $a^{-1}z$: $X(z) = -\sum_{m\geq 1} a^{-m}z^m$, so $x[n] = -a^n u[-n-1]$. Two annuli, two Laurent series, two signals — Example 34.2 *re-derived* instead of table-matched. Check the second against the inversion contour for $n = -1$: $x[-1] = \frac{1}{2\pi j}\oint z^{-2}X(z)\,dz$ with $z^{-2}X(z) = \frac{z^{-1}}{z-a}$ (use $X(z) = \frac{z}{z-a}$); on a circle inside $|z| < |a|$ the only enclosed pole is $z = 0$, with residue $\frac{1}{0-a} = -a^{-1}$. ✓ ($x[-1] = -a^{-1}$.)

Example 48.9 (Residue drill)**.** (a) $\mathrm{Res}_{z=j}\frac{1}{z^2+1} = \frac{1}{2j}$ (p/q'). (b) $\mathrm{Res}_{s=-2}\frac{s+1}{(s+2)^2(s+3)}$: order-2 pole, $\frac{d}{ds}\frac{s+1}{s+3}\big|_{s=-2} = \frac{2}{(s+3)^2}\big|_{-2} = 2$. (c) $\oint_{|z|=2}\frac{e^z}{z(z-1)}dz = 2\pi j\big(\mathrm{Res}_0 + \mathrm{Res}_1\big) = 2\pi j\,(-1+e)$. Note (b) is exactly a repeated-root partial-fraction coefficient — the residue *is* the partial-fraction numerator, which is why the two techniques never disagree.

> **Fourier / grad DSP connection.** Fourier analysis evaluates inversion integrals and the Gaussian's self-transform by exactly these contour moves (shift the path, invoke Cauchy). In graduate DSP: *minimum-phase* systems are those whose H and $1/H$ are both holomorphic on $|z| > 1$ — log-magnitude and phase then determine each other (Hilbert transform relations, from Cauchy's formula); *spectral factorization* (splitting a power spectrum into causal × anticausal, the engine of Wiener filtering) is an exercise in allocating poles and zeros between $|z| < 1$ and $|z| > 1$; and the Paley–Wiener theorems say "causal ⇔ holomorphic in a half-plane" in full generality. None of that is startable without this lesson.

48.3 Exercises

Exercise 48.1 (Hand)**.** Verify the Cauchy–Riemann equations for $f(z) = e^z = e^x(\cos y + j\sin y)$ and their failure for $f(z) = \bar{z}$. Then explain in one sentence why $|X(j\omega)|$ is never holomorphic

in ω (moduli are real-valued), and why that is consistent with $X(s)$ being holomorphic in s.

Exercise 48.2 (Hand)**.** Compute by residues: (a) $\int_{-\infty}^{\infty} \frac{d\omega}{(\omega^2+1)(\omega^2+4)} = \frac{\pi}{6}$; (b) $\operatorname{Res}_{z=0} \frac{\sin z}{z^4}$ (Laurent: $= -\frac{1}{6}$); (c) the inverse Laplace transform of $\frac{1}{(s+1)(s+2)}$ with ROC $\operatorname{Re} s > -1$, by Bromwich-plus-residues, and check against partial fractions.

Exercise 48.3 (Proof, $\star$)**.** Prove Liouville's theorem from the derivative form of the Cauchy formula (ML bound on a radius-R circle, $R \to \infty$), then deduce the Fundamental Theorem of Algebra, then deduce that every strictly proper rational function is a finite sum of terms $\frac{c_{ik}}{(s-p_i)^k}$. (You have now proved that the partial-fraction method of Lesson 34 always works.)

Exercise 48.4 (Proof)**.** Let $x[n]$ be causal with $\sum |x[n]| r^{-n} < \infty$ for some $r < 1$ (exponentially stable). Show $X(z)$ is holomorphic on $|z| > r$, and that $\log X$ is holomorphic there too provided X has no zeros in $|z| > r$ — the minimum-phase condition. (Differentiate under the sum; Lesson 46 licenses it.)

Deeper reading. Bak–Newman chs. 1–3 (complex numbers, analyticity, CR), 4–6 (Cauchy theory, power series, Liouville/FTA), 9–11 (Laurent series, singularities, residues, real-integral applications).

49 Distributions: Making δ Honest

NEEDED FOR: Optional rigor layer for Fourier analysis (which develops exactly this) and statistical inference (white noise); essential for graduate DSP. Not needed for ML.

49.1 Theory

Part V used $\delta(t)$ under a treaty ("defined operationally by sifting"), and paid for it in caveats: δ is not a function (no function is 0 a.e. yet integrates to 1 — Proposition 46.4 forbids it), the square wave was differentiated only by analogy, and the impulse-train identity of Lemma 33.1 was "certified elsewhere." Distribution theory is the elsewhere. The single idea: *stop asking for the value of a signal at a point; ask only for its averages against smooth test signals.* Every measurement in engineering is such an average — no instrument reads $x(t_0)$; it reads $\int x\varphi$ for some aperture φ.

Definition 49.1 (Test functions)**.** $\mathcal{D}$ is the space of infinitely differentiable functions $\varphi : \mathbb{R} \to \mathbb{C}$ that vanish outside a bounded interval. It is not obvious such functions exist; the standard bump

$$\varphi(t) = \begin{cases} e^{-1/(1-t^2)}, & |t| < 1 \\ 0, & |t| \geq 1 \end{cases}$$

is C^∞ (all derivatives of $e^{-1/s}$ vanish as $s \downarrow 0$ faster than any power — Ross §31's classic example), and shifts, scalings, and sums of it generate a rich supply. A sequence $\varphi_n \to \varphi$ *in* $\mathcal{D}$ if all supports sit in one common bounded interval and $\varphi_n^{(k)} \to \varphi^{(k)}$ uniformly for every k — the strongest convergence imaginable, which is the point: the stronger the convergence on test functions, the *weaker* (more inclusive) the dual notion defined next.

Definition 49.2 (Distributions)**.** A *distribution* (generalized function) is a linear functional $f : \mathcal{D} \to \mathbb{C}$, written $\langle f, \varphi\rangle$, that is continuous: $\varphi_n \to \varphi$ in $\mathcal{D}$ implies $\langle f, \varphi_n\rangle \to \langle f, \varphi\rangle$. The set of distributions is $\mathcal{D}'$.

- Every locally integrable signal x *is* a distribution via $\langle x, \varphi\rangle = \int x(t)\varphi(t)\,dt$ — and two signals give the same functional iff they agree a.e., so nothing is lost and nothing spurious is added. (Note: this pairing is linear in φ, without the complex conjugate of Lesson 47's inner product — same bracket, different convention, standard in both subjects.)
- $\langle \delta, \varphi\rangle = \varphi(0)$, and $\langle \delta_{t_0}, \varphi\rangle = \varphi(t_0)$: *sifting is the definition*, not a derived property. Continuity is immediate. O&W's operational account and this one now agree by construction.

Definition 49.3 (Operations, by moving them to the test function)**.** Every operation on distributions is defined by transposing what it would do to an honest signal under the integral sign (substitute, integrate by parts — then *define* the result to be that formula):

shift:	$\langle f(t-t_0), \varphi\rangle = \langle f, \varphi(t+t_0)\rangle$	
scale:	$\langle f(at), \varphi\rangle = \frac{1}{\lvert a\rvert}\langle f, \varphi(t/a)\rangle$	
multiply by smooth g :	$\langle gf, \varphi\rangle = \langle f, g\varphi\rangle$	$(g\varphi \in \mathcal{D}$ ✓$)$
differentiate:	$\langle f', \varphi\rangle = -\langle f, \varphi'\rangle$	(integrate by parts).

Each right-hand side is a continuous functional of φ, so the definitions are legal; and when f is a differentiable signal, integration by parts says the new definition agrees with the classical one. Consequences, instantly:

Proposition 49.4 (The δ calculus, now theorems)**.** *(a)* $\delta(at) = \frac{1}{\lvert a\rvert}\delta(t)$*: both sides send* $\varphi \mapsto \varphi(0)/\lvert a\rvert$.

(b) $g(t)\,\delta(t-t_0) = g(t_0)\,\delta(t-t_0)$ *for smooth* g*: both sides send* $\varphi \mapsto g(t_0)\varphi(t_0)$.

(c) $u' = \delta$*:* $\langle u', \varphi\rangle = -\langle u, \varphi'\rangle = -\int_0^\infty \varphi'(t)\,dt = \varphi(0) = \langle \delta, \varphi\rangle$.

(d) δ' *(the* dipole*) sends* $\varphi \mapsto -\varphi'(0)$*; more generally* $\langle \delta^{(k)}, \varphi\rangle = (-1)^k\varphi^{(k)}(0)$.

(e) Every *distribution has derivatives of all orders. Differentiation, the most fragile operation in classical analysis (Theorem 45.15's hypotheses; Weierstrass's nowhere-differentiable function), is* unconditionally legal *here — the cost being that the derivative may no longer be a function.*

Example 49.5 (Differentiating the square wave, honestly)**.** The T-periodic square wave x of Example 30.5 (+1 then −1) is classically differentiable a.e. with derivative 0 — useless. As a distribution, $x' = \sum_m \big(2\,\delta(t-t_m^{\uparrow}) - 2\,\delta(t-t_m^{\downarrow})\big)$: each jump of height J contributes $J\,\delta$ at the jump (the computation is (c) above, localized). The general rule: *distributional derivative = classical derivative a.e. + one impulse per jump, weighted by the jump.* This is the law behind Part V's "differentiate the triangle to get the square, differentiate the square to get impulses," and it converts the smoothness–decay ladder of Exercise 30.3 into an exact bookkeeping: each differentiation multiplies a_k by $jk\omega_0$, and a signal whose m-th derivative first shows impulses has $\lvert a_k\rvert \sim 1/k^m$.

Definition 49.6 (Convergence of distributions)**.** $f_n \to f$ *in* $\mathcal{D}'$ if $\langle f_n, \varphi\rangle \to \langle f, \varphi\rangle$ for every test function φ — convergence of all measurements. This is the weakest convergence in this booklet, and therefore the most forgiving.

Proposition 49.7 (Nascent deltas)**.** *Let $k \in L^1(\mathbb{R})$ with $\int k = 1$, and set $k_\epsilon(t) = \frac{1}{\epsilon}k(t/\epsilon)$. Then $k_\epsilon \to \delta$ in $\mathcal{D}'$ as $\epsilon \downarrow 0$.*

Proof. $\langle k_\epsilon, \varphi\rangle = \int k(s)\,\varphi(\epsilon s)\,ds \to \varphi(0)\int k = \varphi(0)$, by DCT (Theorem 46.6): the integrands are dominated by $\|\varphi\|_\infty |k| \in L^1$. — Every "limit of narrow pulses" picture of δ (O&W's rectangle of width Δ, the Gaussian of shrinking variance, Exercise 37.5's Cauchy kernels) is this one proposition. Remarkably, decay of k is not needed if oscillation substitutes for it: $\frac{\sin(t/\epsilon)}{\pi t} \to \delta$ in $\mathcal{D}'$ too (its tails cancel by Riemann–Lebesgue, Theorem 46.10) — and *that* family is precisely the truncated Fourier inversion integral, which is why inversion recovers x: partial inversion $= x * (\text{nascent } \delta) \to x$. □

Example 49.8 (What distributions refuse to do)**.** There is no continuous way to multiply two arbitrary distributions: $\delta \cdot \delta$ would need $\varphi(0)\cdot$"$\delta(0)$," which does not exist — consistent with Part V's experience that LTI theory (linear!) handles impulses effortlessly while a nonlinear system fed an impulse ($y = x^2$ of Example 28.7) is meaningless. Similarly $\delta(t)u(t)$ is undefined (u is not smooth at exactly the wrong point). The rule of thumb: distributions may be added, shifted, scaled, differentiated, convolved with nice things, and multiplied by *smooth* functions — and that list is exactly the toolbox LTI theory uses.

Tempered distributions and the Fourier transform. To transform a distribution we need test functions the transform maps to *themselves* — compact support dies (a compactly supported signal has an entire, never compactly supported, transform: Lesson 48). The right class trades support for decay:

Definition 49.9 (Schwartz class; tempered distributions)**.** $\mathcal{S}$ is the space of C^∞ functions all of whose derivatives decay faster than any power ($t^m e^{-t^2}$ and friends). The Fourier transform maps $\mathcal{S}$ onto $\mathcal{S}$, bijectively, with the inversion formula valid classically (Strichartz §3.2–3.3 — the proof is Corollary 46.7 run repeatedly: smoothness of φ buys decay of $\hat\varphi$ and vice versa; the transform trades the two, and $\mathcal{S}$ is where the trade is even). *Tempered distributions* $\mathcal{S}'$ are the continuous linear functionals on $\mathcal{S}$: everything of at most polynomial growth — all of Part V's signals, periodic signals, impulse trains, but not e^{t^2}. For $f \in \mathcal{S}'$, define

$$\langle \mathcal{F}f, \varphi\rangle = \langle f, \mathcal{F}\varphi\rangle.$$

Since $\mathcal{F}$ is a bijection of $\mathcal{S}$, $\mathcal{F}f$ is again a tempered distribution: *every* tempered distribution has a Fourier transform, and inversion holds in $\mathcal{S}'$. The convention matches Part V's: $\mathcal{F}\varphi(\omega) = \int \varphi(t)e^{-j\omega t}dt$.

Theorem 49.10 (The impossible pairs, computed)**.** *In $\mathcal{S}'$:*

$$\mathcal{F}\delta = 1, \qquad \mathcal{F}1 = 2\pi\,\delta(\omega), \qquad \mathcal{F}e^{j\omega_0 t} = 2\pi\,\delta(\omega-\omega_0), \qquad \mathcal{F}\cos(\omega_0 t) = \pi\big(\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\big).$$

Proof. First: $\langle \mathcal{F}\delta, \varphi\rangle = \langle \delta, \hat\varphi\rangle = \hat\varphi(0) = \int \varphi(t)\,dt = \langle 1, \varphi\rangle$. Second: $\langle \mathcal{F}1, \varphi\rangle = \langle 1, \hat\varphi\rangle = \int \hat\varphi(\omega)\,d\omega = 2\pi\varphi(0)$ by the inversion formula on $\mathcal{S}$ evaluated at $t = 0$. The third is the second plus the shift rule (Definition 49.3, transposed through $\mathcal{F}$ — the modulation property of Theorem 31.3, which holds in $\mathcal{S}'$ because it holds on $\mathcal{S}$); the fourth is Euler plus linearity. These are the δ-pairs of Example 31.2, no longer "certified by sifting" but proved. Note the engine: *a formula true on $\mathcal{S}$ transposes to all of $\mathcal{S}'$ for free* — every property in Theorem 31.3 (shift, modulation, differentiation $\mathcal{F}(f') = j\omega\mathcal{F}f$, convolution with an L^1 kernel) is valid for tempered distributions by this one argument. □

Theorem 49.11 (Impulse train ↔ impulse train (Poisson summation))**.** *In $\mathcal{S}'$, with $\omega_s = 2\pi/T$:*

$$\mathcal{F}\Big[\sum_{n\in\mathbb{Z}} \delta(t-nT)\Big] = \frac{2\pi}{T}\sum_{k\in\mathbb{Z}} \delta(\omega - k\omega_s), \qquad \textit{equivalently} \qquad \sum_n \varphi(nT) = \frac{1}{T}\sum_k \hat{\varphi}(k\omega_s) \ \ \forall \varphi \in \mathcal{S}.$$

Proof sketch. The train $p(t) = \sum_n \delta(t - nT)$ is T-periodic, and its Fourier *series* coefficients are all $a_k = \frac{1}{T}\int_{-T/2}^{T/2} p\, e^{-jk\omega_s t} dt = \frac{1}{T}$ (sifting on one period). Summing the resulting series $p = \frac{1}{T}\sum_k e^{jk\omega_s t}$ in $\mathcal{S}'$ (partial sums converge as distributions — test and regroup; Strichartz §7 does the epsilonics) and transforming term by term with Theorem 49.10 gives the claim. Pairing the first identity with φ yields the second — a nontrivial numerical identity between two ordinary sums, free of any δ's, which is the form used in analytic number theory and in filter-bank aliasing analysis alike. □

This retroactively proves Lemma 33.1, hence spectrum replication (Theorem 33.2), hence the sampling theorem (Theorem 33.3): the entire sampling story of Lesson 33 now stands on $\mathcal{S}'$ with no operational IOUs. Likewise white noise — statistical inference's "process with autocorrelation $\sigma^2\delta(\tau)$" — is honestly a distribution-valued object: its sample paths are not functions, but its averages against test apertures are perfectly well-defined random variables, which is exactly how the physical instrument sees it.

49.2 Practice: by hand

Example 49.12 (Drill)**.** (a) $t\,\delta(t) = 0$ (Proposition 49.4(b) with $g = t$); hence the general solution of $t\,f = 0$ in $\mathcal{D}'$ is $f = c\,\delta$ — division by t is possible "up to impulses," the fact behind $\mathcal{F}u = \pi\delta(\omega) + \frac{1}{j\omega}$ (Exercise 40.3). (b) $\langle \delta', t\varphi\rangle = -\frac{d}{dt}(t\varphi)|_0 = -\varphi(0)$, i.e. $t\,\delta' = -\delta$: dipoles differentiate what multiplies them. (c) $\frac{d}{dt}|t| = \operatorname{sign} t$, $\frac{d^2}{dt^2}|t| = 2\delta$: the triangle-wave/square-wave/impulse-train ladder of Part V in one line per rung. (d) $(x * \delta_{t_0}) = x(t - t_0)$: convolution with a shifted impulse shifts — Part V's Example usage, proved by transposition.

Fourier / statistical-inference / grad DSP connection. Osgood's Fourier lecture notes devote two chapters to exactly this material — $\mathcal{S}$, $\mathcal{S}'$, the pairs of Theorem 49.10, Poisson summation — so this lesson is direct course prep, not backstory. In statistical inference, generalized processes (white noise, ideal derivatives of Brownian motion) live in $\mathcal{S}'$. In graduate DSP, distribution theory is load-bearing wherever ideal elements meet analysis: sampling and multirate identities (Theorem 49.11 is the aliasing formula), PDE-based signal models, sparse spike deconvolution (the unknown is a sum of δ's), and the theory of generalized spectral densities.

