Manifolds and Generative AI
A Mathematical Foundation

Vu Hung Nguyen

2025-11-04

Contents

Preface

This book introduces manifolds, geodesics, and the core ideas of Riemannian and information geometry with an intuition-first approach. We begin with simple, concrete examples (lines, circles, planes, spheres), then gradually build the formal tools (tangent spaces, metric tensor, connections, curvature) and show how these concepts power modern generative AI (VAEs, GANs, diffusion, flows).

Who this book is for. Readers in machine learning, data science, and applied mathematics who want a compact, visual route from geometry to practical AI. The text favors clarity over maximal generality; we keep proofs light and provide references for deeper study.

What you need. A working knowledge of linear algebra and multivariable calculus is sufficient; light familiarity with probability and optimization is helpful. No prior differential geometry is assumed. A concise checklist of prerequisites and notation conventions is included in the Overview, and a quick summary appears in the project README.

How to read this book. Part I develops manifold intuition and examples; Part II introduces geodesics and distance; Part III formalizes local neighborhoods with open n-balls; Chapter 9 provides a self-contained introduction to Riemannian geometry (tangent spaces, metrics, Levi-Civita connection, parallel transport, curvature); Chapter 10 connects geometry to generative AI with statistical/parameter manifolds, Fisher information, natural gradient, and Bregman divergences. Appendices collect summaries and metaphors; the glossary provides quick definitions.

Why geometry for AI. Geometry offers principled, invariant ways to measure change, distance, and curvature. These ideas translate directly into stable training (via Fisher geometry and natural gradients), meaningful interpolation (geodesics), and better understanding of model behavior. Our goal is to make these connections precise yet accessible.

Using the materials. You can build the PDF with the provided Makefile and find a release-ready PDF in the `release/` directory. A high-level synopsis—including target audience, prerequisites, and chapter summaries—is available in the README; consult it as a quick guide to the structure and intended outcomes of the book.

Acknowledgments. Many classic texts and modern papers inform this presentation; we include citations and links in the References for further reading.

Chapter 1

Overview

This book introduces the concepts of manifolds and geodesics in an intuitive, accessible way, building from simple examples to understanding how these mathematical foundations power modern generative artificial intelligence. Whether you're interested in geometry, navigation, or AI, you'll discover how curved spaces and shortest paths connect to real-world applications.

Remark 1.1: What is a curved space?

Informally, a *curved space* is one that is not globally flat like Euclidean space. Locally it may look flat (small neighborhoods resemble $\mathbb{R}^n$), but globally straight lines bend and familiar rules change: parallel lines can meet (on a sphere) or spread apart faster than in the plane (on a saddle). Mathematically, curvature is captured by the Riemannian metric and its curvature tensor; nonzero curvature means the geometry deviates from Euclidean.

1.1 Why Manifolds Matter

Imagine standing on Earth's surface. To you, the ground appears flat—you can use a local map as if you were on a plane. Yet we know Earth is a sphere. This intuition—that curved spaces look flat when you zoom in—is the essence of a **manifold**.

Manifolds are everywhere:

- **Physical spaces**: Earth's surface, the universe (in general relativity), molecular configurations
- **Data spaces**: Images, text embeddings, audio signals—all lie on low-dimensional manifolds embedded in high-dimensional spaces
- **AI applications**: Generative models learn to represent and sample from these data manifolds

Understanding manifolds helps us navigate curved spaces, compress high-dimensional data, and build better AI systems. This book provides the mathematical foundation to understand these connections.

1.2 The Journey: From Geometry to AI

We'll build understanding progressively, starting with intuition and concrete examples, then developing systematic methods, and finally connecting to modern AI applications.

1.2.1 Part I: Foundations (Chapters 1-4)

We begin with the core concept: a **manifold** is a space that locally looks like flat Euclidean space. Chapter 1 introduces this idea with visual analogies and the concept of **charts**—mappings that allow us to work with curved spaces locally as if they were flat.

Chapters 2 and 3 explore concrete examples: 1D manifolds (lines and circles) and 2D manifolds (planes, spheres, tori). We'll see how distance calculations work on these spaces and understand why local flatness is preserved even when the global structure is curved.

Chapter 4 examines a counterexample—why a cone's tip is not a manifold—to clarify the definition through contrast. Understanding what fails to be a manifold deepens our grasp of what makes a space a manifold.

1.2.2 Part II: Geodesics and Distance (Chapters 5-7)

On a flat plane, the shortest path between two points is a straight line. On curved surfaces, we need **geodesics**—the shortest paths that stay on the surface. Chapter 5 introduces this concept with intuitive examples: great circle routes on Earth, paths on terrain, and the "string pulled tight" analogy.

Chapter 6 develops systematic methods for finding geodesics: the **variational approach** (minimizing path length) and solving the **geodesic equation**. We'll derive geodesics on lines, circles, planes, and spheres, seeing how geometry determines the shortest paths.

Chapter 7 deepens our understanding of distance calculations, particularly on spheres. We'll explore alternative formulas (like the haversine formula), handle special cases, and discuss practical applications in navigation and mapping.

1.2.3 Part III: Mathematical Foundations (Chapter 8)

Chapter 8 explores **open n-balls**—the fundamental neighborhoods that define what it means for a space to "look like" Euclidean space locally. Understanding open n-balls makes the manifold definition precise and mathematically rigorous.

1.2.4 Part IV: Applications in Generative AI (Chapter 10)

Chapter 10 bridges the mathematical foundations with modern generative AI. We'll see how:

- **Data manifolds**: High-dimensional data (images, text) lies on low-dimensional manifolds
- **VAEs**: Use encoder-decoder pairs as charts mapping between data manifolds and latent spaces
- **GANs**: Learn generators that implicitly define data manifolds

- **Diffusion models**: Move data off and back onto manifolds during generation
- **Geodesic interpolation**: Provides natural paths for smooth transitions in latent spaces

The mathematical concepts we've built—manifolds, geodesics, distances—become concrete tools for understanding and building AI systems.

1.2.5 Part V: Reference (Chapter 11)

Chapter 11 provides a glossary of key terms and concepts for quick reference throughout your reading.

1.3 Key Concepts You'll Master

By the end of this book, you'll understand:

- **Manifolds**: Spaces that are locally flat but globally curved, with charts providing local coordinate systems
- **Geodesics**: The shortest paths on curved surfaces, generalizing straight lines to curved spaces
- **Distance**: How to measure distances on manifolds, particularly on spheres (great circle distance)
- **Local structure**: Open n-balls and neighborhoods that define manifold structure
- **AI connections**: How manifolds describe data structure and how generative models learn to represent them

1.4 Learning Approach

This book is designed for accessibility:

- **Visual learning**: TikZ diagrams throughout help build geometric intuition
- **Concrete examples**: We start with familiar examples (Earth, circles, spheres) before abstracting
- **Progressive complexity**: Each chapter builds on previous concepts
- **Worked examples**: Detailed calculations show how to apply the concepts
- **Practical applications**: Real-world connections from navigation to AI

1.5 Prerequisites

This section is a short checklist of the math background helpful for reading this book. It is not a full course; we provide the geometric details where they first appear (see Chapter 9 for Riemannian geometry and Chapter 10 for information geometry).

1.5.1 Notation and conventions

- **Vectors, matrices, tensors**: Bold for vectors $\mathbf{x}$, uppercase for matrices $\mathbf{A}$; indices x^i, matrix entries A_{ij}.
- **Inner products and norms**: $\langle u, v\rangle$, $\|v\|^2 = \langle v, v\rangle$.
- **Derivatives**: Gradient ∇f, Jacobian $\mathbf{J}_F$, Hessian $\mathbf{H}_f$.
- **Probability**: $\mathbb{E}[\cdot]$, $\mathrm{Var}[\cdot]$, $\mathrm{KL}(p\|q)$.

1.5.2 Linear algebra (core)

Vector spaces and bases; orthonormality; eigenvalues/eigenvectors; positive (semi)definite matrices; SVD and projections; Jacobian/Hessian as matrices/tensors. Used in pullback metrics and curvature approximations (see Chapter 10: Parameter Manifolds, Fisher).

1.5.3 Calculus and multivariable calculus

Partial derivatives and chain rule; gradients/Hessians; Taylor expansions; arc length as an integral. Used for geodesic length and local KL expansions (see Chapter 9: Distances from the Metric; Chapter 10: Fisher).

1.5.4 Ordinary differential equations (light)

Initial value problems and basic numerical integration intuition. Used in geodesic ODEs and parallel transport (see Chapter 9: Geodesics, Parallel Transport).

1.5.5 Probability and statistics (core)

Expectations/variance; log-likelihood and score; common families (Bernoulli, Gaussian). Used in Fisher information and ELBO (see Chapter 10: Statistical Manifolds; Fisher Information Matrix).

1.5.6 Information theory and divergences

Entropy, cross-entropy, KL; Bregman divergences and Legendre duality (at a glance). Used in KL $\approx$ Fisher locally and loss design (see Chapter 10: Bregman Divergence, Natural Gradient).

1.5.7 Optimization (core)

Gradient descent; conditioning and preconditioning; Gauss–Newton vs Hessian; trust regions. Used in geometry-aware training (see Chapter 10: Natural Gradient; Parameter Manifolds).

1.5.8 Differential geometry (orientation only)

Curves and tangent vectors as velocities; coordinate bases; Riemannian metric and line element $ds^2 = g_{ij}dx^i dx^j$; Christoffel symbols and the idea of geodesics. Full treatment in Chapter 9.

Minimal expectation. Comfort with basic linear algebra and multivariable calculus is sufficient. No prior knowledge of manifolds or differential geometry is required.

1.6 The Big Picture

This book connects abstract mathematics to practical applications:

- **From geometry to navigation**: Understanding geodesics helps explain why airplanes follow great circle routes
- **From mathematics to data**: The manifold structure of data explains why dimensionality reduction works
- **From theory to AI**: Geometric foundations enable better generative models
- **From intuition to rigor**: We build both geometric intuition and mathematical precision

The journey from understanding what a manifold is to seeing how it powers generative AI is not just possible—it's essential for anyone working at the intersection of mathematics and modern machine learning.

1.7 How to Use This Book

- **Sequential reading**: Chapters build on each other, so reading in order is recommended
- **Visual aids**: Pay attention to the TikZ diagrams—they're designed to build intuition
- **Examples**: Work through the examples to see concepts in action
- **Glossary**: Use Chapter 11 as a reference when you encounter unfamiliar terms
- **Active learning**: Try to visualize the concepts and connect them to familiar examples

1.8 What Lies Ahead

As you progress through this book, you'll gain:

- A solid mathematical foundation in manifolds and geodesics
- Geometric intuition for curved spaces
- Practical skills for distance calculations and geodesic finding
- Deep insights into how generative AI models work
- A foundation for advanced topics in differential geometry and Riemannian geometry

The mathematical concepts we explore—though abstract at first—have become central to understanding both classical geometry and modern AI. Welcome to the journey from curved spaces to intelligent systems.

Chapter 2

What is a Manifold?

A **manifold** is a space that **locally looks like flat Euclidean space**, even if the overall shape is curved or complex. The idea is that if you zoom in very closely on any small neighborhood in a manifold, it looks like regular n-dimensional space.

Definition 2.1: Topological Manifold

A *topological n-manifold* is a second-countable, Hausdorff topological space $\mathcal{M}$ such that every point $p \in \mathcal{M}$ has an open neighborhood U and a homeomorphism (chart) $\varphi : U \to V \subset \mathbb{R}^n$. An *atlas* is a collection of charts whose domains cover $\mathcal{M}$. If all transition maps $\varphi_j \circ \varphi_i^{-1}$ between overlapping charts are C^k (typically $k = \infty$), $\mathcal{M}$ is a *C^k (smooth) manifold.*

Definition 2.2: Chart and Atlas

A *chart* on $\mathcal{M}$ is a pair (U, φ) with $U \subset \mathcal{M}$ open and $\varphi : U \to \varphi(U) \subset \mathbb{R}^n$ a homeomorphism. An *atlas* is a family of charts whose domains cover $\mathcal{M}$; it is *smooth* when all chart transitions are smooth.

Imagine standing on Earth. To you, the ground seems flat locally (like a plane), but we know Earth is actually a curved sphere globally. The small patch you see is like a flat 2D plane, even though the whole planet curves.

2.1 The Key Concept: Local Mapping to Flat Space

The fundamental property of a manifold is that every point has a neighborhood that can be mapped to a flat Euclidean space. This mapping, called a **chart** or **coordinate chart**, allows us to work with the manifold locally as if it were flat.

This mapping property is what makes manifolds so useful: we can use familiar flat space mathematics (like calculus, geometry, and coordinates) locally on the manifold, even though the manifold itself is curved globally.

Example 2.1: Examples of manifolds

To better understand manifolds, let's visualize some key examples.