49.3 Exercises

Exercise 49.1 (Hand)**.** Compute, by pairing with a test function: (a) the second distributional derivative of $\max(t, 0)$ (answer: δ); (b) $\frac{d}{dt}\big[e^{-t}u(t)\big]$ (answer: $\delta - e^{-t}u(t)$ — the product rule survives because e^{-t} is smooth); (c) $\delta'(-t)$ in terms of δ' (odd).

Exercise 49.2 (Hand)**.** Using Theorem 49.10 and linearity only, transform: (a) a $+1/-1$ square wave via its Fourier series (an impulse train in frequency with weights a_k); (b) $\operatorname{sign}(t) = 2u(t) - 1$ given $\mathcal{F}u = \pi\delta + \frac{1}{j\omega}$; (c) t (differentiate $\mathcal{F}1$: answer $2\pi j\,\delta'(\omega)$).

Exercise 49.3 (Proof, $\star$)**.** Derive $\mathcal{F}u = \pi\delta(\omega) + \frac{1}{j\omega}$ (principal value): write $u = \frac{1}{2} + \frac{1}{2}\operatorname{sign}(t)$, transform the constant with Theorem 49.10, and get $\mathcal{F}[\operatorname{sign}]$ as the $\mathcal{S}'$-limit of $\mathcal{F}[e^{-a|t|}\operatorname{sign}(t)] = \frac{-2j\omega}{a^2+\omega^2}$ as $a \downarrow 0$ (pair with φ and use DCT). Conclude the integration property of Theorem 31.3 — divide by $j\omega$, add $\pi X(0)\delta$ — which Part V stated without proof.

Exercise 49.4 (Proof)**.** Show that if $f_n \to f$ in $\mathcal{D}'$ then $f_n' \to f'$ in $\mathcal{D}'$ — differentiation is *continuous* on distributions (transpose and use the definition), in stark contrast to Example 45.16's second row. Apply it: the Fourier partial sums of the square wave converge in $\mathcal{D}'$, therefore so do their derivatives, to the impulse train of Example 49.5 — termwise differentiation of Fourier series is *always* legal distributionally.

Deeper reading. Strichartz §§1.1–1.4 (what distributions are), 2.1–2.4 (the calculus of distributions), 3.2–3.5 ($\mathcal{S}$, tempered distributions, the Fourier transform); §7 for Poisson summation and sampling. O&W 1.4 and 2.5 are the operational account this lesson upgrades.

50 Dual Spaces, Riesz Representation, and Functional Derivatives

NEEDED FOR: Optional rigor layer; the calculus of graduate DSP, estimation, and machine learning in function space. Not needed for ML (but explains its gradients).

50.1 Theory

Lesson 49 was secretly a lesson about *dual spaces*; this lesson makes the pattern explicit and then builds the derivative that goes with it.

Definition 50.1 (Dual space)**.** For a normed space V, the *dual* V^* is the space of bounded (equivalently continuous) linear functionals $\ell : V \to \mathbb{C}$, with norm $\|\ell\| = \sup_{\|v\|\le 1} |\ell(v)|$. Examples already met: $\mathcal{D}'$ and $\mathcal{S}'$ are duals ($\mathcal{D}^*$, $\mathcal{S}^*$, with continuity in the appropriate convergences); "evaluate at t_0" is in $C[a,b]^*$ (norm 1) but is *not* a bounded functional on L^2 — point values are meaningless in L^2 (a.e.-classes!), which is precisely why $\delta \notin L^2$ and why sampling theory (Lesson 33) needed bandlimitedness to make evaluation continuous.

Theorem 50.2 (Riesz representation)**.** *Every bounded linear functional on a Hilbert space H is an inner product: for each $\ell \in H^*$ there is a unique $y \in H$ with*

$$\ell(x) = \langle x,\, y\rangle \quad \textit{for all } x \in H, \qquad \textit{and} \qquad \|\ell\| = \|y\|.$$

Proof. If $\ell = 0$ take $y = 0$. Otherwise $M = \ker \ell$ is a closed proper subspace (closed because ℓ is continuous). By the projection theorem (Theorem 47.4) pick a unit vector $e \perp M$ (project any $x_0 \notin M$ onto M and normalize the residual). Every $x \in H$ splits as $x = \big(x - \frac{\ell(x)}{\ell(e)}e\big) + \frac{\ell(x)}{\ell(e)}e$, first piece in M (apply ℓ: zero). Take inner products with e: $\langle x,\, e\rangle = \frac{\ell(x)}{\ell(e)}$, i.e. $\ell(x) = \langle x,\, \overline{\ell(e)}\, e\rangle$; set $y = \overline{\ell(e)}\, e$. Uniqueness: if $\langle x,\, y_1 - y_2\rangle = 0$ for all x, test $x = y_1 - y_2$. Norm equality: Cauchy–Schwarz one way, $x = y/\|y\|$ the other. □

So on a Hilbert space the dual is the space itself — the infinite-dimensional justification of a move Part I made without comment: representing the linear functional "score" as a weight *vector* ($\ell(x) = w \cdot x$), and the linear part of a loss change as a gradient vector. Riesz is what lets both survive in function space:

Definition 50.3 (Functional derivative)**.** Let $J : H \to \mathbb{R}$ be a functional on (an open subset of) a Hilbert space of signals. Its *Gâteaux derivative* at x in direction v is

$$\delta J(x; v) = \frac{d}{d\epsilon} J(x + \epsilon v)\Big|_{\epsilon=0},$$

— ordinary one-variable calculus along the line $x + \epsilon v$. If $v \mapsto \delta J(x; v)$ is a bounded linear functional, Riesz supplies a unique $\nabla J(x) \in H$ with

$$\delta J(x; v) = \langle v,\, \nabla J(x)\rangle \quad \text{for all directions } v,$$

the *functional gradient*. When only distributional pairing is available, physics writes the same object as the density $\frac{\delta J}{\delta x(t)}$: $\delta J(x; v) = \int \frac{\delta J}{\delta x(t)}\, v(t)\, dt$ — a functional derivative *is* a distribution in the perturbation, and ∇J is its Riesz representative when it has one. Steepest descent, stationarity ($\nabla J = 0$), and the chain rule all read exactly as in Lesson 7 (Proposition 7.3 is the finite-dimensional case).

Example 50.4 (The three computations that cover practice)**.** All on real L^2, all by expanding $J(x + \epsilon v)$ and reading the $O(\epsilon)$ term.

(a) *Energy.* $J(x) = \frac{1}{2}\|x\|_2^2$: $J(x+\epsilon v) = J(x)+\epsilon\langle v,\, x\rangle+O(\epsilon^2)$, so $\nabla J = x$. (The $\frac{1}{2}\|\cdot\|^2$-gradient-is-identity rule.)

(b) *Data misfit through a filter.* $J(x) = \frac{1}{2}\|h * x - y\|_2^2$:

$$\delta J(x; v) = \langle h * v,\ h * x - y\rangle = \langle v,\ \tilde{h} * (h * x - y)\rangle, \qquad \tilde{h}(t) = h(-t),$$

so $\nabla J = \tilde{h} * (h * x - y)$: *filter the residual with the time-reversed filter.* The middle step moved h across the inner product — the *adjoint* of convolution-by-h is convolution-by-$\tilde{h}$ (substitute $u = t - \tau$; in frequency: the adjoint of multiplication by H is multiplication by $\bar{H}$, Lesson 11's conjugate-transpose in its ω-indexed form). Time-reversed filters in backpropagation-through-convolution and in matched filtering both come from this adjoint.

(c) *Smoothness penalty.* $J(x) = \frac{1}{2}\int (x')^2$: $\delta J(x; v) = \int x'v' = -\int x''v$ (integration by parts, boundary terms zero for compactly supported perturbations — Lesson 49's move), so $\frac{\delta J}{\delta x(t)} = -x''(t)$. Derivative penalties have *second*-derivative gradients: the Laplacian smoothing of graphics and ML.

Theorem 50.5 (Fundamental lemma of the calculus of variations; Euler–Lagrange)**.** *(a) If g is locally integrable and $\int g\,\varphi = 0$ for all $\varphi \in \mathcal{D}(a, b)$, then $g = 0$ a.e. on (a, b) — i.e. the map from signals to distributions is injective, so a vanishing functional derivative pins down the function. (b) Consequently, a smooth minimizer of $J(x) = \int_a^b L\big(t, x(t), x'(t)\big)dt$ with fixed endpoints satisfies*

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\frac{\partial L}{\partial x'} = 0.$$

Proof. (a) is MIRA 3B (pair against smoothed indicators of $\{g > \varepsilon\}$; the bump functions of Definition 49.1 exist precisely to run this argument). (b): at a minimizer, $0 = \delta J(x; \varphi) = \int \big(L_x\varphi + L_{x'}\varphi'\big) = \int \big(L_x - \frac{d}{dt}L_{x'}\big)\varphi$ for every test φ (integrate by parts; endpoint terms vanish since φ does); apply (a). □

Example 50.6 (Matched filter, by Cauchy–Schwarz)**.** Among unit-energy filters h, which maximizes the output $|(s*h)(0)|$ when the input is a known pulse s? $(s*h)(0) = \int s(\tau)h(-\tau)d\tau = \langle s, \tilde{h}\rangle$, and Cauchy–Schwarz (Theorem 2.5) gives $|\langle s, \tilde{h}\rangle| \le \|s\|\|h\|$ with equality iff $\tilde{h} \propto s$: the optimal filter is $h(t) = s(-t)/\|s\|$, the *time-reversed template* — the matched filter of every radar and every communication receiver. In statistical inference's Hilbert space of random variables, the identical argument (with the noise-whitened inner product) yields the SNR-optimal detector; this is Example 22.6's finite-dimensional geometry, grown up.

Example 50.7 (Tikhonov-regularized deconvolution — the capstone's engine)**.** Given noisy filtered data $y \approx h * x$, minimize

$$J(x) = \tfrac{1}{2}\|h*x - y\|_2^2 + \tfrac{\lambda}{2}\|x\|_2^2.$$

By Example 50.4(a)+(b): $\nabla J = \tilde{h} * (h*x - y) + \lambda x$. Setting $\nabla J = 0$ and Fourier-transforming (Plancherel makes this legitimate; $\mathcal{F}\tilde{h} = \bar{H}$): $\bar{H}(H\hat{X} - Y) + \lambda\hat{X} = 0$, so

$$\hat{X}(j\omega) = \frac{\overline{H(j\omega)}}{|H(j\omega)|^2 + \lambda}\, Y(j\omega).$$

At $\lambda = 0$ this is naive inverse filtering Y/H — explosive wherever $H \approx 0$; the λ floor is exactly what tames the ill-conditioning that Lesson 51 will quantify as a condition number. This formula (and its statistical twin, the Wiener filter, where λ becomes the noise-to-signal spectral ratio) is the single most-reused answer in graduate DSP.

Connections, backward and forward. Backward: machine learning's gradient descent (Lesson 7) is this lesson with $H = \mathbb{R}^n$; the LLMS orthogonality principle (Theorem 22.3) is $\nabla J = 0$ for $J = \mathbf{E}[(X - \hat{X})^2]$; least squares (Theorem 6.4) is Example 50.4(b) with matrices. Forward: Wiener and Kalman filtering (statistical inference) are projections computed via these gradients; adaptive filters (LMS, RLS) run stochastic descent on $J = \mathbf{E}|d - h*x|^2$ over the filter h — the gradient is again an adjoint-filtered residual; variational methods, splines, and physics-informed models all start from Theorem 50.5. When a deep-learning paper writes $\frac{\delta \mathcal{L}}{\delta f}$ for a loss over a function, it means Definition 50.3.

50.2 Practice: by hand

Example 50.8 (Drill)**.** (a) $J(x) = \langle x, g\rangle$ (linear): $\nabla J = g$, constant — as in Lesson 7's table. (b) $J(x) = \int c(t)x(t)^2 dt$, c smooth: $\nabla J = 2cx$ (pointwise weight). (c) $J(x) = \frac{1}{2}\|x'\|^2 + \frac{1}{2}\|x\|^2$: stationarity gives $-x'' + x = 0$ — Euler–Lagrange producing a differential equation, the pattern of all variational modeling. (d) Shortest path: $J(x) = \int_a^b \sqrt{1 + x'^2}\, dt$: $L_x = 0$, so $\frac{d}{dt}\frac{x'}{\sqrt{1+x'^2}} = 0$, so x' constant: straight lines — the sanity check every calculus-of-variations user runs once.

50.3 Exercises

Exercise 50.1 (Hand)**.** Compute functional gradients on L^2: (a) $J(x) = \langle x, g\rangle^2$; (b) $J(x) = \frac{1}{4}\int x^4$; (c) $J(x) = \frac{1}{2}\|h*x\|^2$ (use the adjoint). Answers: $2\langle x, g\rangle\, g$; x^3; $\tilde{h} * h * x$.

Exercise 50.2 (Hand)**.** Derive the Euler–Lagrange equation for the smoothing spline functional $J(x) = \frac{1}{2}\sum_i (x(t_i) - y_i)^2 + \frac{\lambda}{2}\int (x'')^2$ formally: $\frac{\delta J}{\delta x(t)} = \sum_i (x(t_i) - y_i)\,\delta(t - t_i) + \lambda x''''(t)$ — note the data term's derivative *is a train of impulses* (Lesson 49), which is why the minimizer is a piecewise cubic with knots exactly at the data.

Exercise 50.3 (Proof, $\star$)**.** Prove the adjoint identity used in Example 50.4(b): $\langle h * v,\, w\rangle = \langle v,\, \tilde{h} * w\rangle$ for real $h, v, w \in L^2$ with $h \in L^1$ (Fubini, Lesson 46, licenses the swap). Conclude that the operator norm of convolution-by-$\tilde{h}$ equals that of convolution-by-h (Theorem 47.9: $|\bar{H}| = |H|$).

Exercise 50.4 (Proof)**.** In Example 50.7, prove J is strictly convex for $\lambda > 0$ (expand $J(x + v) - J(x) - \langle v,\, \nabla J(x)\rangle = \frac{1}{2}\|h * v\|^2 + \frac{\lambda}{2}\|v\|^2 > 0$ for $v \neq 0$), so the stationary point is the unique global minimizer — the function-space copy of Lesson 6's least-squares uniqueness.

Deeper reading. Bowers–Kalton ch. 2 (dual spaces), 7.1–7.2 (Riesz, operators and adjoints); Strichartz §1 (distributions as duals); any calculus of variations opening chapter (e.g. Gelfand–Fomin ch. 1) for Theorem 50.5 in depth.

51 Numerical Linear Algebra: Conditioning, Stability, and Structure

NEEDED FOR: Optional; the compute layer under the numerical work in this booklet and under graduate DSP. Not needed for ML — but it explains numerical mystery failures.

51.1 Theory

Parts I–V trusted `solve`, `lstsq`, `eigh`, `fft`. This lesson is the bare minimum of Trefethen–Bau needed to know *when* to trust them — which, in graduate DSP (where the matrices are correlation matrices and blurring operators, often nearly singular by nature), is a daily question, not a numerical analyst's hobby.

Definition 51.1 (Floating point, in one box)**.** IEEE double precision represents reals with relative error at most $\varepsilon_{\text{mach}} \approx 1.1 \times 10^{-16}$: $\text{fl}(t) = t(1 + \epsilon)$, $|\epsilon| \leq \varepsilon_{\text{mach}}$, and each arithmetic operation is exact up to the same relative fudge. Sixteen digits sounds like plenty; the subject exists because *subtraction of nearly equal numbers* promotes relative error catastrophically: $\text{fl}(1 + 10^{-13}) - 1$ keeps only 3 of its 16 digits. (*Catastrophic cancellation* — T&B Lecture 13.)

Definition 51.2 (Conditioning is the problem's fault)**.** The *condition number* of solving $Av = b$ (or of least squares) is

$$\kappa(A) = \|A\|\,\|A^{-1}\| = \frac{\sigma_1}{\sigma_n}$$

(operator norms; the SVD form is Lesson 13's, and Theorem 47.9's $\max|H|$ is its operator-norm half). It bounds the worst-case relative-error amplification from data to solution: $\frac{\|\delta v\|}{\|v\|} \lesssim \kappa(A)\,\frac{\|\delta b\|}{\|b\|}$. Proof in two lines: $\delta v = A^{-1}\delta b$ gives $\|\delta v\| \leq \|A^{-1}\|\|\delta b\|$, and $b = Av$ gives $\|b\| \leq \|A\|\|v\|$; divide. The bound is attained when b lies along the top left singular vector and δb along the bottom one — the ill-conditioned directions are the small-σ ones, exactly the near-zeros of $|H|$ in Example 50.7.

Definition 51.3 (Stability is the algorithm's virtue)**.** An algorithm for $v = f(b)$ is *backward stable* if its computed answer $\tilde{v}$ is the *exact* answer to a nearby problem: $\tilde{v} = f(\tilde{b})$ with $\|\tilde{b} -$

$b\|/\|b\| = O(\varepsilon_{\text{mach}})$. The two notions compose into the fundamental accuracy law (T&B Lecture 15):

$$\frac{\|\tilde{v} - v\|}{\|v\|} = O\big(\kappa \cdot \varepsilon_{\text{mach}}\big) :$$

a backward stable algorithm loses about $\log_{10}\kappa$ *digits, and no algorithm can do better on a* κ*-conditioned problem.* Householder QR, `lstsq`, `eigh`, and the FFT are backward stable; Gaussian elimination with partial pivoting (`solve`) is in practice; Gram–Schmidt as Lesson 6 wrote it is *not* (its computed Q drifts from orthogonality by $O(\kappa\,\varepsilon_{\text{mach}})$ — "modified" Gram–Schmidt and reorthogonalization exist for this reason, T&B Lectures 7–10).