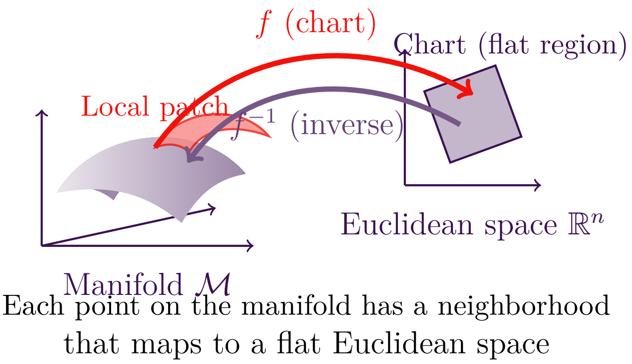
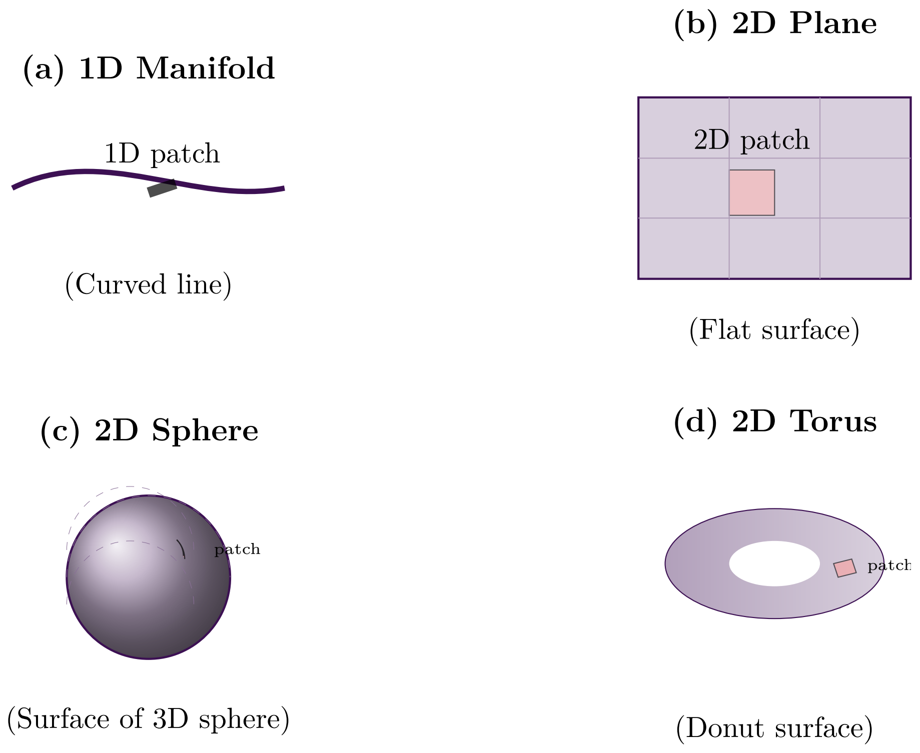


Figure 2.1: The fundamental concept of a manifold: every point has a local neighborhood (patch) that can be mapped to flat Euclidean space via a chart f. The inverse map f^{-1} allows us to work with coordinates on the flat space and transfer them back to the manifold. This local flatness property holds even though the manifold may be globally curved.



(a) 1D Manifold

(Curved line)

(b) 2D Plane

(Flat surface)

(c) 2D Sphere

(Surface of 3D sphere)

(d) 2D Torus

(Donut surface)

Example 2.2

Examples of manifolds: (a) a 1D manifold (curved line), (b) a 2D plane, (c) a 2D sphere surface, and (d) a 2D torus. Each manifold locally looks like flat Euclidean space of the appropriate dimension, as indicated by the highlighted patches.

These visualizations illustrate the key concept: each manifold, regardless of its global shape, has the property that any small neighborhood (patch) looks like flat Euclidean space of the corresponding dimension. The 1D manifold looks like a line locally, the 2D manifolds look like a plane locally, even though globally they may be curved or have complex topology.

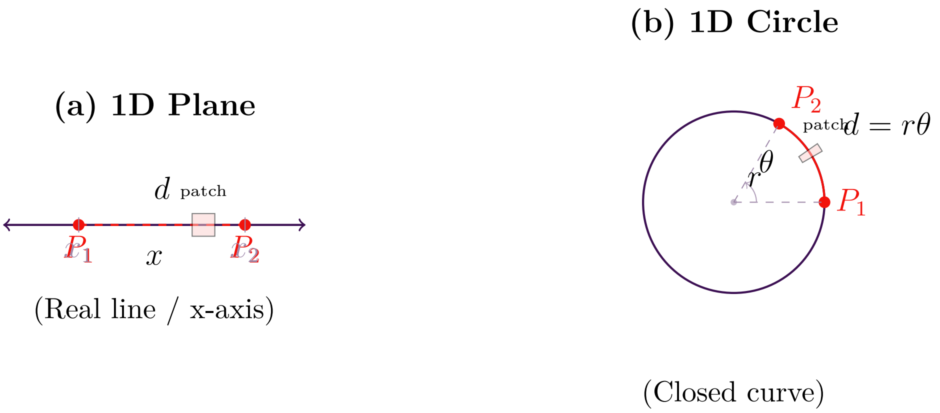
Chapter 3

1D Manifold Examples

Examples of 1D manifolds include a straight line or a circle. Each small segment of a circle (like a tiny arc) looks like a straight line segment when zoomed in enough. But the circle itself loops back, making it a 1D manifold that is "curved" globally.

We explore these examples in detail, showing how local flatness is preserved even when the global structure is curved or closed.

3.1 Visualizing 1D Manifolds



(Real line / x-axis)

(Closed curve)

Example 3.1

Examples of 1D manifolds: (a) a 1D plane (the real line or x-axis), and (b) a 1D circle. Both manifolds locally look like a line segment, but the circle has a closed, curved global structure.

Example 3.2: Circle distance on S^1

Let a circle have radius r and two points with angles θ_1, θ_2. Denote the small angle difference by $\Delta\theta = |\theta_2 - \theta_1|$ with $\Delta\theta \ll 1$.
- Arc-length (geodesic) distance:

$$d_{\text{arc}} = r\,\min(\Delta\theta, 2\pi - \Delta\theta) = r\,\Delta\theta \quad (\text{since } \Delta\theta \ll 1).$$

- Chord distance:

$$d_{\text{chord}} = 2r\sin\left(\frac{\Delta\theta}{2}\right) = r\,\Delta\theta\left(1 - \tfrac{(\Delta\theta)^2}{24} + \cdots\right).$$

Hence $d_{\text{arc}} \approx r\,\Delta\theta$ and the gap to the chord shrinks cubically:

$$d_{\text{arc}} - d_{\text{chord}} = r\left(\Delta\theta - 2\sin\tfrac{\Delta\theta}{2}\right) \approx r\,\frac{(\Delta\theta)^3}{24} \to 0 \quad \text{as } \Delta\theta \to 0.$$

Interpretation: for very close angles, the circle is *locally flat*. The geodesic distance along the circle behaves like the Euclidean distance on a straight line, with arclength coordinate $s = r\theta$. In particular, on the unit circle ($r = 1$), $d_{\text{arc}} \approx |\theta_2 - \theta_1|$.

3.2 Distance Calculations on 1D Manifolds

3.2.1 Distance on a 1D Plane (x-axis)

On a straight line (the x-axis), calculating the distance between two points is straightforward. Given two points P_1 at position x_1 and P_2 at position x_2, the distance is simply the absolute difference:

$$d = |x_2 - x_1|$$

This is the familiar Euclidean distance formula in one dimension.

Example 3.3

If P_1 is at $x_1 = -1$ and P_2 is at $x_2 = 1.2$, then

$$d = |1.2 - (-1)| = |2.2| = 2.2.$$

3.2.2 Distance on a 1D Circle

For a circle of radius r, calculating the distance between two points requires more consideration. The distance along the circle depends on the angle between the two points.

Exact Distance Using Arc Length

Given two points P_1 and P_2 on a circle, we can represent their positions using angles θ_1 and θ_2 measured from a reference axis. The distance along the circle (arc length) is:

$$d = r \cdot \min(|\theta_2 - \theta_1|, 2\pi - |\theta_2 - \theta_1|)$$

where we take the minimum to get the shorter arc between the two points. The angle difference $\theta = |\theta_2 - \theta_1|$ is measured in radians.

For small angles, we can approximate this as:

$$d \approx r \cdot \theta$$

where θ is the angle between the two points in radians.

Using Cartesian Coordinates

If we know the Cartesian coordinates of the points on the circle, we can calculate the distance using trigonometric functions. For a circle centered at the origin with radius r, if point P_1 is at $(r\cos\theta_1, r\sin\theta_1)$ and P_2 is at $(r\cos\theta_2, r\sin\theta_2)$, then:

$$\theta = \arccos\left(\frac{P_1 \cdot P_2}{r^2}\right) = \arccos(\cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2)$$

Using the angle addition formula, this simplifies to:

$$\theta = \arccos(\cos(\theta_2 - \theta_1)) = |\theta_2 - \theta_1|$$

And the distance is:

$$d = r \cdot |\theta_2 - \theta_1|$$

Example 3.4: Circle distance on S^1

Consider a circle with radius $r = 1.2$ units. If point P_1 is at angle $\theta_1 = 0$ (or 0°) and point P_2 is at angle $\theta_2 = \frac{\pi}{3}$ (or 60°), then:

$$\theta = \left|\frac{\pi}{3} - 0\right| = \frac{\pi}{3} \text{ radians}$$

Using the approximate formula $d = r \cdot \theta$:

$$d = 1.2 \times \frac{\pi}{3} = 1.2 \times 1.047 \approx 1.256 \text{ units}$$

For comparison, the straight-line (chord) distance would be:

$$d_{\text{chord}} = \sqrt{(r\cos\theta_2 - r\cos\theta_1)^2 + (r\sin\theta_2 - r\sin\theta_1)^2}$$

$$d_{\text{chord}} = r\sqrt{(\cos\theta - 1)^2 + \sin^2\theta} = r\sqrt{2 - 2\cos\theta} = 2r\sin\left(\frac{\theta}{2}\right)$$

For $\theta = \frac{\pi}{3}$:

$$d_{\text{chord}} = 2 \times 1.2 \times \sin\left(\frac{\pi}{6}\right) = 2.4 \times 0.5 = 1.2 \text{ units}$$

Note that the arc distance ($d = 1.256$) is slightly longer than the chord distance ($d_{\text{chord}} = 1.2$), which is expected since the arc follows the curve of the circle.

Key Takeaways 1

- A 1D plane (like the x-axis) is an open manifold where distance is simply the absolute difference between coordinates.
- A 1D circle is a closed manifold where distance is calculated using arc length: $d = r \cdot \theta$ for angle θ in radians.
- For small angles on a circle, the arc distance formula $d \approx r \cdot \theta$ provides a good approximation.
- The arc distance is always greater than or equal to the straight-line (chord) distance between two points on a circle.

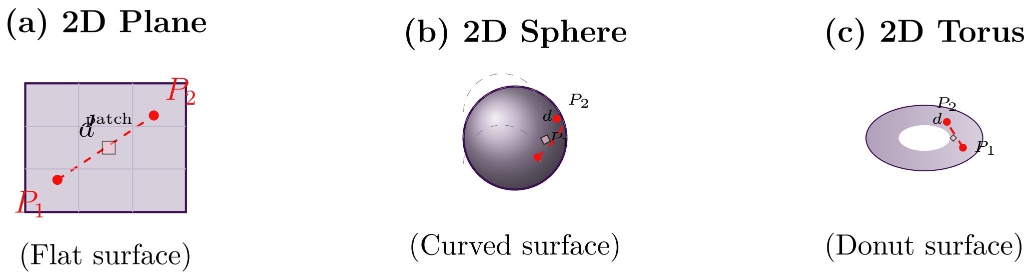
Chapter 4

2D Manifold Examples

Examples of 2D manifolds include a flat plane, the surface of a sphere (like Earth), or the surface of a donut (torus). Each tiny patch on these surfaces looks like a flat 2D disk, but the whole shape can be curved or looped in complex ways.

We examine each of these examples, showing how they satisfy the definition of a manifold while exhibiting different global structures.

4.1 Visualizing 2D Manifolds


(a) 2D Plane

(b) 2D Sphere

(c) 2D Torus

(Flat surface)

(Curved surface)

(Donut surface)

Figure 4.1: Examples of 2D manifolds: (a) a 2D plane (flat surface), (b) a 2D sphere (curved surface), and (c) a 2D torus (complex curved surface). Each manifold locally looks like a flat 2D disk, as indicated by the highlighted patches.

4.2 Distance Calculations on 2D Manifolds

4.2.1 Distance on a 2D Plane

The simplest 2D manifold is the flat plane, which we can think of as the familiar xy-plane. On a plane, calculating the distance between two points is straightforward using the Euclidean distance formula.

Definition 4.1: Distance between two points in $\mathbb{R}^2$

Given two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ on the Euclidean plane, their distance is

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

This is the familiar Pythagorean theorem extended to two dimensions. The distance is the length of the straight line segment connecting the two points.

Example 4.1: Euclidean distance on a plane

Consider two points on a plane: $P_1 = (1, 2)$ and $P_2 = (4, 6)$. The distance between them is:

$$d = \sqrt{(4-1)^2 + (6-2)^2} = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$$

Intuition

On a flat plane, the shortest path between two points is always a straight line. This is the fundamental property of Euclidean geometry. The plane has zero curvature, meaning it's "flat" everywhere, and there's no "bending" of space that would make curved paths shorter.