Proposition 51.4 (Normal equations square the condition number)**.** $\kappa(A^{\mathsf{T}}A) = \kappa(A)^2$ *(singular values square under* $A^{\mathsf{T}}A$*, Lesson 13). Hence solving least squares via* $A^{\mathsf{T}}A\hat{x} = A^{\mathsf{T}}b$ *(Theorem 6.4) computes a* κ^2*-conditioned problem: for* $\kappa(A) = 10^5$ *— routine for Vandermonde, wavelet, or correlation matrices — the normal equations leave* ~ 6 *digits where QR leaves* ~ 11*. The rule taught by every numerical course:* form $A^{\mathsf{T}}A$ for theory, never for computation; call QR-based `lstsq`*. (statistical inference's sample covariance matrices* $\frac{1}{n}X^{\mathsf{T}}X$ *are the same trap wearing statistics clothing; SVD of the data matrix* X *is the stable route to PCA.)*

Proposition 51.5 (The SVD is the numerical microscope)**.** *For measured data, exact rank is meaningless (*fl *noise fills every dimension); the usable notion is* numerical rank*: the number of singular values above the noise floor* $\sigma_1 \cdot O(\varepsilon)$ *or above the measurement noise level. Truncating the SVD at that threshold,* $A_r = \sum_{i\le r}\sigma_i u_i v_i^{\mathsf{T}}$*, is the best rank-r approximation (Eckart–Young, Lesson 13's optimality), and solving* $Av = b$ *by* $v = \sum_{i\le r}\frac{u_i^{\mathsf{T}}b}{\sigma_i}v_i$ *(*truncated SVD*) or by damping* $\frac{\sigma_i}{\sigma_i^2+\lambda}$ *(*Tikhonov *— exactly Example 50.7 with singular vectors in place of frequencies) is how ill-posed deconvolution, system identification, and subspace methods (MUSIC/ESPRIT-style: signal subspace = top singular vectors of a data matrix) are actually computed in graduate DSP.*

Proposition 51.6 (Structure is speed: circulant and Toeplitz)**.** *An LTI filter on a length-*N *window is a* Toeplitz *matrix (constant diagonals, Lesson 29); with periodic boundary it is* circulant*, and Lesson 12 proved every circulant is diagonalized by the DFT:* $C = F^{-1}\Lambda F$ *with* $\Lambda = \operatorname{diag}(\hat{c})$*. Consequences:*

- *circulant multiply / solve = FFT, pointwise op, inverse FFT:* $O(N\log N)$ *and backward stable (the FFT is a product of unitary-scaled sparse factors), vs.* $O(N^2)/O(N^3)$ *dense — and* $\kappa(C) = \max|\hat{c}|/\min|\hat{c}|$*, read straight off the frequency response: an invertible-looking filter with a deep spectral notch is an ill-conditioned matrix, no SVD needed;*
- *Toeplitz systems — the normal equations of LMS/Wiener filtering, autocorrelation matrices of stationary processes (statistical inference) — admit* $O(N^2)$ *Levinson–Durbin recursion (the workhorse of linear prediction and speech coding) and* $O(N\log^2 N)$ *superfast variants; embedding a Toeplitz matrix in a circulant of twice the size makes Toeplitz* multiplication *an FFT, the trick behind every fast convolution in this booklet (Lesson 29).*

Proposition 51.7 (Iteration beats factorization at scale)**.** *When* N *is too large to factor, solve* $Av = b$ *(*A *symmetric positive definite) by* conjugate gradients*: each step needs only a matrix–vector product (an FFT, for our structured matrices), and the error contracts like* $2\big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\big)^k$ *(T&B Lecture 38) — conditioning again, now as a* speed *of convergence; preconditioning is the art of lowering the effective* κ*. Similarly* power iteration $v \mapsto Av/\|Av\|$ *converges to the top*

eigenvector at rate $|\lambda_2/\lambda_1|^k$ (Lesson 10's powers-of-A picture): top-k PCA on data too big for `eigh`*. Steepest descent on $\frac{1}{2}v^\mathsf{T}Av - b^\mathsf{T}v$ is the $H = \mathbb{R}^n$ shadow of Lesson 50's descent — one theory, three sizes: analytic (Lesson 50), factorized (this lesson), iterated (this proposition).*

51.2 Practice: by hand

Example 51.8 (Conditioning without a computer)**.** $A = \begin{pmatrix} 1 & 1 \\ 1 & 1.0001 \end{pmatrix}$: nearly dependent columns. $\sigma_1 \approx 2$, $\sigma_2 \approx 5 \times 10^{-5}$ (product = det = 10^{-4}, sum of squares = $\|A\|_F^2 \approx 4$), so $\kappa \approx 4 \times 10^4$. Perturb $b = (2, 2.0001)$ — solution $(1, 1)$ — to $b' = (2, 2.0002)$: solution jumps to $(0, 2)$. A 5×10^{-5} relative data change produced an $O(1)$ solution change: κ in action. This is what a near-flat spectral notch, a nearly coherent pair of sensors, or two nearly identical features (Lesson 6's duplicated-column `LinAlgError`) all look like to the arithmetic.

Grad DSP connection. Estimation and adaptive filtering run on exactly three numerical kernels: least squares (use QR/SVD, mind Proposition 51.4), eigen/SVD of covariance and data matrices (PCA, subspace tracking, MUSIC), and Toeplitz/circulant solves (Wiener, linear prediction, fast convolution). Conditioning is not a corner case there: correlation matrices of oversampled or narrowband signals are *structurally* near-singular, and regularization (λ, diagonal loading, SVD truncation) is the standing repair — Lesson 50 gave its variational meaning, this lesson its numerical one.

51.3 Exercises

Exercise 51.1 (Hand)**.** Predict the digits: you solve a least-squares problem with $\kappa(A) = 10^6$ in double precision. How many correct digits does QR-based `lstsq` leave? The normal equations? Would single precision ($\varepsilon \approx 6 \times 10^{-8}$) with QR beat double precision with normal equations? (Answers: $\sim$10; $\sim$4; no: $\kappa\varepsilon \approx 6 \times 10^{-2}$ leaves $\sim$1 digit, worse than the normal equations' $\sim$4 — stability cannot buy back precision that the format never had.)

Exercise 51.2 (Hand)**.** A filter has frequency response magnitudes between 0.01 and 2.5 on the DFT grid. What is the condition number of its length-N circulant matrix? How does Tikhonov regularization with $\lambda = 10^{-2}$ change the effective ratio max / min of $\frac{|H|}{|H|^2+\lambda}$-damped inversion gains? (Compute the new worst gain max $\frac{|H|}{|H|^2+\lambda} \le \frac{1}{2\sqrt{\lambda}} = 5$.)

Exercise 51.3 (Proof, $\star$)**.** Prove $\kappa(A^\mathsf{T}A) = \kappa(A)^2$ from the SVD, and prove Eckart–Young for the operator norm: if $\operatorname{rank} B \le r$ then $\|A - B\| \ge \sigma_{r+1}$ (find a unit vector in $\operatorname{span}(v_1, \dots, v_{r+1}) \cap \operatorname{null} B$ — dimension counting, Lesson 5 — on which A acts with norm $\ge \sigma_{r+1}$).

Exercise 51.4 (Proof)**.** Show that the DFT matrix satisfies $\kappa(F) = 1$ (unitary up to scale, Theorem 12.4) — transforms lose no digits — and that for a circulant $C = F^{-1}\Lambda F$, backward-stable FFTs give a circulant solve with relative error $O(\kappa(C)\varepsilon)$: structure does not evade conditioning, it only evades cost.

Deeper reading. Trefethen–Bau Lectures 4–5 (SVD), 7–11 (QR and least squares), 12–15 (conditioning, floating point, stability), 18–19 (conditioning and stability of least squares), 24–27 (eigenvalue iterations), 38 (conjugate gradients). Levinson–Durbin: any statistical DSP text, e.g. Hayes ch. 5.

52 Capstone VII: The Certification Lab

NEEDED FOR: Optional synthesis of Part VII. Not needed for ML.

One program, six licenses exercised. Run it top to bottom, then do the checkpoints — each one is a Part V manipulation paired with the Part VII theorem that certifies it.

```
import numpy as np
rng = np.random.default_rng(3)

# ---------- 1. Gibbs vs L2 vs Cesaro (Lessons 36, 38) ----------
t = np.linspace(-np.pi, np.pi, 40001)
x = np.sign(np.sin(t))
def partial(N, cesaro=False):
    k = np.arange(1, N+1, 2)
    w = (1 - k/(N+1)) if cesaro else np.ones_like(k, float)
    return (4/np.pi)*(w*np.sin(np.outer(t, k))/k).sum(axis=1)
for N in (21, 201, 2001):
    SN, FN = partial(N), partial(N, cesaro=True)
    print(N, SN.max()-1, np.sqrt(np.trapezoid((SN-x)**2, t)), FN.max()-1)
# overshoot -> 2*(0.0895): Gibbs constant; L2 error -> 0 (Lesson 47);
# Cesaro overshoot <= 0 ALWAYS: Fejer's kernel is >= 0, so the averaged
# sums never exceed the signal's bounds -- no Gibbs, and they converge
# uniformly on any closed interval missing the jumps.
gibbs = (np.trapezoid(np.sin(np.linspace(1e-9, np.pi, 10001)) /
                      np.linspace(1e-9, np.pi, 10001),
                      np.linspace(1e-9, np.pi, 10001)) / np.pi - 0.5)
print("Gibbs constant:", gibbs)                     # 0.08949

# ---------- 2. delta, tested not plotted (Lesson 49) ----------
phi = np.where(np.abs(t) < 1, np.exp(-1/np.clip(1-t**2, 1e-12, None)), 0)
pair = lambda f: np.trapezoid(f*phi, t)
for eps in (0.1, 0.02, 0.005):
    print(pair(np.sin(t/eps)/(np.pi*t + 1e-300)))
# -> phi(0) = 1/e = 0.3679 (last digits limited by quadrature of the
# fast oscillation). The truncated INVERSION INTEGRAL is a nascent delta --
# "F^-1 F = id" and "sinc family -> delta" are the same statement.

# ---------- 3. a pair by residues, checked (Lesson 48) ----------
th = np.linspace(0, 2*np.pi, 4096, endpoint=False)
def zinv(X, n, r):
    z = r*np.exp(1j*th)
    return (X(z)*z**n).mean().real                  # (1/2pij) oint X z^{n-1} dz
X = lambda z: 1/((1-0.9/z)*(1-0.5/z))
pf = lambda n: (0.9**(n+1)-0.5**(n+1))/0.4          # partial fractions
print([round(zinv(X, n, 1.2), 6) for n in range(5)],
      [round(pf(n), 6) for n in range(5)])          # identical

# ---------- 4. deconvolution: gradient = closed form (Lessons 41, 38) ----
N = 1024; dt = 1/N
g = np.exp(-((np.arange(N)-N/2)*dt)**2/(2*0.01**2)); g /= g.sum()
G = np.fft.fft(np.fft.ifftshift(g))
xt = ((np.arange(N)%256) < 128).astype(float)       # square wave again
```

```
y = np.fft.ifft(G*np.fft.fft(xt)).real + 0.02*rng.standard_normal(N)
print("min |G|:", np.abs(G).min())   # underflows to 0: raw kappa is INFINITE
for lam in (1e-1, 1e-3, 1e-5, 1e-8):
    xr = np.fft.ifft(np.conj(G)*np.fft.fft(y)/(np.abs(G)**2+lam)).real
    print(lam, np.abs(xr-xt).mean(),
          (np.abs(G)/(np.abs(G)**2+lam)).max())      # damped worst gain
# error is U-shaped in lambda: bias up, variance down -- pick the valley;
# without lambda the problem is not merely ill-conditioned but singular.

# ---------- 5. kappa^2 in the wild (Lesson 51) ----------
A = np.vander(np.linspace(0, 1, 100), 12)
c = rng.standard_normal(12); b = A @ c
print(np.linalg.cond(A),
      np.abs(np.linalg.lstsq(A, b, rcond=None)[0]-c).max(),
      np.abs(np.linalg.solve(A.T@A, A.T@b)-c).max())
```

Checkpoints (do all of these).

1. Step 1: the sup-norm column refuses to converge while the L^2 column dies — say which theorem forbids the first (Corollary 45.10) and which guarantees the second (Theorem 47.6). Then explain the Cesàro column's good behavior from the proof sketch of Theorem 47.6 (Fejér's nonnegative kernel: averaging is a nascent-δ convolution with $k \geq 0$, so no overshoot — connect to Proposition 49.7).

2. Step 2: replace the bump φ by $\mathbf{1}_{[-1,1]}$ (not smooth!). The sinc pairing still converges (to 1), but slowly and with oscillation — tabulate the error vs. ϵ for both test functions. Smoothness of the test class is what buys the fast, monotone convergence; this is why $\mathcal{D}$ and $\mathcal{S}$ insist on C^∞.

3. Step 3: move the contour to $r = 0.7$ and $r = 0.4$ and identify the two other signals sharing the formula $X(z)$ (Lesson 48's three-ROC story for two poles: which is two-sided?). Verify $n = -1$ values against the Laurent expansions.

4. Step 4: set λ by the discrepancy principle (smallest λ with residual $\|g * x_r - y\| \leq 1.1\sqrt{N} \cdot 0.02$) and check it lands near the error valley. Then double the blur width and watch both the raw κ and the optimal λ move — quantify Lesson 50's Exercise 41.5 with Lesson 51's numbers.

5. Step 5: state the two error ratios as powers of $\kappa\varepsilon_{\text{mach}}$ and check the exponents (Proposition 51.4).

6. Throughout: every integral in this lab was a Lebesgue integral approximated by a finite sum, every swap was a DCT/Fubini instance, and the whole lab ran in ℓ^2-representable double precision — name, for each step, the Part VII lesson that certifies it. That closing exercise *is* the part.

Final self-check list for Part VII

From memory: state the completeness axiom and prove the monotone convergence of sequences; define pointwise/uniform convergence and prove the uniform limit theorem and the M-test;

explain why Gibbs is forced; state MCT/DCT/Fubini with their counterexamples (spike, escape, the $\pm\pi/4$ double integral) and use DCT to differentiate under an integral; define a.e. and explain what L^2 signals are; prove the projection theorem and Bessel, state Riesz–Fischer, Parseval, Plancherel, and the boundedness of stable filters on ℓ^2; state Cauchy's theorem and formula, compute residues, invert a rational z-transform by contour or Laurent series for each ROC, and explain why ROCs are pole-bounded; define distributions and tempered distributions, differentiate a jump, prove $\delta(at) = \delta(t)/|a|$ and $u' = \delta$, transform 1, $e^{j\omega_0 t}$, and the impulse train, and state Poisson summation; state Riesz representation and compute the functional gradients of $\frac{1}{2}\|h * x - y\|^2 + \frac{\lambda}{2}\|x\|^2$ and $\frac{1}{2}\int (x')^2$; derive Euler–Lagrange and the matched filter; define κ and backward stability, state the accuracy law $O(\kappa\varepsilon)$, explain why normal equations lose double the digits, and diagonalize a circulant with the FFT. That is the license layer: every formal manipulation in Part V now has a theorem behind it, and the vocabulary of graduate DSP — Hilbert spaces, operators, distributions, conditioning — is installed.

Part VIII: Convex Optimization and Information Theory for ML, Signals, and Sensors

Every part of this booklet has quietly leaned on the same two unnamed disciplines. Lesson 6 minimized $\|Ax-b\|$; Lesson 40 minimized a mean-square error; Lesson 37 asked for the smallest ripple a length-N filter can achieve; Lesson 41 asked how distinguishable two noisy hypotheses are; Lessons 38 and 43 traded bits against distortion in converters and codecs. Each time, the question underneath was one of two: *among all designs meeting the constraints, which is best — and how would we certify it?* (optimization), and *how much can the data reveal at all, whatever the algorithm?* (information). Part VIII names the two disciplines and installs their working cores:

1. What makes a set or a function convex, why does convexity turn searching into certifying, and how are least squares, LASSO, logistic regression, and Chebyshev filter design all the same kind of problem? (Lesson 53.)

2. What do penalty terms actually do to a solution — and why is the ridge term the exact cure for Lesson 51's ill-conditioning? (Lesson 54.)

3. What is a dual problem, what do multipliers mean, and how do the KKT conditions certify an answer and price a constraint? (Lesson 55 — where water-filling is solved by hand, to be spent in Lesson 58.)

4. What do training loops compute, why do they slow down exactly when eigenvalues spread, and how does a corner in $\|x\|_1$ become an exact zero? (Lesson 56.)

5. What, in bits, does a measurement carry? (Lesson 57: entropy, divergence, mutual information, and the inequalities — Jensen, data processing, Fano — that police them.)

6. How fast can hypotheses be distinguished, how many bits can a noisy channel or a noisy sensor deliver, and what floor does Fisher information put under every estimator? (Lesson 58.)

7. How few bits keep a source within a distortion budget, and what does that say about quantizers, codecs, and learned representations? (Lesson 59.)

The capstone runs the whole part as one sensor pipeline: calibrate (54), sparsify (56), detect (58), allocate (55), and quantize (59), each step checked against the theorem that predicted it. Two sources, both self-contained here at working depth: Boyd–Vandenberghe, *Convex Optimization* (BV) for Lessons 53–56, and Polyanskiy–Wu, *Information Theory: From Coding to Learning* (PW) for Lessons 57–59. The ground rule is Part VII's: a proof appears when it teaches a mechanism you will reuse (Jensen, Cauchy–Schwarz, the KKT balance of forces); a theorem is stated with a pointer when its proof is a course of its own (Shannon's coding theorems). Prerequisites: Parts I, III, and V, with Lessons 22–24 (Part IV) wanted for the information half; Part VI makes every example land harder but is not assumed. Conventions: log is base 2 when the unit is bits and natural when calculus is being done — each use says which; problems are minimizations unless stated; $h_2(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the binary entropy, a function worth memorizing.

53 Convex Models: Sets, Functions, and Problems

NEEDED FOR: ML (losses, regularizers, margins), DSP (filter design), inference and sensors (estimation under constraints).

53.1 Theory

Definition 53.1 (Convex set)**.** $C \subseteq \mathbb{R}^n$ is *convex* if for all $x, y \in C$ and $t \in [0,1]$, $tx+(1-t)y \in C$: the segment between any two points stays inside.

The examples are the sets this booklet already lives in. Affine sets $\{x : Ax = b\}$ (solution sets of Lesson 5) are convex: $A(tx + (1-t)y) = tb + (1-t)b = b$. Halfspaces $\{x : a^\mathsf{T} x \le b\}$ are convex by linearity of the dot product. Norm balls $\{x : \|x - x_0\| \le r\}$ are convex by the triangle inequality — for *every* norm, so ℓ_1 diamonds and ℓ_∞ boxes count, not just Euclidean balls. The positive semidefinite matrices form a convex cone: $z^\mathsf{T}(tP + (1-t)Q)z \ge 0$ termwise (the set Lesson 11 certified covariances into, Theorem 23.3).

Proposition 53.2 (Intersection preserves convexity)**.** *An arbitrary intersection of convex sets is convex. In particular a polyhedron $\{x : Ax \le b,\ Cx = d\}$ — an intersection of halfspaces and hyperplanes — is convex, so* any *list of linear constraints defines a convex feasible set.*

Proof. If x, y lie in every C_α, then $tx + (1-t)y \in C_\alpha$ for each α (convexity of that one set), hence in the intersection. □

Definition 53.3 (Convex function)**.** $f : C \to \mathbb{R}$ on a convex domain is *convex* if for all $x, y \in C$, $t \in [0,1]$,

$$f(tx + (1-t)y) \ \le \ tf(x) + (1-t)f(y),$$

i.e. chords lie above the graph; *strictly convex* if the inequality is strict for $x \ne y$, $t \in (0,1)$; *concave* if $-f$ is convex.