4.2.2 Distance on a 2D Sphere

The surface of a sphere is a 2D manifold that is curved. Unlike the plane, the shortest path between two points on a sphere is not a straight line (which would cut through the sphere), but rather a **great circle arc**—the intersection of the sphere with a plane passing through the center and both points.

Formula: Great Circle Distance

Given two points on a sphere, we can represent them using spherical coordinates. If we have two points with latitude and longitude (ϕ_1, λ_1) and (ϕ_2, λ_2), or equivalently, their unit vectors from the center, the distance along the sphere's surface (great circle distance) is:

$$d = R \cdot \arccos(\vec{v}_1 \cdot \vec{v}_2)$$

where R is the radius of the sphere, and $\vec{v}_1$ and $\vec{v}_2$ are unit vectors pointing to the two points from the sphere's center.

In terms of spherical coordinates:

$$d = R \cdot \arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\lambda_2 - \lambda_1))$$

Definition 4.2: Haversine formula

Let two points on a sphere of radius R have latitudes ϕ_1, ϕ_2 and longitude difference $\Delta\lambda = \lambda_2 - \lambda_1$. Define the haversine $\operatorname{hav}(\theta) = \sin^2(\frac{\theta}{2})$. The central angle $\Delta\sigma$ satisfies

$$\operatorname{hav}(\Delta\sigma) \;=\; \operatorname{hav}(\phi_2 - \phi_1) \;+ \cos\phi_1 \cos\phi_2 \; \operatorname{hav}(\Delta\lambda).$$

Equivalently,

$$\Delta\sigma \;=\; 2 \,\arcsin\left(\sqrt{\sin^2\frac{\phi_2 - \phi_1}{2} \;+ \cos\phi_1 \cos\phi_2 \, \sin^2\frac{\Delta\lambda}{2}}\right),$$

and the great-circle distance is

$$d \;=\; R\,\Delta\sigma \;=\; 2R \,\arcsin\left(\sqrt{\sin^2\frac{\phi_2 - \phi_1}{2} \;+ \cos\phi_1 \cos\phi_2 \, \sin^2\frac{\Delta\lambda}{2}}\right).$$

Simplified Case: Points on the Equator

For points on the equator (where $\phi_1 = \phi_2 = 0$), the formula simplifies to:

$$d = R \cdot |\lambda_2 - \lambda_1|$$

where λ is the longitude difference in radians. This is similar to the 1D circle case we saw earlier.

Example 4.2: Sphere distance on the equator

Consider a sphere with radius $R = 1$ unit. Two points are located at:

- P_1: latitude 0°, longitude 0° (on the equator)
- P_2: latitude 0°, longitude 60° (also on the equator, 60° away)

Using the simplified formula:

$$d = 1 \cdot \left|\frac{\pi}{3} - 0\right| = \frac{\pi}{3} \approx 1.047 \text{ units}$$

For a more general case, if P_1 is at $(0°, 0°)$ and P_2 is at $(30°, 60°)$, we use the full formula with $\phi_1 = 0$, $\phi_2 = \frac{\pi}{6}$, $\lambda_1 = 0$, $\lambda_2 = \frac{\pi}{3}$:

$$d = 1 \cdot \arccos\left(\sin(0)\sin\left(\frac{\pi}{6}\right) + \cos(0)\cos\left(\frac{\pi}{6}\right)\cos\left(\frac{\pi}{3}\right)\right)$$

$$d = \arccos\left(0 \cdot \frac{1}{2} + 1 \cdot \frac{\sqrt{3}}{2} \cdot \frac{1}{2}\right) = \arccos\left(\frac{\sqrt{3}}{4}\right) \approx 0.722 \text{ radians} \approx 1.0 \text{ units}$$

Example 4.3: Real-World: Sydney to New York

Let's calculate the great circle distance between two real-world locations on Earth:

- **Opera House Sydney**: latitude $\phi_1 = -33.8567°$, longitude $\lambda_1 = 151.2151°$
- **Bow Bridge, New York**: latitude $\phi_2 = 40.7757°$, longitude $\lambda_2 = -73.9718°$

The Earth's radius is approximately $R = 6371$ km. First, we convert the coordinates to radians:

$$\begin{aligned}
\phi_1 &= -33.8567° \times \frac{\pi}{180} \approx -0.5903 \text{ radians} \\
\lambda_1 &= 151.2151° \times \frac{\pi}{180} \approx 2.6390 \text{ radians} \\
\phi_2 &= 40.7757° \times \frac{\pi}{180} \approx 0.7108 \text{ radians} \\
\lambda_2 &= -73.9718° \times \frac{\pi}{180} \approx -1.2905 \text{ radians}
\end{aligned}$$

Now we calculate the great circle distance using the formula:

$$d = R \cdot \arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\lambda_2 - \lambda_1))$$

Substituting the values:

$$\begin{aligned}
\sin\phi_1 &= \sin(-0.5903) \approx -0.5564 \\
\sin\phi_2 &= \sin(0.7108) \approx 0.6522 \\
\cos\phi_1 &= \cos(-0.5903) \approx 0.8309 \\
\cos\phi_2 &= \cos(0.7108) \approx 0.7580 \\
\lambda_2 - \lambda_1 &= -1.2905 - 2.6390 = -3.9295 \text{ radians} \\
\cos(\lambda_2 - \lambda_1) &= \cos(-3.9295) \approx -0.7077
\end{aligned}$$

Plugging into the formula:

$$\begin{aligned}
d &= 6371 \cdot \arccos((-0.5564)(0.6522) + (0.8309)(0.7580)(-0.7077)) \\
&= 6371 \cdot \arccos(-0.3629 - 0.4456) \\
&= 6371 \cdot \arccos(-0.8085) \\
&= 6371 \cdot 2.5130 \\
&\approx 16,015 \text{ km}
\end{aligned}$$

So the great circle distance between Sydney Opera House and Bow Bridge in New York is approximately **16,015 kilometers** (about 9,946 miles). This is the shortest path along the Earth's surface—the route that airplanes would follow for the most efficient flight.

Intuition

On a sphere, the shortest path between two points is always along a great circle—the largest circle that can be drawn on the sphere. This is why airplanes flying long distances appear to follow curved paths on a flat map: they're actually following great circle routes, which are the shortest paths on the Earth's spherical surface.

The sphere has positive curvature, meaning it curves "outward" everywhere. This curvature makes the geometry different from the flat plane, and distances are measured along the curved surface rather than through space.

4.2.3 Distance on a 2D Torus

A torus (the surface of a donut) is a more complex 2D manifold. Unlike the sphere, which has constant positive curvature, a torus has regions of both positive and negative curvature, and its geometry is more complicated.

Approximate Formula

For a torus, calculating the exact distance between two points is more complex because the shortest path (geodesic) can wrap around the torus in different ways. However, we can approximate the distance for nearby points.

Consider a torus parameterized by two angles (θ, ϕ), where:

- θ is the angle around the "tube" (the smaller circle)
- ϕ is the angle around the "hole" (the larger circle)

The torus has two radii: R (the distance from the center of the torus to the center of the tube) and r (the radius of the tube itself).

For two nearby points $P_1 = (\theta_1, \phi_1)$ and $P_2 = (\theta_2, \phi_2)$ on a torus, an approximate distance formula is:

$$d \approx \sqrt{(R + r\cos\theta_1)^2(\phi_2 - \phi_1)^2 + r^2(\theta_2 - \theta_1)^2}$$

This approximation works well when the points are close together and the angle differences are small.

Simplified Approximation for Small Distances

For very small distances, we can use a simpler approximation. If the points are close enough that the torus looks locally flat, we can use:

$$d \approx \sqrt{(R + r)^2(\Delta\phi)^2 + r^2(\Delta\theta)^2}$$

where $\Delta\phi = \phi_2 - \phi_1$ and $\Delta\theta = \theta_2 - \theta_1$ are small angle differences.

Example 4.4: Approximate distance on a torus

Consider a torus with $R = 2$ (major radius) and $r = 1$ (minor radius). Two points are located at:

- P_1: $(\theta_1 = 0, \phi_1 = 0)$
- P_2: $(\theta_2 = \frac{\pi}{6}, \phi_2 = \frac{\pi}{12})$ (small angles)

Using the simplified approximation:

$$d \approx \sqrt{(2+1)^2 \left(\frac{\pi}{12}\right)^2 + 1^2 \left(\frac{\pi}{6}\right)^2}$$

$$d \approx \sqrt{9 \cdot \left(\frac{\pi}{12}\right)^2 + \left(\frac{\pi}{6}\right)^2} = \sqrt{\frac{9\pi^2}{144} + \frac{\pi^2}{36}}$$

$$d \approx \sqrt{\frac{\pi^2}{16} + \frac{\pi^2}{36}} = \pi\sqrt{\frac{1}{16} + \frac{1}{36}} = \pi\sqrt{\frac{9+4}{144}} = \frac{\pi\sqrt{13}}{12} \approx 0.944 \text{ units}$$

Intuition

The torus is more complex because:

1. **Multiple paths**: Unlike the sphere or plane, there can be multiple geodesics (shortest paths) between two points on a torus, depending on how many times the path wraps around the hole or the tube.
2. **Mixed curvature**: The torus has positive curvature on the outer surface (like a sphere) and negative curvature on the inner surface (saddle-like). This mixed geometry makes distance calculations more involved.
3. **Local flatness**: Despite the global complexity, small patches on the torus still look like flat 2D planes, which is why our approximation works for nearby points.
4. **Practical consideration**: For real-world applications, exact geodesic calculations on a torus often require numerical methods or more advanced differential geometry.

4.3 Mathematical Background

4.3.1 Why These Formulas Work

The distance formulas we've discussed are based on the concept of a **metric**—a way of measuring distances on a manifold. Each manifold has its own metric that determines how distances are calculated.

- **Plane**: Uses the Euclidean metric $ds^2 = dx^2 + dy^2$, which gives rise to the familiar distance formula.
- **Sphere**: Uses the spherical metric, which in spherical coordinates is $ds^2 = R^2(d\phi^2 + \sin^2\phi\, d\lambda^2)$, leading to the great circle distance formula.

- **Torus**: Uses a more complex metric that depends on both the major and minor radii, making the distance calculations more involved.

4.3.2 Geodesics

The shortest paths on manifolds are called **geodesics**:

- On a plane: Geodesics are straight lines.
- On a sphere: Geodesics are great circle arcs.
- On a torus: Geodesics can be more complex, potentially wrapping around the torus multiple times.

Key Takeaways 2

- The 2D plane uses the simple Euclidean distance formula: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.
- On a sphere, distances are calculated using great circle arcs: $d = R \cdot \arccos(\vec{v}_1 \cdot \vec{v}_2)$.
- On a torus, distance calculations are more complex and typically require approximations for nearby points: $d \approx \sqrt{(R + r\cos\theta)^2(\Delta\phi)^2 + r^2(\Delta\theta)^2}$.
- Each manifold has different curvature properties that affect how distances are measured, but all maintain local flatness—small patches look like flat planes.

Chapter 5

Why Not a Cone?

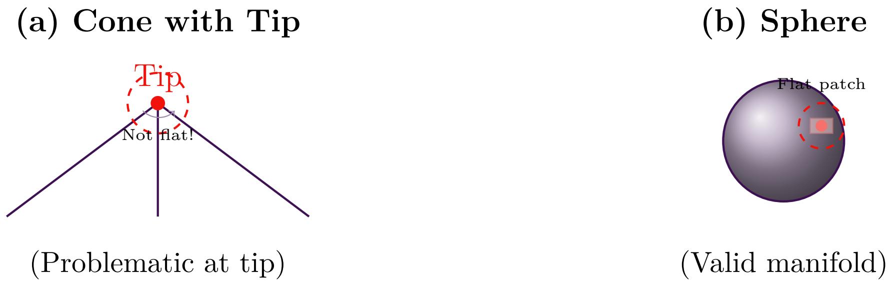
A sharp tip of a cone is not a manifold because around the tip, the neighborhood doesn't look flat; it looks like a sharp point instead of a smooth surface. This chapter explores what makes a space fail to be a manifold, helping to clarify the definition through counterexamples.

5.1 The Problem: A Cone's Tip

We've seen that manifolds are spaces that locally look like flat Euclidean space. A sphere works because if you zoom in on any point, it looks like a flat plane. A plane works because it's already flat. But what about a cone?

A cone seems smooth everywhere—except at its tip. Let's examine why the tip causes problems.

5.2 Visual Comparison


(a) Cone with Tip

(b) Sphere

(Problematic at tip)

(Valid manifold)

Figure 5.1: Comparison: (a) A cone's tip cannot be flattened into a disk, while (b) any point on a sphere can be approximated by a flat patch.

5.3 Why the Tip Fails

5.3.1 The Local Flatness Requirement

For a space to be a manifold, every point must have a neighborhood that looks like Euclidean space. Specifically, we need to be able to find:

- An open set U containing the point
- A homeomorphism (continuous bijection with continuous inverse) mapping U to an open disk in $\mathbb{R}^2$

At the tip of a cone, this fails. Let's see why.

5.3.2 The Angle Deficit Problem

Imagine trying to flatten the tip of a cone. If you cut along a line from the tip to the base and try to flatten it, you'll notice something: the angles around the tip don't add up to 360° (or 2π radians) like they would in a flat plane.