Theorem 53.4 (First-order characterization: tangents are global underestimators)**.** *A differentiable f on a convex domain is convex if and only if*

$$f(y) \ \ge \ f(x) + \nabla f(x)^\mathsf{T}(y - x) \qquad \textit{for all } x, y.$$

Proof. ($\Rightarrow$) Convexity gives $f(x + t(y-x)) \le (1-t)f(x) + tf(y)$, so for $t \in (0,1]$

$$\frac{f(x + t(y-x)) - f(x)}{t} \ \le \ f(y) - f(x).$$

The left side tends to the directional derivative $\nabla f(x)^\mathsf{T}(y-x)$ as $t \downarrow 0$ (Lesson 7's definition of the gradient), and the right side does not move. ($\Leftarrow$) Let $z = tx + (1-t)y$ and write the tangent inequality at z twice: $f(x) \ge f(z) + \nabla f(z)^\mathsf{T}(x - z)$ and $f(y) \ge f(z) + \nabla f(z)^\mathsf{T}(y - z)$. Multiply by t and $1-t$ and add: the gradient terms cancel because $t(x-z) + (1-t)(y-z) = 0$, leaving $tf(x) + (1-t)f(y) \ge f(z)$. □

This one inequality is the engine of the whole part: it converts a *local* object (the gradient at x) into a *global* statement (a bound valid at every y). Everything that follows — global optimality, duality bounds, convergence proofs — is Theorem 53.4 wearing different clothes.

Theorem 53.5 (Second-order characterization)**.** *A twice-differentiable f on a convex domain is convex iff its Hessian satisfies $\nabla^2 f(x) \succeq 0$ everywhere.*

Proof. Restrict to a line: $g(t) = f(x + tv)$ has $g''(t) = v^{\mathsf{T}} \nabla^2 f(x + tv)\, v$. If $\nabla^2 f \succeq 0$ then every $g'' \geq 0$, so every restriction is a convex scalar function (its derivative is nondecreasing, which gives the tangent inequality by the mean value theorem), and convexity on every line is convexity. Conversely if $v^{\mathsf{T}} \nabla^2 f(x) v < 0$ for some x, v, the restriction along v is locally concave, violating the chord inequality nearby. □

So $f(x) = \frac{1}{2}\|Ax - b\|^2$ is convex — Hessian $A^{\mathsf{T}} A \succeq 0$ (Lesson 6) — and a quadratic $\frac{1}{2} x^{\mathsf{T}} Q x + c^{\mathsf{T}} x$ is convex exactly when $Q \succeq 0$, which the spectral theorem (Theorem 11.4) reads as "no negative eigenvalue directions."

Proposition 53.6 (The calculus of convexity)**.** *(a) Nonnegative combinations: if f_1, f_2 are convex and $\alpha, \beta \geq 0$, then $\alpha f_1 + \beta f_2$ is convex. (b) Affine substitution: if f is convex, so is $x \mapsto f(Ax + b)$. (c) Pointwise maximum: if $f_1, \dots, f_m$ are convex, so is $f(x) = \max_i f_i(x)$ — and likewise a supremum over an arbitrary family.*

Proof. (a) and (b) are direct substitutions into the chord inequality. For (c),

$$f(tx + (1-t)y) = \max_i f_i(tx + (1-t)y) \leq \max_i \big[t f_i(x) + (1-t) f_i(y)\big] \leq t f(x) + (1-t) f(y),$$

since a max of sums is at most the sum of maxes. □

Part (c) is the workhorse of signal processing models: a *worst-case* quantity — maximum passband ripple over frequencies, worst residual over measurements, largest eigenvalue of a symmetric matrix (a sup of the convex functions $x^{\mathsf{T}} P z \dots$ over unit z) — is convex *because* it is a sup of convex functions, even though it is not differentiable. Convex analysis is the calculus that survives corners.

Definition 53.7 (Convex optimization problem)**.**

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & Ax = b, \end{array}$$

with $f_0, \dots, f_m$ convex and the equality constraints affine. The feasible set is then convex (Proposition 53.2 plus the fact that sublevel sets $\{f_i \leq 0\}$ of convex functions are convex). Its optimal value is $p^\star$.

Theorem 53.8 (Local implies global; the optimality condition)**.** *For a convex problem: (a) any locally optimal x is globally optimal. (b) If f_0 is differentiable, a feasible $x^\star$ is optimal iff*

$$\nabla f_0(x^\star)^{\mathsf{T}} (y - x^\star) \;\geq\; 0 \qquad \textit{for every feasible } y.$$

(c) If moreover the problem is unconstrained, (b) reduces to $\nabla f_0(x^\star) = 0$.

Proof. (a) Suppose x is locally optimal but some feasible y has $f_0(y) < f_0(x)$. The segment $x_t = (1-t)x + ty$ is feasible (convex feasible set) and $f_0(x_t) \leq (1-t) f_0(x) + t f_0(y) < f_0(x)$ for every $t \in (0, 1]$ — arbitrarily close to x, contradicting local optimality. (b) If the condition holds, Theorem 53.4 gives $f_0(y) \geq f_0(x^\star) + \nabla f_0(x^\star)^{\mathsf{T}} (y - x^\star) \geq f_0(x^\star)$ for all feasible y: done. If it fails for some y, then moving from $x^\star$ toward y decreases f_0 to first order while staying feasible. (c) Take $y = x^\star - \nabla f_0(x^\star)$; the condition forces $-\|\nabla f_0(x^\star)\|^2 \geq 0$. □

Part (c) closes a loop: setting the gradient of $\frac{1}{2}\|Ax-b\|^2$ to zero gave the normal equations (Corollary 7.4), and convexity is the reason that stationary point was the *global* minimum — a fact Lesson 6 asserted geometrically and can now certify in one line.

Recognition is the skill. Almost no problem announces itself in the standard form; it is *rewritten* into it. Three rewrites cover a remarkable fraction of practice:

- *Epigraph trick* — minimize a max by naming the max: $\min_x \max_i |a_i^\mathsf{T} x - b_i|$ becomes $\min_{x,t} t$ subject to $-t \le a_i^\mathsf{T} x - b_i \le t$. The nondifferentiable objective became $2m$ linear inequalities and a linear objective: a linear program (LP). This is exactly how minimax (equiripple) FIR design is posed — Lesson 37's Parks–McClellan specification is an LP in the coefficients, since $H(e^{j\omega})$ is linear in h at each frequency.
- *Absolute values split* — $|u| \le t$ is the pair $u \le t$, $-u \le t$; so $\min \sum_i |a_i^\mathsf{T} x - b_i|$ and anything built from $\|\cdot\|_1$ or $\|\cdot\|_\infty$ is an LP.
- *Composition with affine maps* (Proposition 53.6b) — the logistic loss $\sum_i \log(1 + e^{-y_i a_i^\mathsf{T} w})$ is convex in w because $s \mapsto \log(1+e^{-s})$ is convex in the scalar s (second derivative $e^s/(1+e^s)^2 > 0$) and $s = y_i a_i^\mathsf{T} w$ is affine. The same two-line argument certifies least squares, ridge, LASSO, SVMs with hinge loss, and maximum-likelihood fitting of any log-concave model (BV §7.1) — which is why "the training problem is convex" was true for every classical ML model before deep networks.

ML / DSP / sensors connection. Convexity is the boundary between "run the solver and trust the answer" and "pray over the initialization." For classical ML it means the fitted model is a property of the data and the loss, not of the optimizer's starting point; for deep learning it is the lost paradise whose vocabulary (loss surfaces, saddle points, conditioning) still frames every training discussion. For DSP, the epigraph trick turns worst-case specs — ripple, overshoot, sidelobe level — into constraints a solver can certify, and Lesson 51's structured-matrix machinery is what makes those solvers fast. For sensors and estimation, Part IV's estimators reappear as convex programs: least squares (Theorem 6.4), LLMS (Theorem 22.4), and, in Lesson 58, detector and experiment design.

53.2 Practice: by hand

Example 53.9 (Chebyshev fit as an LP, end to end)**.** Fit a scalar line $y \approx c_1 u + c_0$ to $(u, y) \in \{(0,0),(1,1),(2,3)\}$, minimizing the *worst* error. Epigraph form: minimize t subject to $|c_0 - 0| \le t$, $|c_0 + c_1 - 1| \le t$, $|c_0 + 2c_1 - 3| \le t$. At the optimum the extreme residuals alternate in sign and tie (the equioscillation of Lesson 37, in miniature): try $c_1 = \frac{3}{2}$, $c_0 = -\frac{1}{4}$, giving residuals $-\frac{1}{4}, +\frac{1}{4}, -\frac{1}{4}$ — so $t = \frac{1}{4}$, and no line can do better because the middle point pulls opposite to the outer two. Compare least squares, which would trade a larger worst error for smaller total square error.

Example 53.10 (A convexity certificate in two lines)**.** Is $f(x_1, x_2) = e^{x_1} + e^{x_2} + (x_1 - x_2)^2$ convex? e^{x_i} are convex (second derivative $e^{x_i} > 0$), $(x_1 - x_2)^2$ is a convex quadratic composed with the affine map $x \mapsto x_1 - x_2$, and sums of convex functions are convex (Proposition 53.6a,b). No Hessian computation needed — the calculus of convexity is usually faster than the calculus of derivatives.

53.3 Exercises

Exercise 53.1 (Hand)**.** Which are convex? (a) $\{x \in \mathbb{R}^2 : x_1 x_2 \geq 1,\ x_1 > 0\}$; (b) $\{x : \|x\|_2 \geq 1\}$; (c) $\{x : \max(x_1, x_2) \leq 1\}$. (Answers: (a) yes — it is $\{x_1 > 0, x_2 \geq 1/x_1\}$ and $1/x_1$ is convex there, so this is a region above a convex curve's graph; check the chord inequality directly; (b) no — the segment from $(1, 0)$ to $(-1, 0)$ passes through the hole; (c) yes — intersection of two halfspaces.)

Exercise 53.2 (Hand)**.** Rewrite $\min_x \sum_{i=1}^k |a_i^\mathsf{T} x - b_i|$ as an LP with variables (x, u), $u \in \mathbb{R}^k$: state the objective and all $2k$ inequalities. Then rewrite $\min_x \|Ax - b\|_\infty$ the same way with a single extra scalar t.

Exercise 53.3 (Proof, $\star$)**.** Prove Jensen's inequality by induction from the definition: if f is convex, $\theta_i \geq 0$, $\sum_{i=1}^k \theta_i = 1$, then $f\big(\sum_i \theta_i x_i\big) \leq \sum_i \theta_i f(x_i)$; conclude $f(\mathbf{E}[X]) \leq \mathbf{E}[f(X)]$ for a discrete random variable X (write the expectation as a convex combination, Lesson 17). This inequality is the sole engine of Lesson 57's $D \geq 0$, and through it of essentially all of information theory — prove it once, carefully.

Exercise 53.4 (Proof)**.** Show that the sublevel set $\{x : f(x) \leq \alpha\}$ of a convex function is convex, and give a nonconvex function all of whose sublevel sets are convex ($f(x) = \sqrt{|x|}$), so the converse fails. (Such f are called quasiconvex; BV §3.4.)

Deeper reading. BV 2.1–2.3 and 2.5 (convex sets, separating hyperplanes), 3.1–3.2 (convex functions and the calculus that preserves them), 4.1–4.4 (problem classes: LP, QP, QCQP, SOCP).

54 Regularized Fitting and Inverse Problems

NEEDED FOR: ML (ridge, LASSO, robustness), DSP and imaging (deconvolution, restoration), sensors (calibration, fusion).

54.1 Theory

Lesson 51 diagnosed the disease: when A has small singular values, the least-squares solution multiplies measurement noise by $1/\sigma_i$ and the answer drowns (Capstone VII watched it happen to deconvolution). Regularization is the cure, and it is a *modeling* act, not a numerical trick: it states, in the objective, which solutions were plausible before the data arrived,

$$\min_x \ \tfrac{1}{2}\|Ax - y\|^2 + \lambda R(x),$$

with the data term demanding "explain the measurements," R pricing implausible answers, and λ the exchange rate between the two currencies.

Theorem 54.1 (Ridge regression: existence, uniqueness, and the SVD filter factors)**.** *For $\lambda > 0$ and $R(x) = \frac{1}{2}\|x\|^2$ (Tikhonov, Example 50.7), the minimizer is unique for every A:*

$$x_\lambda = (A^\mathsf{T} A + \lambda I)^{-1} A^\mathsf{T} y.$$

In the SVD $A = U\Sigma V^{\mathsf{T}}$ (Theorem 13.1) with singular values σ_i,

$$x_\lambda = \sum_i \underbrace{\frac{\sigma_i}{\sigma_i^2 + \lambda}}_{\textit{filter factor}} (u_i^{\mathsf{T}} y)\, v_i, \qquad \textit{versus} \qquad x_{\mathrm{LS}} = \sum_i \frac{1}{\sigma_i}(u_i^{\mathsf{T}} y)\, v_i.$$

Each data component $u_i^{\mathsf{T}} y$ is passed with gain $\sigma_i/(\sigma_i^2 + \lambda)$: $\approx 1/\sigma_i$ where $\sigma_i^2 \gg \lambda$ (strong directions untouched), $\approx \sigma_i/\lambda$ where $\sigma_i^2 \ll \lambda$ (weak directions muted instead of amplified). As $\lambda \downarrow 0$, x_λ tends to the minimum-norm least-squares solution; as $\lambda \to \infty$, to 0.

Proof. The objective is convex with Hessian $A^{\mathsf{T}}A + \lambda I \succ 0$ (add $\lambda > 0$ to each nonnegative eigenvalue of $A^{\mathsf{T}}A$), so it is strictly convex: the stationary point (Theorem 53.8c) is the unique global minimum, and setting the gradient $A^{\mathsf{T}}(Ax-y)+\lambda x$ to zero gives the formula. Substituting the SVD: $A^{\mathsf{T}}A + \lambda I = V(\Sigma^{\mathsf{T}}\Sigma + \lambda I)V^{\mathsf{T}}$ and $A^{\mathsf{T}}y = V\Sigma^{\mathsf{T}}U^{\mathsf{T}}y$, so the v_i coordinate of x_λ is $(\sigma_i^2 + \lambda)^{-1}\sigma_i(u_i^{\mathsf{T}}y)$. The two limits read off the filter factor: $\sigma_i/(\sigma_i^2 + \lambda) \to 1/\sigma_i$ for $\sigma_i > 0$ and $\to 0$ for $\sigma_i = 0$, which is exactly the minimum-norm pseudoinverse solution. □

The condition number of the ridge system is $(\sigma_{\max}^2+\lambda)/(\sigma_{\min}^2+\lambda)$ — capped near $\sigma_{\max}^2/\lambda$ no matter how singular A is. That is Definition 51.2's κ being *designed*, not suffered; the frequency-domain deconvolution of Capstone VII, with its damped gain $\overline{G}/(|G|^2 + \lambda)$, is Theorem 54.1 in the Fourier basis, where convolution's singular vectors are complex exponentials (Lesson 12).

Theorem 54.2 (Weighted least squares is Gaussian maximum likelihood)**.** *If $y = Ax + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \Sigma)$, $\Sigma \succ 0$, then the maximum-likelihood estimate of x is*

$$\hat{x} = \arg\min_x (Ax - y)^{\mathsf{T}}\Sigma^{-1}(Ax - y),$$

i.e. least squares in the whitened variables $\Sigma^{-1/2}y$, $\Sigma^{-1/2}A$.

Proof. The density of y given x (Lesson 23) is $c\exp\big(-\frac{1}{2}(y - Ax)^{\mathsf{T}}\Sigma^{-1}(y - Ax)\big)$ with c independent of x; maximizing the log-likelihood is minimizing the quadratic form. Factoring $\Sigma^{-1} = \Sigma^{-1/2}\Sigma^{-1/2}$ (spectral theorem, Theorem 11.4) exhibits it as $\|\Sigma^{-1/2}(Ax - y)\|^2$. □

So "weight each sensor by its inverse variance" is not a heuristic: it is the likelihood speaking. A high-noise sensor gets a small weight; *correlated* sensor errors get rotated apart before weighting. This is the estimation-side twin of Lesson 40's Wiener filter and the workhorse of every calibration and fusion problem downstream.

Priors, formally. Add a prior $x \sim \mathcal{N}(0, (\lambda^{-1})I \cdot s^2)$ and maximize the posterior instead: by Bayes' rule (Theorem 16.4) the log posterior is the log likelihood plus $-\frac{\lambda}{2s^2}\|x\|^2$ plus constants — ridge *is* Gaussian MAP estimation, and λ is a noise-to-prior-variance ratio, which is why good λ's track the noise floor (Exercise 54.4 makes this exact). Swap the Gaussian prior for a Laplacian $\propto e^{-\alpha\|x\|_1}$ and MAP becomes the LASSO.

The ℓ_1 penalty and why corners make zeros. With $R(x) = \|x\|_1 = \sum_i |x_i|$ the problem stays convex (Proposition 53.6) but loses differentiability at $x_i = 0$ — on purpose. Take the cleanest case $A = I$ (denoising): the problem separates into scalars,

$$\min_{x_i}\ \tfrac{1}{2}(x_i - y_i)^2 + \lambda|x_i|,$$

For $x_i > 0$ the derivative is $x_i - y_i + \lambda$, vanishing at $x_i = y_i - \lambda$, valid only if $y_i > \lambda$; symmetrically for $x_i < 0$; and $x_i = 0$ is optimal whenever the one-sided slopes disagree in sign, i.e. $|y_i| \le \lambda$. The solution is the *soft threshold*

$$x_i = \mathrm{soft}_\lambda(y_i) = \mathrm{sign}(y_i)\max(|y_i| - \lambda, 0) :$$

an entire interval of data maps to *exactly* zero. A smooth penalty can only shrink coordinates toward zero (ridge multiplies by $1/(1+\lambda)$, never reaching it); the corner is what makes sparsity an *output* of the optimization rather than a rounding step. Lesson 56 upgrades this scalar fact into an algorithm for general A.