For a cone with opening angle α, if you cut and flatten it, you'll get a sector of a circle with angle less than 2π. The "missing" angle is called the **angle deficit**.

This means that no matter how small a neighborhood you take around the tip, you cannot smoothly map it to a flat disk without distortion. The tip is a **singularity**—a point where the manifold structure breaks down.

5.3.3 Visualizing the Problem

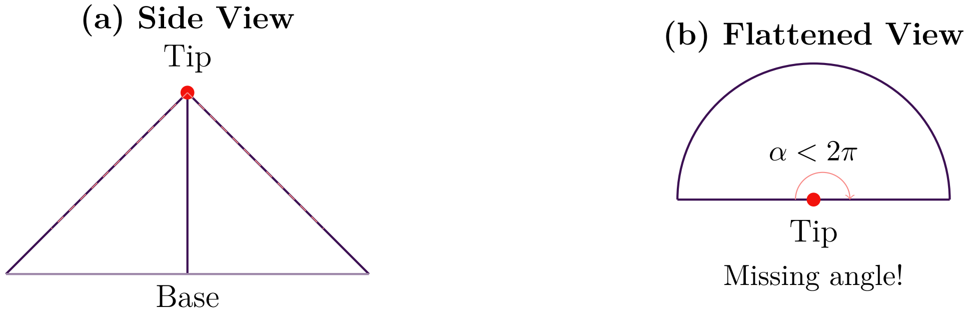


Figure 5.2: When you cut and flatten a cone, the tip maps to a sector with angle less than 2π, showing the angle deficit. This prevents the tip from having a neighborhood homeomorphic to a disk.

> **Remark 5.1: What is homeomorphic?**
>
> Two spaces are *homeomorphic* if there is a continuous bijection between them with a continuous inverse. Intuitively, one can be deformed into the other without cutting or gluing. In manifold language, "locally homeomorphic to Euclidean space" means each point has a neighborhood topologically equivalent to an open ball.

5.4 The Rest of the Cone is Fine

Here's an important point: **the cone minus its tip IS a manifold!**

If you remove the tip, every remaining point has a perfectly good neighborhood that looks like a flat disk. You can smoothly "unroll" any small patch of the cone (except near the tip) into a flat surface.

This illustrates an important principle: a space can fail to be a manifold at just a single point, while being perfectly fine everywhere else.

5.5 Mathematical Explanation

5.5.1 Attempting a Homeomorphism

Let's try to construct a homeomorphism from a neighborhood of the tip to an open disk in $\mathbb{R}^2$.

Suppose we have a cone with tip at point P. Consider any open neighborhood U of P. If we try to map U to an open disk D in $\mathbb{R}^2$, we run into problems:

1. **Angle preservation**: In a flat disk, the angles around any point sum to 2π. But at the cone's tip, the angles sum to less than 2π (or more, depending on the cone's geometry).

2. **Continuity issues**: Any continuous mapping that tries to "flatten" the tip will necessarily distort distances or angles in a way that prevents it from being a homeomorphism.

3. **Local structure**: The tip has a fundamentally different local structure than a point in $\mathbb{R}^2$. It's like trying to smoothly map a corner to a smooth curve—it's impossible without breaking the smooth structure.

5.5.2 The Curvature Singularity

Remark 5.2: Singularity and singular point

In differential geometry, a *singularity* is a point where the smooth structure or geometric quantities fail to be well-defined (e.g., not differentiable) or become unbounded. A *singular point* of a surface is a point at which no coordinate chart yields a smooth embedding with a regular (full-rank) differential, so standard objects like a well-defined tangent plane or curvature tensors cannot be assigned in the usual way. For a cone, the tip is singular because any local parametrization has rank deficiency at the apex, and curvature measures are not defined there.

Remark 5.3: Curvature (brief)

Curvature quantifies deviation from flatness. On surfaces in $\mathbb{R}^3$, *Gaussian curvature* K at a regular point is the product of principal curvatures and can be computed from the first and second fundamental forms; for smooth Riemannian manifolds, curvature is encoded by the Riemann curvature tensor R, with sectional curvature giving a 2D-plane slice of R. Smooth points have finite, well-defined curvature; at the cone tip, curvature is not defined in the smooth sense (it concentrates as a singular defect).

The tip of a cone represents a **curvature singularity**. In differential geometry, we can measure how much a surface curves at each point. For a smooth surface like a sphere, the curvature is well-defined and continuous everywhere. But at the cone's tip, the curvature becomes infinite or undefined—it's a singularity.

This is why we require manifolds to be locally homeomorphic to Euclidean space: we want to avoid these problematic singularities where the geometry breaks down.

5.6 Comparison with Valid Manifolds

Let's compare the cone's tip with points on manifolds we know work:

5.6.1 Sphere

On a sphere, pick any point. No matter how you zoom in, you can always find a small patch that looks like a flat disk. The sphere's curvature is smooth and continuous, so there's no singularity.

5.6.2 Plane

On a plane, every point is already in a flat neighborhood. The curvature is zero everywhere, so there's no problem at all.

5.6.3 Cylinder

A cylinder (without its edges if it's a finite cylinder) is also a manifold. Even though it's curved, you can "unroll" any small patch into a flat rectangle. The key is that the curvature is smooth and doesn't have singularities.

5.6.4 Cone Tip

The cone's tip is different: it has a singularity. No matter how small a neighborhood you take, you cannot smoothly flatten it because of the angle deficit.

5.7 Intuitive Understanding

5.7.1 The Paper Analogy

Think of trying to flatten a piece of paper with a sharp crease. If you try to flatten the crease, you'll either:

- Have to tear the paper (breaking continuity)
- Leave a gap or overlap (not a valid mapping)
- Accept that it's not truly flat (violating the manifold requirement)

The cone's tip is like a permanent crease that cannot be smoothed out.

5.7.2 Zooming In

One way to test if something is a manifold is to "zoom in" infinitely.

- On a sphere: As you zoom in, the surface becomes flatter and flatter, eventually looking like a plane.
- On a cone (away from tip): Same thing—it looks flatter as you zoom in.

- On a cone (at the tip): No matter how much you zoom in, you still see a sharp point with the same angle deficit. It never looks flat!

This is the essence of why the tip fails: it doesn't have the local flatness property that defines a manifold.

5.8 Other Examples of Non-Manifolds

Example 5.1: Other non-manifold cases

Understanding why a cone fails helps us identify other non-manifolds:

- **Cusps**: Sharp points where two curves meet, similar to the cone tip.
- **Corners on polyhedra**: The vertices of a cube are not manifolds (though the edges and faces are).
- **Self-intersections**: A figure-8 curve intersecting itself creates a point that's not a manifold.
- **Boundary points**: Points on the edge of a disk (if we consider the disk as a 2D object) are not manifolds in the strict sense, though they form a "manifold with boundary."