Robustness is a loss choice. Squared error prices a residual r at r^2: one gross outlier outbids a hundred honest measurements. The ℓ_1 loss $\sum_i |r_i|$ prices linearly, so the fit resists outliers (its scalar optimum is a median, not a mean — Exercise 54.3); Huber's loss

$$\phi_{\mathrm{hub}}(r) = \begin{cases} r^2, & |r| \le M \\ M(2|r| - M), & |r| > M \end{cases}$$

is quadratic where the Gaussian story is believable and linear where it is not, and is convex (BV §6.1.2). Choosing among them is a statement about the failure modes of your sensor, encoded where the solver can act on it.

ML / imaging / sensors connection. Every entry of the modern fitting menu is this lesson's template with a different (R, loss) pair: ridge and LASSO regression, elastic net, total-variation denoising in imaging (Lesson 42's restoration problems with $R = \|\nabla x\|_1$), robust regression in computer vision, and the data-plus-prior structure of every Kalman update (Lesson 40's filter minimizes exactly a weighted-least-squares-plus-prior objective at each step). The bias–variance story is Lesson 51's κ made quantitative: raising λ buys conditioning with bias, and the U-shaped error curve of Capstone VII step 4 is the trade being paid.

54.2 Practice: by hand

Example 54.3 (Filter factors on a 2×2 problem)**.** $A = \mathrm{diag}(1,\ 0.1)$, $y = (1, 1)$, so $\sigma_1 = 1, \sigma_2 = 0.1$. Least squares: $x = (1, 10)$ — the weak direction blew up a unit measurement into 10. Ridge with $\lambda = 0.01$: filter factors $\frac{1}{1.01} \approx 0.99$ and $\frac{0.1}{0.02} = 5$, so $x_\lambda \approx (0.99,\ 5)$: the strong coordinate is barely touched, the weak one is halved. With $\lambda = 0.1$: $(0.91,\ 0.91)$ — now both are shrunk; λ has crossed σ_1^2's scale and is biasing directions the data actually supported. The art is $\sigma_{\min}^2 \ll \lambda \ll \sigma_{\max}^2$ when the spectrum allows it.

Example 54.4 (Two sensors, one constant)**.** Sensors report $y_1 = 10$ (variance 1) and $y_2 = 16$ (variance 9). Weighted least squares (Theorem 54.2) minimizes $(x-10)^2 + \frac{1}{9}(x-16)^2$, giving

$$\hat{x} = \frac{10 + 16/9}{1 + 1/9} = \frac{106}{10} = 10.6.$$

The clean sensor dominates 9 : 1 — and the posterior variance, $(1 + \frac{1}{9})^{-1} = 0.9$, is *smaller than either sensor's alone*: fusion helped even though the second sensor was nine times noisier. (Compare Lesson 22: this is LLMS estimation with the prior removed.)

54.3 Exercises

Exercise 54.1 (Hand)**.** For the 2×2 example above, find the λ minimizing the actual error $\|x_\lambda - x_{\text{true}}\|$ when $x_{\text{true}} = (1, 1)$ and y was generated with noise $(0, +0.9)$ added to Ax_{true} — i.e. the second measurement is mostly noise. (Set up x_λ by filter factors and search a few values: $\lambda \approx 0.02$–0.1 does well; the point is that the best λ is near the noise-to-signal scale, not near machine epsilon.)

Exercise 54.2 (Hand)**.** Soft-threshold the vector $y = (3, -0.4, 0.2, -2)$ with $\lambda = 0.5$ and with $\lambda = 1$; then apply ridge shrinkage $y/(1 + \lambda)$ with the same λ's. Which coordinates become exactly zero under each rule, and which merely small?

Exercise 54.3 (Hand)**.** Show that $\hat{c} = \arg\min_c \sum_i |y_i - c|$ is any median of $\{y_i\}$: compute the one-sided derivatives of the objective in c and find where the count of y_i above c balances the count below. This is why ℓ_1 data terms resist outliers: the optimum moves with the *ranks*, not the values.

Exercise 54.4 (Proof, $\star$)**.** Prove Theorem 54.1's filter-factor formula from scratch (substitute the SVD, use $V^\mathsf{T}V = I$), then prove the MAP claim: for $y = Ax + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, prior $x \sim \mathcal{N}(0, \tau^2 I)$, the posterior mode minimizes $\frac{1}{2\sigma^2}\|Ax - y\|^2 + \frac{1}{2\tau^2}\|x\|^2$, so $\lambda = \sigma^2/\tau^2$. Conclude: doubling the measurement noise should quadruple... no — work out exactly how λ should move, and why.

Deeper reading. BV 6.1–6.4 (norm approximation, least-norm, regularized and robust approximation — the Huber discussion is 6.1.2), 6.5 (function fitting), 7.1 (maximum likelihood, including logistic regression as a convex ML problem). For the numerical side, Lesson 51; for the statistical side, Lessons 22 and 23.

55 Duality, KKT Conditions, and Certificates

NEEDED FOR: ML (SVMs, kernels, sparsity certificates), communications (power allocation), sensors (resource trade-offs, constrained estimation).

55.1 Theory

A solver's answer to "minimize f_0" is a number $f_0(\hat{x})$ — an *upper* bound on $p^\star$, witnessed by $\hat{x}$. How would you ever know you are close, without knowing $p^\star$? You need lower bounds, and duality is a machine that manufactures them.

Definition 55.1 (Lagrangian and dual function)**.** For the problem of Definition 53.7 written with general constraints $f_i(x) \le 0$ $(i = 1, \dots, m)$ and $h_j(x) = 0$ $(j = 1, \dots, p)$, the *Lagrangian* is

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_j(x),$$

and the *dual function* is $g(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$, with $\lambda \succeq 0$ required and ν free.

Theorem 55.2 (Weak duality, and concavity of the dual)**.** *(a) For every $\lambda \succeq 0$ and every ν, $g(\lambda, \nu) \le p^\star$. (b) g is concave in (λ, ν) — always, even when the primal problem is not convex.*

Proof. (a) Let x be any feasible point. Then $\lambda_i f_i(x) \le 0$ (nonnegative multiplier times nonpositive constraint) and $\nu_j h_j(x) = 0$, so $L(x,\lambda,\nu) \le f_0(x)$. Hence $g(\lambda,\nu) = \inf_z L(z,\lambda,\nu) \le L(x,\lambda,\nu) \le f_0(x)$; take the infimum over feasible x. (b) For each fixed x, $L(x,\lambda,\nu)$ is *affine* in (λ,ν). So g is a pointwise infimum of affine functions, which is concave by (the concave version of) Proposition 53.6c. □

So every choice of nonnegative multipliers is a *certificate*: if a solver returns $\hat{x}$ and someone hands you (λ,ν) with $g(\lambda,\nu) = f_0(\hat{x}) - 10^{-9}$, then $\hat{x}$ is provably within 10^{-9} of optimal — no faith in the solver required. The *dual problem* is to find the best certificate: maximize $g(\lambda,\nu)$ over $\lambda \succeq 0$; its optimal value $d^\star \le p^\star$, and the gap $p^\star - d^\star$ is the *duality gap*.

Theorem 55.3 (Strong duality; Slater's condition)**.** *If the primal problem is convex and some strictly feasible point exists ($f_i(\tilde{x}) < 0$ for all i, $A\tilde{x} = b$), then $d^\star = p^\star$: the best certificate is exact. (Proof: BV §5.3.2, a separating-hyperplane argument; the geometry — the set of achievable (constraint, objective) pairs is convex, and a nonvertical supporting hyperplane at the optimum* is *the multiplier vector — is worth absorbing even without the proof.)*

Theorem 55.4 (KKT conditions)**.** *Suppose $f_0,\dots,f_m$ are differentiable. If strong duality holds and $x^\star, (\lambda^\star,\nu^\star)$ are primal and dual optimal, then*

$$\begin{aligned}
&\textit{(i) } f_i(x^\star) \le 0, \quad h_j(x^\star) = 0 && \textit{(primal feasibility)}\\
&\textit{(ii) } \lambda_i^\star \ge 0 && \textit{(dual feasibility)}\\
&\textit{(iii) } \lambda_i^\star f_i(x^\star) = 0 \ \textit{ for each } i && \textit{(complementary slackness)}\\
&\textit{(iv) } \nabla f_0(x^\star) + \textstyle\sum_i \lambda_i^\star \nabla f_i(x^\star) + \sum_j \nu_j^\star \nabla h_j(x^\star) = 0 && \textit{(stationarity).}
\end{aligned}$$

Conversely, for a convex *problem, any (x,λ,ν) satisfying (i)–(iv) is primal–dual optimal with zero gap: KKT points are certified optima.*

Proof. Forward direction: with zero gap,

$$f_0(x^\star) = g(\lambda^\star,\nu^\star) = \inf_z L(z,\lambda^\star,\nu^\star) \le L(x^\star,\lambda^\star,\nu^\star) = f_0(x^\star) + \sum_i \lambda_i^\star f_i(x^\star) \le f_0(x^\star),$$

using feasibility at the last step. So both inequalities are equalities. Equality in the second says $\sum_i \lambda_i^\star f_i(x^\star) = 0$; each term is ≤ 0, so each is 0: that is (iii). Equality in the first says $x^\star$ minimizes $L(\cdot,\lambda^\star,\nu^\star)$ over all z, and L is differentiable, so its gradient vanishes there: that is (iv). Converse: (iv) says x minimizes the convex function $L(\cdot,\lambda,\nu)$ (convex since $\lambda \succeq 0$ weights convex f_i), so $g(\lambda,\nu) = L(x,\lambda,\nu) = f_0(x) + \sum \lambda_i f_i(x) = f_0(x)$ by (iii). A feasible point whose objective equals a dual value is optimal by weak duality. □

Read (ii)–(iv) as mechanics. At the optimum, the objective's downhill force $-\nabla f_0$ is balanced by constraint forces $\lambda_i \nabla f_i$ pushing back along the walls' normals; a wall can only push ($\lambda_i \ge 0$), and only if you are touching it ($\lambda_i = 0$ whenever $f_i(x^\star) < 0$). Every fact about SVMs' support vectors, active sets, and sparse solutions is this picture.

Theorem 55.5 (Multipliers are prices)**.** *Let $p^\star(u)$ be the optimal value when the constraints are relaxed to $f_i(x) \le u_i$. For a convex problem with strong duality, $p^\star(u)$ is convex in u, and*

$$p^\star(u) \ \ge \ p^\star(0) - \lambda^{\star\mathsf{T}} u \qquad \textit{for all } u;$$

if $p^\star$ is differentiable at 0, then $\lambda_i^\star = -\partial p^\star / \partial u_i$.

Proof. Fix u and let x be feasible for the relaxed problem. Then $f_0(x) \geq L(x, \lambda^\star, \nu^\star) - \sum_i \lambda_i^\star f_i(x) \geq g(\lambda^\star, \nu^\star) - \lambda^{\star\mathsf{T}} u = p^\star(0) - \lambda^{\star\mathsf{T}} u$, using $f_i(x) \leq u_i$, $\lambda^\star \succeq 0$, and strong duality at $u = 0$. Minimize the left side over feasible x. The global affine lower bound touching at $u = 0$ forces the gradient claim when the derivative exists (and convexity of $p^\star$ follows by the usual perturbation argument, BV §5.6.1). □

This is the theorem that makes duality an *engineering* tool: $\lambda_i^\star$ is the exact marginal value of relaxing constraint i. If the power budget's multiplier is 0.3 bits per joule, a slightly bigger battery is worth exactly that much capacity — computed for free, from the solve you already did.

Water-filling, by hand. The classic worked KKT example, and one we will spend in Lesson 58: allocate power P across n parallel channels with noise levels N_i to maximize total capacity,

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n \log\bigl(1 + p_i/N_i\bigr) \\ \text{subject to} & p \succeq 0, \quad \mathbf{1}^\mathsf{T} p = P. \end{array}$$

KKT (with multiplier ν on the sum and $\mu_i \geq 0$ on $-p_i \leq 0$): stationarity gives $\frac{1}{N_i + p_i} = \nu - \mu_i$; complementary slackness kills μ_i wherever $p_i > 0$. So active channels satisfy $N_i + p_i = 1/\nu$ — a common water level — and channels with $N_i \geq 1/\nu$ get nothing:

$$p_i^\star = \max\bigl(1/\nu - N_i,\ 0\bigr), \qquad \nu \text{ chosen so } \textstyle\sum_i p_i^\star = P.$$

Pour water of total volume P into a tank whose floor is the noise profile N_i: it settles at a common surface, and channels whose floor pokes above the surface stay dry. One picture, one page of KKT, and it is optimal — certified, not plausible.

ML / communications / sensors connection. The SVM's margin problem has a dual whose variables live one-per-training-point; complementary slackness says only points touching the margin get $\lambda_i > 0$ — the *support vectors* — and the data enter the dual only through inner products, which is the door kernels walk through. The LASSO's optimality conditions certify which coefficients are zero (the dual constraint is slack there). In communications, water-filling above is how OFDM loads subcarriers (Lesson 41); in sensor design, multipliers price the watt/bit/false-alarm trade-offs that define an operating point — the sensitivity theorem is why those prices are believable numbers rather than folklore.

55.2 Practice: by hand

Example 55.6 (Projection onto the nonnegative axis, with forces)**.** Minimize $\frac{1}{2}(x-a)^2$ subject to $-x \leq 0$. Lagrangian $L = \frac{1}{2}(x-a)^2 - \lambda x$; KKT: $x - a - \lambda = 0$, $\lambda \geq 0$, $x \geq 0$, $\lambda x = 0$. Case $x > 0$: then $\lambda = 0$, $x = a$ — consistent iff $a > 0$. Case $x = 0$: then $\lambda = -a$ — consistent iff $a \leq 0$. So $x^\star = \max(a, 0)$ and $\lambda^\star = \max(-a, 0)$: the multiplier is precisely how hard the wall must push to stop the objective, and Theorem 55.5 predicts $dp^\star/du = -\lambda^\star$ — relaxing $x \geq 0$ to $x \geq -u$ helps at rate $|a|$ exactly when $a < 0$. Check it: for $a < 0$ and small $u \geq 0$, $p^\star(u) = \frac{1}{2}(a+u)^2$; differentiate at $u = 0$ and see.

Example 55.7 (An equality-constrained solve)**.** Minimize $x_1^2 + x_2^2$ subject to $x_1 + x_2 = 1$. Stationarity: $(2x_1, 2x_2) + \nu(1, 1) = 0$, so $x_1 = x_2 = -\nu/2$; feasibility forces $x_1 = x_2 = \frac{1}{2}$,

$\nu = -1$. The dual function: $g(\nu) = \inf_x\big(x_1^2 + x_2^2 + \nu(x_1 + x_2 - 1)\big) = -\frac{\nu^2}{2} - \nu$, maximized at $\nu = -1$ with $d^\star = \frac{1}{2} = p^\star$: zero gap, as Theorem 55.3 promised (affine constraints need no strict feasibility). Geometrically this is Lesson 6's projection of the origin onto a line, and ν is the length of the perpendicular force — duality keeps recovering Part I's geometry with prices attached.

55.3 Exercises

Exercise 55.1 (Hand)**.** Solve $\min \frac{1}{2}(x-3)^2$ s.t. $x \le 1$ by KKT (both cases), report $(x^\star, \lambda^\star)$, and verify the sensitivity prediction by computing $p^\star(u)$ for the relaxed constraint $x \le 1 + u$ and differentiating at $u = 0$. (Answer: $x^\star = 1$, $\lambda^\star = 2$, $p^\star(u) = \frac{1}{2}(2-u)^2$, $\frac{d}{du}p^\star(0) = -2$.)

Exercise 55.2 (Hand)**.** Water-fill $P = 6$ over noise floors $N = (1, 2, 5)$ (natural log): find $1/\nu$, the allocation, and which channel is dry. Then raise P until the dry channel first turns on. (Answer: try all-wet: $1/\nu = (6+8)/3 = 14/3$, $p = (11, 8, -1)/3$ — infeasible; two-wet: $1/\nu = (6+3)/2 = 4.5$, $p = (3.5, 2.5, 0)$, and channel 3 stays dry until $1/\nu$ reaches 5, i.e. $P = (5-1) + (5-2) = 7$.)

Exercise 55.3 (Proof, $\star$)**.** Prove weak duality and the concavity of g (Theorem 55.2) without looking, then derive the dual of the LP $\min c^\mathsf{T}x$ s.t. $Ax \le b$: show $g(\lambda) = -b^\mathsf{T}\lambda$ if $A^\mathsf{T}\lambda + c = 0$ and $-\infty$ otherwise, so the dual is $\max -b^\mathsf{T}\lambda$ s.t. $A^\mathsf{T}\lambda + c = 0$, $\lambda \succeq 0$. (LP duality, which Part I's Farkas-flavored geometry has been circling all along.)

Exercise 55.4 (Proof)**.** Verify that the water-filling allocation satisfies all four KKT conditions, and use Theorem 55.5 to interpret ν: show that $dC^\star/dP = \nu$, the marginal capacity per watt — the number a link designer actually quotes.

Deeper reading. BV 5.1–5.2 (dual function and problem), 5.3–5.4 (geometric and saddle-point views), 5.5 (KKT — water-filling is §5.5.3), 5.6 (perturbation and sensitivity), 5.7 (examples worth reading with a pencil).

56 Algorithms: Gradient, Newton, and Proximal Steps

NEEDED FOR: ML (training dynamics, conditioning), embedded DSP and sensors (real-time and large-scale solving).

56.1 Theory

Lessons 53–55 say what the answer means; this lesson is how machines find it, and — more valuable day to day — *why they are slow when they are slow.* The template is

$$x_{k+1} = x_k + t_k \Delta x_k,$$

a direction Δx_k and a step size t_k (chosen by a line search: backtrack from $t = 1$ until the decrease matches a fixed fraction of what the tangent predicted, BV §9.2). Gradient descent takes $\Delta x = -\nabla f(x)$ — Lesson 7's steepest descent. Its behavior is completely computable on the problem that locally approximates every smooth problem: a quadratic.

Theorem 56.1 (Gradient descent on a quadratic: mode-by-mode contraction)**.** *Let* $f(x) = \frac{1}{2}x^{\mathsf{T}}Qx - b^{\mathsf{T}}x$ *with* $Q \succ 0$*, eigenvalues* $0 < \lambda_{\min} = \lambda_1 \le \cdots \le \lambda_n = \lambda_{\max}$*, and minimizer* $x^\star = Q^{-1}b$*. With fixed step* t*, the error* $e_k = x_k - x^\star$ *obeys, in the eigenbasis of* Q *(Theorem 11.4),*

$$e_{k+1}^{(i)} = (1 - t\lambda_i)\, e_k^{(i)} :$$

each eigenmode contracts independently by its own factor. Convergence for all initializations iff $t < 2/\lambda_{\max}$*; the best fixed step* $t^\star = 2/(\lambda_{\min} + \lambda_{\max})$ *yields worst-mode factor*

$$\max_i |1 - t^\star\lambda_i| = \frac{\kappa - 1}{\kappa + 1}, \qquad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}.$$

Proof. $\nabla f(x) = Qx - b = Q(x - x^\star)$, so $e_{k+1} = e_k - tQe_k = (I - tQ)e_k$. Diagonalize $Q = U\Lambda U^{\mathsf{T}}$ and write e in the eigenbasis: the iteration matrix is diagonal with entries $1 - t\lambda_i$. All factors have modulus < 1 iff $0 < t\lambda_i < 2$ for every i, i.e. $t < 2/\lambda_{\max}$. The optimal t balances the two extreme modes, $1 - t\lambda_{\min} = -(1 - t\lambda_{\max})$, giving $t^\star$ and, on substitution, the factor $(\kappa - 1)/(\kappa + 1)$. □

Everything practical is in that factor. $\kappa = 10$: factor 0.82, fine. $\kappa = 10^4$ (a badly scaled feature matrix, Lesson 51): factor 0.9998 — ten thousand iterations per digit. And it is the *same* eigenvalue spread that throttles the LMS filter's adaptation (Lesson 40: LMS is exactly stochastic gradient descent on Lesson 22's MSE quadratic, and its modes converge at rates set by the input correlation matrix's eigenvalues). Preconditioning, feature normalization, momentum, whitening: all are attacks on κ. For general smooth strongly convex f ($mI \preceq \nabla^2 f \preceq MI$) the same linear convergence holds with $\kappa = M/m$ (BV §9.3) — the quadratic is not a toy, it is the local truth.

Newton's method. Gradient descent is blind to curvature; Newton's step

$$\Delta x_{\text{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

solves the local quadratic model exactly — on a true quadratic it lands on $x^\star$ in one step from anywhere ($x - Q^{-1}(Qx - b) = Q^{-1}b$).

Proposition 56.2 (Newton is affinely invariant)**.** *Let* T *be invertible and* $\tilde{f}(y) = f(Ty)$*. Newton iterates for* $\tilde{f}$ *started at* $y_0 = T^{-1}x_0$ *satisfy* $y_k = T^{-1}x_k$ *for all* k*: Newton's method* cannot see *linear changes of coordinates, so no preconditioning is needed and* κ *disappears from its convergence story.*

Proof. $\nabla \tilde{f}(y) = T^{\mathsf{T}}\nabla f(Ty)$ and $\nabla^2 \tilde{f}(y) = T^{\mathsf{T}}\nabla^2 f(Ty)\, T$, so

$$\Delta y = -\big(T^{\mathsf{T}}\nabla^2 f\, T\big)^{-1} T^{\mathsf{T}}\nabla f = -T^{-1}\nabla^2 f^{-1}\nabla f = T^{-1}\Delta x,$$

and induction carries the correspondence forward. □

The price is solving $\nabla^2 f\, \Delta x = -\nabla f$ each step — a structured linear system, which is where Lesson 51 earns its keep (Cholesky on the Hessian, or exploiting bandedness/circulance; BV Appendix C). Near the optimum Newton converges *quadratically* — digits double per iteration — which is why interior-point solvers finish in tens of iterations regardless of scale. With

equality constraints $Ax = b$, stationarity of the Lagrangian plus linearized feasibility give the *KKT system*

$$\begin{bmatrix} \nabla^2 f(x) & A^\mathsf{T} \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ w \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ Ax - b \end{bmatrix},$$

Lesson 55's balance of forces, refreshed at every iterate. Inequalities are handled by the *barrier method*: replace $f_i(x) \le 0$ by a cost $-\frac{1}{t}\sum_i \log(-f_i(x))$ that soars at the boundary, solve by Newton, increase t, repeat; the central path it follows has duality gap exactly m/t (BV §11.2), so the barrier's own multipliers certify how far you are from optimal at every moment. That is the algorithm inside CVXPY when Lesson 60 calls it.

Corners, handled exactly: proximal steps. For composite objectives $f(x) + \lambda\|x\|_1$ the gradient half is smooth and the corner half has a closed-form minimizer — Lesson 54's soft threshold. The *proximal gradient* method (ISTA) alternates them:

$$x_{k+1} = \mathrm{soft}_{\lambda t}\big(x_k - t\nabla f(x_k)\big),$$

a gradient step on the smooth part, then the exact scalar shrinkage on each coordinate. Because the threshold sets small coordinates to *exactly* zero, the iterates are genuinely sparse from early on — the algorithm output inherits the model's corner, which no amount of plain gradient descent on a smoothed $\|x\|_1$ would achieve. Convergence holds for $t \le 1/\lambda_{\max}(A^\mathsf{T}A)$ when $f = \frac{1}{2}\|Ax - y\|^2$, by the same eigenvalue logic as Theorem 56.1.

ML / DSP connection. Training a model *is* this lesson: SGD is Theorem 56.1 with noisy gradients (and Lesson 24's averaging is why the noise washes out); learning-rate limits are $t < 2/\lambda_{\max}$ wearing a tuning-guide costume; ill-conditioned features slow training by exactly $(\kappa - 1)/(\kappa + 1)$; batch normalization is preconditioning. On the DSP side, LMS (Lesson 40) is SGD avant la lettre, ISTA is how sparse deconvolution and compressed-sensing reconstructions actually run, and the KKT system's block structure is why real-time MPC and array-processing solvers can hit kilohertz rates on embedded hardware (Lesson 44's platforms).

56.2 Practice: by hand

Example 56.3 (Watching the slow mode)**.** $f(x) = \frac{1}{2}(10x_1^2 + x_2^2)$: $\lambda = (10, 1)$, $\kappa = 10$. From $x_0 = (1, 1)$ with $t = 0.1$: x_1-mode factor $1 - 1 = 0$, x_2-mode factor $1 - 0.1 = 0.9$. One step kills the stiff coordinate completely — and then the algorithm crawls along the valley floor at 0.9 per step: $x_k = (0,\, 0.9^k)$. The picture (steep walls, shallow floor) is the universal cartoon of ill-conditioned descent; optimal $t^\star = 2/11$ would balance the modes at factor $9/11 \approx 0.82$.

Example 56.4 (One proximal step, by hand)**.** $f = \frac{1}{2}\|x - y\|^2$, $y = (3, -0.4, 0.2)$, $\lambda = 1$, $t = 1$: gradient step lands on y itself, and $\mathrm{soft}_1(y) = (2,\, 0,\, 0)$. One iteration, two exact zeros, and the survivor pulled toward zero by exactly λt — denoiser, sparsifier, and proximal map are one operation.

56.3 Exercises

Exercise 56.1 (Hand)**.** For $f(x) = \frac{1}{2}(100x_1^2 + x_2^2)$: find the largest stable fixed step, the optimal fixed step, and the number of iterations of optimal-step gradient descent needed to shrink

the slow mode by 10^{-3}. Then report Newton's iteration count for the same task. (Answers: $t < 2/100$; $t^\star = 2/101$; factor $99/101$ needs $\approx \ln(10^{-3})/\ln(99/101) \approx 345$ steps; Newton: one.)

Exercise 56.2 (Hand)**.** Run two ISTA iterations by hand on $\min_x \frac{1}{2}(x-4)^2+|x|$ with $t = \frac{1}{2}$ from $x_0 = 0$, then solve the problem exactly by the soft-threshold formula and compare. (Iterates: $x_1 = \text{soft}_{1/2}(2) = 1.5$, $x_2 = \text{soft}_{1/2}(1.5 + \frac{1}{2}(4 - 1.5)) = \text{soft}_{1/2}(2.75) = 2.25$; exact answer $\text{soft}_1(4) = 3$, and the iterates are climbing toward it at rate $1 - t = \frac{1}{2}$.)

Exercise 56.3 (Proof, $\star$)**.** Prove that $\text{prox}_{\alpha|\cdot|}(z) = \arg\min_x \frac{1}{2}(x-z)^2+\alpha|x|$ is the soft threshold, by the three-case analysis of Lesson 54 — then prove the vector statement: for $R(x) = \alpha\|x\|_1$ the prox acts coordinatewise. (Separability of both the quadratic and the ℓ_1 norm is the whole proof; note where it would *fail* for $\|x\|_2$.)

Exercise 56.4 (Proof)**.** Prove Proposition 56.2's claim that gradient descent is *not* affinely invariant: exhibit the transformed gradient step and show it equals T^{-1} times the original step only when $TT^\mathsf{T} = I$. (Moral: gradient descent depends on the inner product you silently chose; Newton does not. Whitening data is choosing a better inner product.)

Deeper reading. BV 9.1–9.5 (descent, exact rates, Newton), 10.1–10.2 (equality-constrained Newton and the KKT system), 11.1–11.3 (barrier and central path); Appendix C for the linear algebra inside each iteration. Proximal methods are post-BV: this lesson's ISTA is the entry point, and Lesson 60 runs it.

57 Entropy, KL Divergence, and Mutual Information

NEEDED FOR: ML (cross-entropy losses, information gain), inference and sensors (uncertainty accounting), communications (the currency of Lessons 58–59).

57.1 Theory

Optimization measured cost in whatever units the objective carried. Information theory's first move is more radical: it puts a *unit on uncertainty itself*, and then proves that the unit is conserved, spent, and lost in lawful ways.

Definition 57.1 (Entropy)**.** For a discrete random variable X with pmf p (Lesson 17),

$$H(X) = -\sum_x p(x)\log_2 p(x) = \mathbf{E}\big[\log_2 \tfrac{1}{p(X)}\big] \quad \text{bits},$$

with $0\log 0 = 0$. $\log \frac{1}{p(x)}$ is the *surprise* of the outcome x — rare outcomes surprise more — and H is the average surprise, equivalently the average number of yes/no questions an optimal strategy needs to learn X (Lesson 59 makes "questions" into "code bits" exactly).

Definition 57.2 (KL divergence)**.** For pmfs p, q on the same alphabet,

$$D(P\|Q) = \sum_x p(x)\log\frac{p(x)}{q(x)} = \mathbf{E}_P\Big[\log\frac{p(X)}{q(X)}\Big],$$

($+\infty$ if p puts mass where q has none). It is the average log-likelihood advantage of the true model P over a rival Q, per observation — not a distance ($D(P\|Q) \neq D(Q\|P)$ in general), but the exact exchange rate at which data accumulates evidence, as Lesson 58 will prove.

Theorem 57.3 (Information inequality, and the two extremes of entropy)**.** *(a) $D(P\|Q) \geq 0$, with equality iff $P = Q$. (b) For an alphabet of size K: $0 \leq H(X) \leq \log_2 K$, with the maximum exactly at the uniform distribution and the minimum exactly at deterministic X.*

Proof. (a) $-D(P\|Q) = \sum_x p(x) \log \frac{q(x)}{p(x)} \leq \log \sum_x p(x) \frac{q(x)}{p(x)} \leq \log 1 = 0$, where the first step is Jensen's inequality (Exercise 53.3) applied to the concave log: $\mathbf{E}[\log Z] \leq \log \mathbf{E}[Z]$ with $Z = q(X)/p(X)$. Equality in Jensen for a strictly concave function forces Z constant, and a constant likelihood ratio with both sides summing to 1 forces $P = Q$. (b) Take Q uniform: $0 \leq D(P\|Q) = \log_2 K - H(X)$. And $H \geq 0$ because every term $-p \log p \geq 0$, vanishing only at $p \in \{0, 1\}$. □

One inequality, and it never stops paying: everything below — cross-entropy's lower bound, conditioning reducing entropy, data processing, channel converses, compression limits — is Theorem 57.3a applied to a well-chosen pair.

Cross-entropy, and what ML actually minimizes. Splitting the log in the *cross-entropy* $H(P, Q) = -\mathbf{E}_P[\log q(X)]$ gives

$$H(P, Q) = H(P) + D(P\|Q) \;\geq\; H(P).$$

Train a classifier by minimizing cross-entropy between the empirical labels and the model's predictive q: the $H(P)$ term is fixed by the data, so the training loss is *exactly* $D(P\|Q)$ plus a constant — the model is being fitted in KL, and maximum likelihood (Lesson 54, Capstone III) is the same statement read right to left. The loss on your training dashboard has been an information quantity all along.

Definition 57.4 (Conditional entropy and mutual information)**.** $H(X \mid Y) = \sum_y p(y) H(X \mid Y = y)$, the average residual uncertainty after seeing Y; and

$$I(X; Y) = D\big(P_{XY} \,\|\, P_X P_Y\big) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X).$$

The three faces are equal by expanding logs (Exercise 57.2 walks one identity), and each earns its keep: the KL form makes $I \geq 0$ automatic (Theorem 57.3a with the pair $(P_{XY}, P_X P_Y)$) and shows $I(X; Y) = 0$ iff X, Y independent (Lesson 18's product criterion, Theorem 18.4); the second face reads "information is uncertainty destroyed"; the third says the destruction is symmetric. Two corollaries fall out free: $H(X \mid Y) \leq H(X)$ — *conditioning reduces entropy on average* — and the chain rule $H(X, Y) = H(X) + H(Y \mid X)$ (split $\log \frac{1}{p(x,y)}$).

Theorem 57.5 (Data-processing inequality)**.** *If $X \to Y \to Z$ is a Markov chain (Lesson 26: Z depends on X only through Y), then*

$$I(X; Z) \;\leq\; I(X; Y).$$

No processing of Y — deterministic or random, clever or dumb — can increase the information it carries about X.

Proof. Expand $I(X; Y, Z)$ by the chain rule for mutual information ($I(X; Y, Z) = I(X; Z) + I(X; Y \mid Z) = I(X; Y) + I(X; Z \mid Y)$, each line the entropy chain rule applied twice). Markovity says X and Z are conditionally independent given Y, so $I(X; Z \mid Y) = 0$; and $I(X; Y \mid Z) \geq 0$ as a (conditional) mutual information. Therefore $I(X; Z) \leq I(X; Y)$. □

Theorem 57.6 (Fano's inequality)**.** *Let X take K values, let $\hat{X} = g(Y)$ be* any *estimator of X from Y, and let $P_e = \mathbf{P}(\hat{X} \neq X)$. Then*

$$H(X \mid Y) \;\leq\; h_2(P_e) + P_e \log_2(K-1).$$

Contrapositive, the useful direction: if $H(X \mid Y)$ is large, no algorithm — present or future — can guess X from Y reliably: $P_e \;\geq\; \big(H(X \mid Y) - 1\big)/\log_2 K$.

Proof. Let $E = \mathbf{1}\{\hat{X} \neq X\}$. Chain rule twice on $H(E, X \mid Y)$:

$$H(X \mid Y) \leq H(E, X \mid Y) = H(E \mid Y) + H(X \mid E, Y) \leq h_2(P_e) + H(X \mid E, Y).$$

Given $E = 0$, $X = g(Y)$ is determined: zero entropy. Given $E = 1$, X is one of the $K-1$ values other than $g(Y)$: entropy at most $\log_2(K-1)$ (Theorem 57.3b). So $H(X \mid E, Y) \leq (1 - P_e) \cdot 0 + P_e \log_2(K-1)$. □

Fano is the theorem that turns information accounting into *impossibility proofs*: compute (or bound) the information a measurement chain delivers, and Fano converts any shortfall into a floor under the error of every conceivable decoder. Lesson 58's channel converse is exactly this move.

Continuous variables. For a density f (Lesson 19), the *differential entropy* $h(X) = -\int f \log f$ plays the same algebraic role — KL, mutual information, chain rules, and DPI all go through with integrals — but h itself loses two comforts: it can be negative (a uniform on $[0, \frac{1}{2}]$ has $h = -1$ bit), and it changes under rescaling ($h(aX) = h(X) + \log_2 |a|$). Mutual information, being a KL, keeps all its properties and its invariance — the robust citizen of the continuous world. The one continuous entropy fact this booklet needs constantly: among all densities with variance σ^2, the Gaussian uniquely maximizes h, with $h = \frac{1}{2}\log_2(2\pi e\sigma^2)$ — proved in Lesson 58 (Lemma 58.6), spent there on AWGN capacity and in Lesson 59 on the Gaussian rate-distortion converse.

ML / inference / sensors connection. Cross-entropy loss *is* $H(P) + D(P\|Q)$; the information gain that splits a decision tree is $I(\text{feature}; \text{label})$; variational inference optimizes a KL by construction. In inference, D is the error exponent of Lesson 58 and the natural metric on model space (Fisher information is its curvature — Lesson 58). For sensors, mutual information is the honest scorecard: a sensor is worth what $I(\Theta; Y)$ says, DPI warns that downstream DSP can only spend that budget, never grow it, and Fano converts the budget into a floor on any detector's error — three sentences that justify an entire measurement-design methodology.

57.2 Practice: by hand

Example 57.7 (The binary entropy function)**.** $X \sim \text{Ber}(p)$: $H = h_2(p)$, with $h_2(\frac{1}{2}) = 1$ bit, $h_2(0.11) \approx 0.5$, $h_2(0.01) \approx 0.081$. Steep near the ends: certainty is cheap to dent. And $D(\text{Ber}(\frac{1}{4})\|\text{Ber}(\frac{1}{2})) = \frac{1}{4}\log_2\frac{1/4}{1/2} + \frac{3}{4}\log_2\frac{3/4}{1/2} = -\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{2} \approx 0.189$ bits: each coin flip of a quarter-biased coin builds 0.189 bits of evidence against the fair-coin hypothesis — a number Lesson 58 will convert into error rates.

Example 57.8 (A binary symmetric channel, fully accounted)**.** $X \sim \mathrm{Ber}(\frac{1}{2})$, $Y = X \oplus N$ with $N \sim \mathrm{Ber}(\epsilon)$ independent. $H(Y) = 1$ (still uniform by symmetry); $H(Y \mid X) = h_2(\epsilon)$ (given X, the only randomness is the flip). So

$$I(X;Y) = 1 - h_2(\epsilon) :$$

1 bit at $\epsilon = 0$, 0.5 bits at $\epsilon \approx 0.11$, 0 at $\epsilon = \frac{1}{2}$ — a coin-flip channel carries nothing, and (check $h_2(1) = 0$) a channel that *always* lies carries a full bit, because consistent lying is a code.

57.3 Exercises

Exercise 57.1 (Hand)**.** Compute, in bits: $H(X)$ for $p = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$; then design a prefix code (0, 10, 110, 111) and check its average length equals H exactly. (Answer: $H = 1.75$; dyadic probabilities are the case where Lesson 59's bound is met with no slack.)

Exercise 57.2 (Hand)**.** From the joint pmf $p(x, y)$ with rows $x \in \{0, 1\}$: $p(0,0) = \frac{1}{2}$, $p(0,1) = \frac{1}{4}$, $p(1,0) = 0$, $p(1,1) = \frac{1}{4}$, compute $H(X)$, $H(Y)$, $H(X,Y)$, $H(X \mid Y)$, and $I(X;Y)$, and verify both identities $I = H(X) - H(X \mid Y) = H(X) + H(Y) - H(X,Y)$. (Answers: $H(X) = h_2(\frac{1}{4}) \approx 0.811$, $H(Y) = 1$, $H(X,Y) = 1.5$, $H(X \mid Y) = 0.5$, $I \approx 0.311$.)

Exercise 57.3 (Proof, $\star$)**.** Prove $D(P\|Q) \geq 0$ and both extremes of entropy (Theorem 57.3) from Jensen alone, then prove the chain rule $H(X,Y) = H(X) + H(Y \mid X)$ and deduce $H(X_1, \ldots, X_n) \leq \sum_i H(X_i)$ with equality iff independent.

Exercise 57.4 (Proof)**.** Prove the DPI (Theorem 57.5) for the deterministic special case $Z = g(Y)$ directly from the chain-rule expansion, and give an example where processing *loses* information strictly: X uniform on $\{0, 1, 2, 3\}$, $Y = X$, $Z = Y \bmod 2$ — compute $I(X;Y)$ and $I(X;Z)$. (2 bits vs 1 bit.)

Deeper reading. PW 1.1 (entropy), 2.1–2.5 (divergence, chain rules, data processing — stated there for general alphabets and worth seeing in that generality), 3.1–3.6 (mutual information and Fano), 5.1 (convexity of information measures, the bridge to this part's optimization half).

58 Testing, Channels, and Sensor Information

NEEDED FOR: Communications (capacity, coding), inference (detection, estimation floors), sensors and radar (design as information maximization), ML (generalization's information bounds).

58.1 Theory

Lesson 41 built detectors — matched filter, threshold, CFAR — and measured them by SNR. This lesson supplies the deeper ledger: how fast evidence accumulates (divergence), the hard floor under every estimator (Fisher), and the ceiling on every channel and sensor (capacity). All three are the same mathematics.

Theorem 58.1 (Neyman–Pearson lemma)**.** *Observations $Y \sim P$ or $Y \sim Q$; a test decides. Among all tests with false-alarm probability $\mathbf{P}_Q(\text{decide } P) \le \alpha$, the likelihood-ratio test*

$$\text{decide } P \iff L(y) = \frac{p(y)}{q(y)} \ge \tau$$

(with τ set so the false-alarm constraint is met, randomizing on the boundary if needed) minimizes the miss probability $\beta = \mathbf{P}_P(\text{decide } Q)$.

Proof. Let $\phi^\star(y) \in \{0,1\}$ be the LRT's probability of deciding P, and ϕ any competitor. Pointwise,

$$\big(\phi^\star(y) - \phi(y)\big)\big(p(y) - \tau q(y)\big) \;\ge\; 0,$$

because where $p \ge \tau q$ the LRT has $\phi^\star = 1 \ge \phi$, and where $p < \tau q$ it has $\phi^\star = 0 \le \phi$. Sum (or integrate) over y:

$$\big[(1-\beta^\star) - (1-\beta)\big] - \tau\big[\alpha^\star - \alpha_\phi\big] \ge 0 \implies \beta - \beta^\star \;\ge\; \tau\,(\alpha^\star - \alpha_\phi) \;\ge\; 0$$

whenever the competitor spends no more false alarm than the LRT ($\alpha_\phi \le \alpha^\star$). Four lines, and detector design is over: only the threshold remains to be chosen. □

This is why Lesson 41's receivers all compute likelihood ratios — the matched filter is the LRT for a known signal in Gaussian noise (Example 50.6 derived it as a functional gradient; here it is optimal among *all* detectors, not just linear ones).

How fast does evidence accumulate? With n i.i.d. observations the log-likelihood ratio adds: $\log L_n = \sum_{k=1}^{n} \log \frac{p(Y_k)}{q(Y_k)}$. Under P, the WLLN (Theorem 24.2) drives $\frac{1}{n}\log L_n \to \mathbf{E}_P \log \frac{p}{q} = D(P\|Q)$: divergence *is* the per-sample drift of the evidence. The precise statement is:

Theorem 58.2 (Stein's lemma — KL is the testing exponent)**.** *For i.i.d. observations and any fixed false-alarm level $\alpha \in (0,1)$, the optimal miss probability satisfies*

$$-\frac{1}{n}\log \beta_n^\star \;\longrightarrow\; D(P\|Q)\,:$$

misses die like $e^{-nD(P\|Q)}$ (nats), no faster, no slower. (Proof: PW §14.5 — achievability is the LLN drift above pushing $\log L_n$ past any fixed threshold; the converse is a change-of-measure argument. Symmetrically, testing with both error exponents positive trades them along a curve whose corner cases are $D(P\|Q)$ and $D(Q\|P)$ — the Chernoff regime, PW ch. 16.)

So the asymmetric-looking KL has an exact operational meaning: $D(P\|Q)$ prices *mistaking P-data for Q*. For Gaussians of common variance, $D(\mathcal{N}(\mu_1,\sigma^2)\|\mathcal{N}(\mu_0,\sigma^2)) = \frac{(\mu_1-\mu_0)^2}{2\sigma^2}$ nats (Exercise 58.2) — one half the matched-filter SNR of Lesson 41: the two ledgers agree where they overlap, and KL keeps working where SNR stops meaning anything (non-Gaussian noise, unequal variances, discrete data).

The estimation floor. Testing bounds decisions; Fisher information bounds *continuous* estimates.

Definition 58.3 (Score and Fisher information)**.** For a family of densities $p_\theta(y)$ smooth in the parameter $\theta \in \mathbb{R}$, the *score* is $s_\theta(y) = \frac{\partial}{\partial\theta} \ln p_\theta(y)$ and the *Fisher information* is $J(\theta) = \mathbf{E}_\theta[s_\theta(Y)^2]$ — the mean squared sensitivity of the log-likelihood to the parameter. Under the usual regularity (differentiation through the integral, certified by Theorem 46.6), $\mathbf{E}_\theta[s_\theta(Y)] = \int \frac{\partial p_\theta}{\partial\theta}\, dy = \frac{\partial}{\partial\theta} \int p_\theta\, dy = 0$.

Theorem 58.4 (Cramér–Rao lower bound)**.** *Any unbiased estimator $\hat\theta(Y)$ ($\mathbf{E}_\theta\hat\theta = \theta$ for all θ) satisfies*

$$\operatorname{var}(\hat\theta) \;\geq\; \frac{1}{J(\theta)},$$

and for n i.i.d. observations, $\geq 1/(nJ(\theta))$.

Proof. Differentiate the unbiasedness identity $\int \hat\theta(y)\, p_\theta(y)\, dy = \theta$ in θ: $\mathbf{E}_\theta[\hat\theta\, s_\theta] = 1$. Since $\mathbf{E}s_\theta = 0$, this says $\operatorname{cov}(\hat\theta, s_\theta) = 1$. Cauchy–Schwarz in the covariance inner product (Theorem 21.3):

$$1 = \operatorname{cov}(\hat\theta, s_\theta)^2 \leq \operatorname{var}(\hat\theta)\, \operatorname{var}(s_\theta) = \operatorname{var}(\hat\theta)\, J(\theta).$$

Independence adds Fisher informations (the score of a product is a sum of scores, and cross-terms vanish by zero mean), giving the n-sample form. □

For $Y \sim \mathcal{N}(\theta, \sigma^2)$: $s_\theta = (y-\theta)/\sigma^2$, $J = 1/\sigma^2$, and the sample mean achieves σ^2/n exactly — the CRLB is tight, and Lesson 22's "the mean is the best estimator" becomes a theorem about *all* unbiased estimators, not just linear ones. Fisher information is also the local curvature of KL: $D(P_\theta \| P_{\theta+\delta}) = \frac{J(\theta)}{2}\delta^2 + o(\delta^2)$ (PW §2.6) — testing and estimation are priced by the same geometry, KL globally, Fisher infinitesimally. In sensor language: J is per-measurement resolving power, the CRLB is the spec-sheet floor no clever post-processing beats (that is the DPI, Theorem 57.5, in estimation clothing), and Lesson 40's Kalman filter is this accounting run recursively for linear-Gaussian dynamics.

Channels and capacity. A *channel* is a conditional distribution $P_{Y|X}$ — exactly what a sensor is: state in, noisy reading out. Using it n times to carry one of M messages at rate $R = \frac{1}{n}\log_2 M$ bits/use:

Theorem 58.5 (Shannon's channel coding theorem)**.** *For a memoryless channel, every rate below*

$$C = \max_{P_X} I(X;Y)$$

is achievable with error probability $\to 0$ as $n \to \infty$, and every rate above C forces error bounded away from zero. (PW chs. 18–19. The converse is Fano's inequality, Theorem 57.6, plus the DPI — information that never entered cannot be decoded out; the achievability is Shannon's random-coding argument, probabilistically brilliant and practically silent about which *code — sixty years of coding theory live in that gap.)*

Two capacities carry most of engineering. *BSC(ϵ)*: by Lesson 57's accounting $I(X;Y) = H(Y) - h_2(\epsilon) \leq 1 - h_2(\epsilon)$, achieved by uniform input, so $C = 1 - h_2(\epsilon)$ bits/use. *AWGN*, $Y = X + Z$, $Z \sim \mathcal{N}(0, N)$, power constraint $\mathbf{E}X^2 \leq P$: here the maximization needs the fact promised in Lesson 57 —

Lemma 58.6 (Gaussians maximize entropy)**.** *Among densities with variance* $\le \sigma^2$*,* $h(Y) \le \frac{1}{2}\log_2(2\pi e\sigma^2)$*, with equality only for* $\mathcal{N}(m,\sigma^2)$*.*

Proof. Let f have variance $\le \sigma^2$ (mean m, wlog $m = 0$) and g be the $\mathcal{N}(0,\sigma^2)$ density. Then

$$0 \le D(f\|g) = \int f \log_2 \frac{f}{g} = -h(Y) - \int f \log_2 g.$$

But $\log_2 g(y) = -\frac{1}{2}\log_2(2\pi\sigma^2) - \frac{y^2}{2\sigma^2}\log_2 e$ is a *quadratic* in y, so $-\int f \log_2 g$ depends on f only through its second moment, and is at most its value under g itself: $-\int f\log_2 g \le -\int g\log_2 g = \frac{1}{2}\log_2(2\pi e\sigma^2)$. Rearranged: $h(Y) \le \frac{1}{2}\log_2(2\pi e\sigma^2)$, with equality iff $D(f\|g) = 0$, i.e. $f = g$ (Theorem 57.3). □

Then $I(X;Y) = h(Y) - h(Y \mid X) = h(Y) - h(Z) \le \frac{1}{2}\log_2\big(2\pi e(P+N)\big) - \frac{1}{2}\log_2(2\pi eN)$, giving

$$C = \frac{1}{2}\log_2\Big(1 + \frac{P}{N}\Big) \text{ bits per real use,}$$

achieved by Gaussian input. Information grows *logarithmically* in power: the first watt is worth more than the next ten. For n parallel Gaussian channels with noises N_i and a total power budget — exactly Lesson 55's water-filling problem, whose KKT solution is now revealed as the capacity-achieving allocation. That is the two halves of Part VIII meeting: information theory says what the objective *means*; convex duality solves it and prices the budget.

Sensors as channels; measurements as experiments. A sensor suite maps state θ to readings Y through $P_{Y|\theta}$; "which measurements should we take?" becomes "maximize the information delivered." In the linear-Gaussian case $y_i = a_i^{\mathsf{T}}\theta + z_i$, both criteria of this lesson agree on the answer's shape: the Fisher information matrix is $J = \sum_i a_i a_i^{\mathsf{T}}/\sigma^2$, the CRLB generalizes to $\mathrm{cov}(\hat\theta) \succeq J^{-1}$, and with a Gaussian prior the mutual information is $I(\theta;Y) = \frac{1}{2}\log_2\det(I + \Sigma_\theta J)$ (Lesson 23's Gaussian machinery). Choosing how often to fire each candidate measurement to maximize $\log\det J$ is *D-optimal experiment design* — concave in the firing frequencies, hence a convex program (BV §7.5), solvable and certifiable by Lessons 55–56. Radar dwell scheduling, array geometry, calibration-sequence design: all instances.

Communications / radar / ML connection. Lesson 41's link budgets are this lesson quantified: BER curves are testing exponents, capacity is the ceiling that coding (LDPC, turbo) has now essentially reached, and $\frac{1}{2}\log_2(1 + \mathrm{SNR})$ is why every additional dB buys less throughput. Radar detection is Neyman–Pearson verbatim (CFAR sets τ adaptively to hold α). In ML, Fano-style arguments give minimax lower bounds — *no* learner beats them — and PW Part VI builds statistical learning theory on exactly the D, I, J triad of this lesson.

58.2 Practice: by hand

Example 58.7 (BSC capacities, felt)**.** $C(\epsilon) = 1 - h_2(\epsilon)$: $C(0) = 1$, $C(0.01) \approx 0.919$, $C(0.11) \approx 0.5$, $C(0.5) = 0$. An 11% bit-flip rate costs *half* the channel — noise is expensive out of all proportion to its rate, because the decoder must pay to locate the flips, not just count them.

Example 58.8 (Three cheap sensors vs one good one)**.** Binary state through BSC(0.1) sensors. One sensor: error 0.1; majority of three: $3(0.1)^2(0.9) + (0.1)^3 = 0.028$. Information accounting: one sensor delivers $1 - h_2(0.1) \approx 0.531$ bits; three deliver at most (and, being independent, exactly) $3\times$ the per-sensor mutual information $= 1.59$ bits about a 1-bit state — redundancy that the majority vote spends imperfectly (it discards the unanimity/2–1 distinction). The LRT uses all of it; with symmetric sensors majority *is* the LRT, and the gap to zero error is closed only exponentially, at Stein's rate nD.

Example 58.9 (AWGN arithmetic)**.** SNR $= 15$ (≈ 11.8 dB): $C = \frac{1}{2}\log_2 16 = 2$ bits/real use. To get 3 bits: SNR $= 63$, four times the power for one more bit. At low SNR, $C \approx \frac{\mathrm{SNR}}{2\ln 2}$ — linear in power, so spreading power over bandwidth is free there: the design logic of spread-spectrum and of every energy-starved sensor link.

58.3 Exercises

Exercise 58.1 (Hand)**.** Compute C for BSC(ϵ), $\epsilon \in \{0, 0.1, 0.5, 0.9\}$, and explain in one sentence why $\epsilon = 0.9$ matches $\epsilon = 0.1$. Then: how many uses of BSC(0.11) are needed, at minimum, to convey 500 bits reliably? (Answer: $C \approx \frac{1}{2}$, so ≥ 1000 uses — and Theorem 58.5 says some code gets arbitrarily close.)

Exercise 58.2 (Hand)**.** Show $D\big(\mathcal{N}(\mu_1, \sigma^2) \,\|\, \mathcal{N}(\mu_0, \sigma^2)\big) = \frac{(\mu_1 - \mu_0)^2}{2\sigma^2}$ nats by direct integration (the quadratic terms cancel; only the cross term survives). Conclude: a radar return displaced by one noise standard deviation builds evidence at $\frac{1}{2}$ nat per pulse, and Stein's lemma prices $\beta_n \approx e^{-n/2}$.

Exercise 58.3 (Proof, $\star$)**.** Prove the Cramér–Rao bound (Theorem 58.4) including the score's zero mean and the additivity of J over independent samples. Then compute J for $Y \sim \mathrm{Ber}(\theta)$ (answer: $J = \frac{1}{\theta(1-\theta)}$) and check that the sample frequency achieves it — the CRLB is tight for both of this booklet's favorite models.

Exercise 58.4 (Proof)**.** Prove Lemma 58.6 without looking. Then show where it was spent twice in this lesson (once on $h(Y)$, once — in Lesson 59 — on the error's entropy), and read PW §5.5 to state the *Gaussian saddle point*: for fixed signal and noise variances, Gaussian input is best and Gaussian noise is worst, so $\frac{1}{2}\log_2(1 + P/N)$ is simultaneously a guarantee against the worst noise and a ceiling for the best code — the reason link budgets may assume Gaussian everything without apology.

Deeper reading. PW 14.1–14.5 (Neyman–Pearson and Stein), 15.1–15.2 and ch. 16 (large deviations, Chernoff exponents), 19.1–19.3 (capacity and the coding theorem), 20.1–20.4 (Gaussian channels, water-filling), 2.6 and 29.1–29.2 (Fisher information and Cramér–Rao); BV 7.3 (detector design as an LP — worth seeing once) and 7.5 (experiment design). In Part VI: Lessons 41 and 40 are the practice this theory grades.

59 Rate-Distortion, Compression, and Representation Learning

NEEDED FOR: DSP and audio/image coding (quantizers, codecs), ML (bottlenecks, learned representations), sensors (bit budgets at the edge).

59.1 Theory

Lesson 57 measured information; this lesson spends it. Two regimes: lossless — reproduce exactly, pay H — and lossy — reproduce within distortion D, pay the rate-distortion function.

Theorem 59.1 (Kraft inequality)**.** *A binary prefix code (no codeword a prefix of another — decodable on the fly, as in Exercise 57.1) with lengths $l_1, \dots, l_K$ exists iff $\sum_i 2^{-l_i} \le 1$.*

Proof. ($\Rightarrow$) Map each codeword c of length l to the dyadic interval $[0.c,\ 0.c + 2^{-l}) \subset [0,1)$ of its binary expansion. The prefix property says no codeword's interval contains another's, so the intervals are disjoint, and their total length $\sum_i 2^{-l_i}$ fits in $[0,1)$. ($\Leftarrow$) Sort $l_1 \le \cdots \le l_K$ and greedily assign intervals left to right; the inequality guarantees room at every step, and dyadic alignment preserves the prefix property. (PW §10.3 draws the same proof on the binary tree.) □

Theorem 59.2 (Entropy is the price of lossless coding)**.** *For a source $X \sim p$ and any prefix code with lengths $l(x)$, the expected length obeys $\bar{L} = \mathbf{E}[l(X)] \ge H(X)$; and there is a prefix code (Shannon's $l(x) = \lceil \log_2 \frac{1}{p(x)} \rceil$) with $\bar{L} < H(X) + 1$. Coding blocks of n symbols squeezes the overhead to $\frac{1}{n}$: entropy is exactly the asymptotic bits-per-symbol cost.*

Proof. Lower bound: let $q(x) = 2^{-l(x)}/Z$ with $Z = \sum_x 2^{-l(x)} \le 1$ (Kraft). Then

$$\bar{L} - H(X) = \sum_x p(x)\Big[l(x) + \log_2 p(x)\Big] = \sum_x p(x) \log_2 \frac{p(x)}{q(x)} - \log_2 Z = D(P\|Q) - \log_2 Z \ \ge\ 0,$$

both terms nonnegative (Theorem 57.3; $-\log_2 Z \ge 0$ since $Z \le 1$). Upper bound: the ceiling lengths satisfy Kraft ($\sum 2^{-\lceil \log 1/p \rceil} \le \sum p = 1$), so the code exists, and each length exceeds $\log_2 \frac{1}{p(x)}$ by less than 1. □

Note what the proof *is*: a code is a probability model q in disguise ($q = 2^{-l}$), and its overhead over entropy is exactly $D(p\|q)$ — you pay KL for coding with the wrong model. That identity is the engine of arithmetic coding and of the compression-view of ML (a language model is a code; its log-loss *is* its compressed size; PW ch. 13). The block-coding claim is the AEP: by the WLLN (Theorem 24.2) applied to $\frac{1}{n} \log_2 \frac{1}{p(X_1)\cdots p(X_n)} \to H(X)$, long i.i.d. strings concentrate on $\approx 2^{nH}$ *typical* sequences of near-equal probability — index those and you are done (PW §11.2).

Quantization: analog sources meet finite bits. A b-bit uniform quantizer with step Δ rounds x to its bin center; modeling the error as uniform on $(-\frac{\Delta}{2}, \frac{\Delta}{2}]$ — honest when the signal moves through many bins per sample, dubious otherwise (Lesson 43 added dither precisely to launder this assumption) — gives

$$\mathbf{E}[e^2] = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2 \, de = \frac{\Delta^2}{12}.$$

Each extra bit halves Δ, quartering the error power: 6.02 dB per bit, the number behind every ADC spec (Lesson 38) and audio word-length argument (Lesson 43). But scalar rounding is not the end of the story — the true frontier is:

Definition 59.3 (Rate-distortion function)**.** For source X and distortion measure $d(x,\hat{x})$,

$$R(D) = \min_{P_{\hat{X}|X}:\, \mathbf{E}[d(X,\hat{X})]\le D} I(X;\hat{X}),$$

and Shannon's rate-distortion theorem (PW ch. 25) says $R(D)$ is exactly the minimal bits/symbol at which block coders can hit expected distortion D. It is the constrained-information twin of capacity: nonincreasing and convex in D, and $R(0) = H(X)$ for discrete sources — lossless coding is its endpoint.

Theorem 59.4 (Gaussian rate-distortion)**.** *For $X \sim \mathcal{N}(0,\sigma^2)$ with squared-error distortion,*

$$R(D) = \tfrac{1}{2}\log_2 \frac{\sigma^2}{D} \quad (0 < D \le \sigma^2), \qquad \textit{equivalently} \qquad D(R) = \sigma^2\, 2^{-2R}.$$

Proof of the converse (the achievability is a test-channel construction, PW §26.1). For any reproduction rule with $\mathbf{E}(X-\hat{X})^2 \le D$:

$$I(X;\hat{X}) = h(X) - h(X \mid \hat{X}) \;\ge\; h(X) - h(X-\hat{X}),$$

since conditioning reduces (differential) entropy and shifting by the known $\hat{X}$ does not change it. Now spend Lemma 58.6 on the error: $h(X-\hat{X}) \le \frac{1}{2}\log_2(2\pi e D)$. With $h(X) = \frac{1}{2}\log_2(2\pi e\sigma^2)$,

$$I(X;\hat{X}) \;\ge\; \tfrac{1}{2}\log_2 \frac{\sigma^2}{D}. \qquad \square$$

Halving the distortion always costs exactly half a bit per sample; 6 dB per bit again — but now as an *unbeatable* frontier rather than a quantizer model. The gap between a real scalar quantizer and $D(R)$ (a factor $\frac{\pi e}{6} \approx 1.53$ dB at high rate, with entropy coding) is the fee for coding samples one at a time; transform and vector coders exist to shrink it. For *correlated* Gaussian vectors, diagonalize the covariance (Theorem 23.3) and the problem splits into independent scalar sources with variances λ_i; the optimal bit allocation solves a convex problem whose KKT conditions are *reverse water-filling*: all components are coded to a *common* distortion level D^*, and components with $\lambda_i \le D^*$ get zero bits (PW §26.1) — Lesson 55's picture upside down, poured into eigenvalues instead of channels.

Transform coding, and why codecs look the way they do. Reverse water-filling is a recipe: (1) rotate into decorrelated coordinates — the KLT, which is PCA (Lesson 23) and is approximated at zero covariance-knowledge cost by the DCT for smooth signals (Lesson 42's JPEG, Lesson 43's MDCT); (2) quantize each coordinate to the common distortion level, spending $\frac{1}{2}\log_2(\lambda_i/D^*)$ bits where it is due and *zero* on weak components — which is why most of a JPEG block's coefficients are simply dropped; (3) entropy-code the result (Theorem 59.2 collects the last free bits). Every mainstream codec is these three steps plus perceptual weighting of the distortion measure.

Representation learning as rate-distortion. An autoencoder's bottleneck enforces a rate constraint; its reconstruction loss is a distortion; training sketches a point on an $R(D)$ curve for the data distribution — with the network's expressiveness standing in for Shannon's optimal

coder. The *information bottleneck* makes the analogy an objective: compress X into Z keeping what matters for a task Y,

$$\min_{P_{Z|X}} \; I(X;Z) - \beta\, I(Z;Y),$$

rate against *task-relevant* information rather than reconstruction. The DPI (Theorem 57.5) is the standing warning: $I(Z;Y) \le I(X;Y)$ always — representations budget information, they never mint it. And the granddaddy abstraction, *metric entropy* (PW ch. 27: the number of ϵ-balls needed to cover a function class, i.e. the bits needed to name a hypothesis to accuracy ϵ), is how these ideas price learning itself: classes that compress well generalize, a theme PW Part VI develops into theorems.

DSP / ML / sensors connection. Lesson 43's quantization-noise arithmetic, Lesson 38's oversampling-for-resolution trades, and Lesson 42's transform codecs are all points on or gaps from this lesson's curves — $\Sigma\Delta$ conversion, in this language, buys rate with bandwidth and spends it via noise shaping. At the sensing edge, $R(D)$ answers the design question "how many bits must the radio carry so the fusion center's estimate stays within spec?" before any hardware is chosen. In ML, the compression view of generalization — minimum-description-length, bottlenecks, the code = model identity in Theorem 59.2's proof — is the field's most durable explanation of why simple models learned from finite data can be trusted.

59.2 Practice: by hand

Example 59.5 (An ADC's bit budget)**.** A 12-bit ADC at 48 kHz emits 576 kbit/s raw. If the analog front end leaves only ≈ 8.5 effective bits (Lesson 44: noise floors are specs too), the honest information rate is ≈ 408 kbit/s, and entropy coding of the actual signal distribution recovers still more — transmitting raw ADC words ships noise at premium prices.

Example 59.6 (Gaussian $D(R)$ numbers)**.** $\sigma^2 = 1$: $R = 1$ bit/sample gives $D = 0.25$; $R = 2$ gives 0.0625. Real 2-bit scalar quantizers: the best possible (Lloyd–Max, with levels placed where the Gaussian lives) achieves ≈ 0.118, and the lazy uniform-over-$\pm 4\sigma$ of Capstone step 5 measures ≈ 0.34 — the frontier, the best scalar coder, and an actual design, in one column of numbers. To halve any distortion achieved *on* the frontier: half a bit, always.

Example 59.7 (Bernoulli rate-distortion endpoints)**.** For Ber(p), $p \le \frac{1}{2}$, Hamming distortion: $R(D) = h_2(p) - h_2(D)$ for $0 \le D < p$, and $R(D) = 0$ for $D \ge p$ — because guessing the constant 0 already achieves distortion p with zero bits. Sanity: $R(0) = h_2(p) = H(X)$, the lossless price, as promised.

59.3 Exercises

Exercise 59.1 (Hand)**.** For $p = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$ build the Shannon code $l = \lceil \log_2 1/p \rceil$, verify Kraft holds with equality, and compare $\bar{L}$ to H. Then perturb to $p = (0.45, 0.3, 0.15, 0.1)$: compute H, the Shannon lengths, and the overhead — and identify which term of Theorem 59.2's proof the overhead is. ($H \approx 1.79$; lengths $(2, 2, 3, 4)$; $\bar{L} = 2.35$; the overhead is $D(p\|2^{-l})$ plus the slack in Kraft.)

Exercise 59.2 (Hand)**.** A sensor node observes $\mathcal{N}(0, \sigma^2)$ samples and may transmit 3 bits/sample. What distortion does the frontier allow? What does a 3-bit uniform quantizer's $\Delta^2/12$ model

predict with range $\pm 4\sigma$? Explain the ordering. (Frontier: $\sigma^2 2^{-6} \approx 0.016\sigma^2$; uniform: $\Delta = \sigma$, $\Delta^2/12 \approx 0.083\sigma^2$ — the model is above the frontier, as it must be.)

Exercise 59.3 (Proof, $\star$)**.** Prove Theorem 59.4's converse from scratch, naming every inequality used (conditioning reduces entropy; translation invariance; Lemma 58.6). Then deduce the "half a bit per factor of two" rule and compute the bits per sample needed to reproduce a Gaussian source at 40 dB SNR. (10^4 ratio: $\frac{1}{2}\log_2 10^4 \approx 6.6$ bits.)

Exercise 59.4 (Proof)**.** Prove both directions of Kraft (Theorem 59.1) — draw the binary tree for $(1, 2, 3, 3)$ — and then show that *any* uniquely decodable code obeys the same inequality is harder (McMillan; PW §10.3): just verify the claim on the non-prefix but decodable code $\{0, 01, 011\}$.

Deeper reading. PW 10.1 and 10.3 (codes, Kraft, Huffman), 11.1–11.2 (block coding and the AEP), ch. 24 (quantization, including optimal and high-resolution theory), 25.1 and 26.1 (the rate-distortion theorem; Bernoulli and Gaussian evaluations, reverse water-filling), 27.1–27.2 and 27.6 (metric entropy and its rate-distortion connection); PW 4.8 and Part VI for the learning links. In Part VI: Lessons 38, 42, and 43 are where these curves meet silicon.

60 Capstone VIII: Optimize a Sensor Pipeline Under an Information Budget

NEEDED FOR: Synthesis of Part VIII: allocation (55), sparse recovery (54, 56), detection (58), estimation floors (58), and bit budgets (59), in NumPy.

One pipeline, five theorems exercised. Run it top to bottom; every printed number has a lesson that predicted it.

```
import numpy as np
from math import erfc, log, sqrt
rng = np.random.default_rng(8)

# ---- 1. water-filling a power budget (Lessons 55, 58) ----
Nfl = np.array([0.5, 1.0, 2.0, 4.0])          # noise floors
def waterfill(P):
    lo, hi = Nfl.min(), Nfl.max() + P
    for _ in range(60):                        # bisect the water level
        w = (lo + hi) / 2
        if np.maximum(w - Nfl, 0).sum() > P: hi = w
        else: lo = w
    p = np.maximum(w - Nfl, 0)
    return w, p, 0.5 * np.log2(1 + p / Nfl).sum()
w, p, C = waterfill(4.0)
print(p, C)                                    # (2, 1.5, 0.5, 0): ch4 dry
print((waterfill(4.01)[2] - C) / 0.01,         # marginal capacity per watt
      1 / (2 * np.log(2) * w))                 # = KKT multiplier's prediction

# ---- 2. sparse deconvolution by ISTA (Lessons 54, 56) ----
n = 300
h = np.exp(-0.5 * (np.arange(-10, 11) / 1.5) ** 2); h /= h.sum()
```

```
A = sum(hk * np.eye(n, k=k) for k, hk in zip(range(-10, 11), h))
x0 = np.zeros(n); x0[rng.choice(n, 10, replace=False)] = 3 * rng.standard_normal(10)
y = A @ x0 + 0.05 * rng.standard_normal(n)
t = 1 / np.linalg.norm(A, 2) ** 2            # step <= 1/lambda_max(A^T A)
soft = lambda z, a: np.sign(z) * np.maximum(np.abs(z) - a, 0)
x = np.zeros(n)
for _ in range(4000):
    x = soft(x - t * (A.T @ (A @ x - y)), 0.05 * t)
xr = np.linalg.solve(A.T @ A + 0.05 * np.eye(n), A.T @ y)   # ridge, same lam
print((x != 0).sum(), (np.abs(xr) > 1e-3).sum(),            # sparsity: 17 vs 297
      np.abs(x - x0).max(), np.abs(xr - x0).max())          # error: 0.52 vs 4.6

# ---- 3. evidence accumulates at rate D(P||Q) (Lessons 57, 58) ----
Phi = lambda z: 0.5 * erfc(-z / sqrt(2))      # N(0,1) cdf
for m in (10, 100, 1000):                     # P = N(1,1) vs Q = N(0,1)
    tau = 1.645 / sqrt(m)                     # NP threshold, alpha = 0.05
    beta = Phi((tau - 1) * sqrt(m))           # miss probability
    print(m, -log(beta) / m)                  # -> D = 1/2 nat (slowly!)

# ---- 4. the Cramer-Rao floor, audited (Lesson 58) ----
sig, m, T = 2.0, 25, 200000
est = (1.0 + sig * rng.standard_normal((T, m))).mean(axis=1)
print(est.var(), sig ** 2 / m)                # variance vs CRLB 1/(m J)

# ---- 5. quantize against the frontier (Lesson 59) ----
X = rng.standard_normal(400_000)
for b in (2, 4, 6, 8):
    K, D_ = 2 ** b, 8.0 / 2 ** b              # bins over +-4 sigma
    q = np.clip(np.floor((X + 4) / D_), 0, K - 1)
    mse = np.mean((X - ((q + 0.5) * D_ - 4)) ** 2)
    ph = np.bincount(q.astype(int), minlength=K) / len(X)
    Hq = -(ph[ph > 0] * np.log2(ph[ph > 0])).sum()
    print(b, mse, D_ ** 2 / 12, 2.0 ** (-2 * b), Hq)
    #      rate  actual  granular  Shannon D(R)  bits truly needed
```

Checkpoints (do all of these).

1. Step 1: verify the allocation $(2, 1.5, 0.5, 0)$ against your hand KKT solve (Exercise 55.2's method), and confirm the finite-difference marginal matches $1/(2\ln 2 \cdot w)$ — Theorem 55.5 as a printed number. Raise P until channel 4 wets and re-check.

2. Step 2: ISTA returns 17 nonzeros — the true 10-spike support, plus one-sample shoulders and a few tiny artifacts (LASSO's bias; a debiasing least-squares refit on the support removes it) — with *exact* zeros elsewhere; ridge returns 297 nonzeros and misses the spike heights by 4.6. Explain both from the corner geometry of Lesson 54 and the prox theorem (Exercise 56.3). Now double the blur width ($1.5 \to 3$): watch recovery degrade and connect to Theorem 54.1's filter factors — which singular values did the wider blur destroy?

3. Step 3: the sequence $-\frac{1}{m}\log\beta_m$ climbs toward $D = \frac{1}{2}$ nat but is visibly below it at $m = 1000$. The gap is the $O(1/\sqrt{m})$ CLT correction (Theorem 24.3; in coding it is called channel dispersion, PW §22.5) — estimate the deficit's $1/\sqrt{m}$ coefficient from your three data points.

4. Step 4: the audited variance should sit within a percent of $\sigma^2/m = 0.16$. Now square the samples first (estimate θ from Y_i^2's mean): the estimator is still unbiased for $\mathbf{E}Y^2$, but for θ it is not even consistent — write one sentence on why the CRLB applies to estimators of θ, not to whatever your pipeline happens to output. (DPI: processing cannot add Fisher information either.)

5. Step 5: three curves, one story. `mse` hugs $\Delta^2/12$ (at $b = 2$ the agreement is luck — the wide bins violate the granular model, but clipping loss and the peaked density cancel; check by widening the range to $\pm 6\sigma$), both sit a constant factor above Shannon's 2^{-2b}, and $H_q < b$ says entropy coding would recover real bits (≈ 0.95 at $b = 8$). State which theorem owns each column, and where the $\frac{\pi e}{6}$ scalar-coding gap shows up.

6. Close the loop: the pipeline measured, in order, a certified allocation, a sparse inverse solution, an evidence rate, an estimation floor, and a rate-distortion gap. For each, name the Part VIII lesson and the single inequality (weak duality; prox optimality; Jensen via $D \geq 0$; Cauchy–Schwarz; $D \geq 0$ again through Lemma 58.6) doing the work. Two inequalities carried the entire part — say which two.

Final self-check list for Part VIII

From memory: define convex sets and functions and prove the first-order characterization; recognize LS, ridge, LASSO, logistic regression, Chebyshev and equiripple filter design as convex problems via the three standard rewrites; prove local-implies-global; derive the ridge filter factors from the SVD and state what λ trades; write the Lagrangian, prove weak duality, state Slater, derive and interpret all four KKT conditions, and solve water-filling by hand; explain why multipliers are prices; compute gradient descent's contraction factor from eigenvalues and connect it to LMS; state why Newton is affinely invariant and what the prox of $\|\cdot\|_1$ is; define H, D, I, prove $D \geq 0$ from Jensen and data processing from the chain rule; state Fano and what it forbids; prove Neyman–Pearson and Cramér–Rao and state Stein; compute the BSC and AWGN capacities and prove the Gaussian max-entropy lemma they lean on; prove Kraft and $\bar{L} \geq H$, derive $\Delta^2/12$, state $R(D)$, and prove the Gaussian rate-distortion converse; explain reverse water-filling, transform coding, and the information bottleneck in one paragraph each. The part in one sentence: *convex duality certifies the best design the constraints allow, and information theory certifies the best any design could ever do — an engineer should always know both numbers.*