Key Takeaways 3

- The tip of a cone is **not** a manifold because it cannot be locally mapped to a flat disk without distortion.
- The problem is the **angle deficit** at the tip—the angles don't sum to 2π like they would in Euclidean space.
- The cone **minus its tip** IS a valid manifold, showing that a single problematic point can disqualify an entire space.
- Manifolds must be locally flat **everywhere**—having even one point that violates this condition means the space is not a manifold.
- Understanding counterexamples like the cone tip helps clarify the manifold definition by showing what it means to fail the local flatness requirement.

5.9 Conclusion

The cone tip serves as an excellent counterexample because it's visually intuitive—we can see why it's problematic—while also having a clear mathematical reason for its failure. By understanding why the cone tip is not a manifold, we gain deeper insight into what makes a space a manifold: it must be locally like Euclidean space at **every single point**, without any singularities or special points where this property breaks down.

Chapter 6

Geodesics: Introduction

A **geodesic** is the **shortest path between two points on a curved surface or manifold**. It's a generalization of "straight lines" to curved spaces. This chapter introduces the concept and prepares the ground for exploring geodesics in various contexts.

6.1 What is a Geodesic?

Definition 6.1: Geodesic

On a Riemannian manifold $(\mathcal{M}, g)$, a smooth curve $\gamma : [t_1, t_2] \to \mathcal{M}$ is a *geodesic* if its velocity is parallel transported along itself,

$$\nabla_{\dot{\gamma}}\dot{\gamma} \;=\; 0,$$

equivalently, γ is a stationary curve of the length (or energy) functional with fixed endpoints. In local coordinates $x^i(t)$ this is

$$\ddot{x}^i + \Gamma^i_{jk}\,\dot{x}^j\dot{x}^k = 0.$$

A geodesic is the shortest path between two points that stays entirely on the surface or manifold. Think of it as the natural generalization of a straight line to curved spaces.

- On a flat plane: A geodesic is simply a straight line.
- On a sphere: A geodesic is an arc of a great circle (the largest circle on the sphere).
- On a cylinder: A geodesic can be a straight line (when unrolled) or a helix.
- On any curved surface: A geodesic is the path a string would take if you pulled it tight between two points on the surface.

The key insight is that geodesics are determined by the geometry of the surface itself—they're the "straightest possible" paths while staying on the surface.

6.2 Why Straight Lines Don't Work on Curved Surfaces

In flat Euclidean space, the shortest path between two points is a straight line. But on a curved surface, a straight line in the ambient space might cut through the surface, which violates our requirement that paths must stay on the surface.

6.2.1 The Problem

Consider trying to find the shortest path between two cities on Earth. If we draw a straight line through the Earth (cutting through the planet), that's not useful—we need to travel along the Earth's surface!

(a) Straight Line
Straight line
P_2
P_1
(Cuts through sphere)
(b) Geodesic
Great circle arc
P_2
P_1
(Stays on surface)

Figure 6.1: Comparison: (a) A straight line through space cuts through the sphere, while (b) a geodesic (great circle arc) stays on the surface and represents the shortest path along the surface.

6.2.2 The Solution: Geodesics

Geodesics solve this problem by finding the shortest path that **stays on the surface**. They're the natural generalization of straight lines to curved spaces.

6.3 Visual Examples of Geodesics

Example 6.1: Visual examples of geodesics

This example previews geodesics on several manifolds; see the panels below.

6.4 Key Properties of Geodesics

Geodesics have several important properties:

6.4.1 Shortest Path Property

The fundamental property of a geodesic is that it minimizes the distance between two points, measured along the surface. This is why we use geodesics to calculate distances on manifolds.

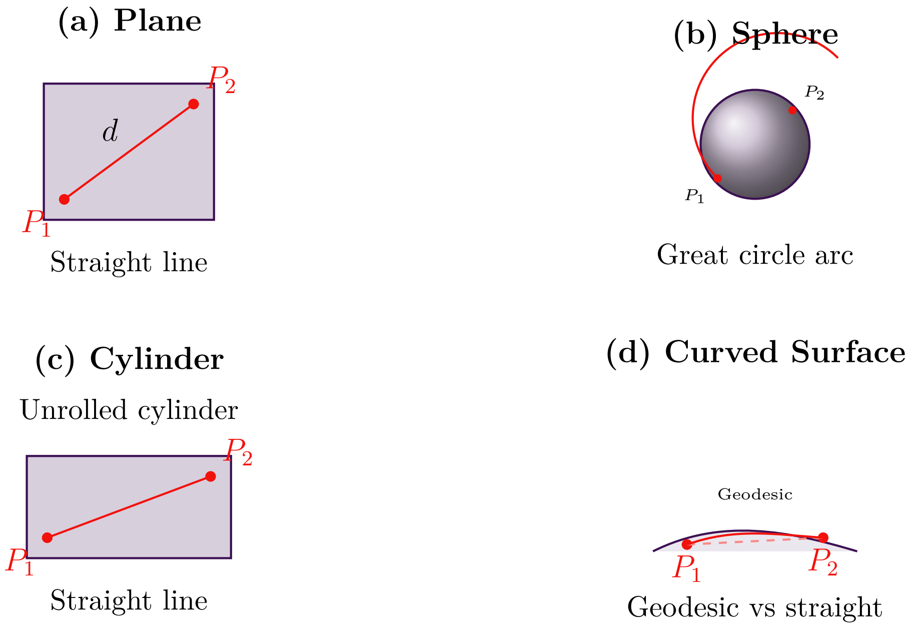


Figure 6.2: Examples of geodesics on different manifolds: (a) On a plane, geodesics are straight lines. (b) On a sphere, geodesics are great circle arcs. (c) On a cylinder, geodesics appear as straight lines when unrolled. (d) On a curved surface, the geodesic follows the surface while a straight line cuts through space.

6.4.2 Local Straightness

Geodesics are "locally straight"—if you zoom in on any small segment of a geodesic, it looks like a straight line. This connects to the idea that manifolds are locally flat.

Example 6.2: Local straightness on S^1

On a circle of radius R, two nearby angles θ_1, θ_2 with $\Delta\theta = |\theta_2 - \theta_1| \ll 1$ have geodesic (arc) distance

$$d_{\text{arc}} = R\,\Delta\theta,$$

while the chord distance is $d_{\text{chord}} = 2R\sin(\Delta\theta/2) = R\,\Delta\theta\Big(1 - \frac{(\Delta\theta)^2}{24} + \cdots\Big)$. Thus, over small neighborhoods the geodesic looks straight to first order (arc and chord agree to $O(\Delta\theta)$).

6.4.3 Uniqueness (or Not)

Depending on the manifold:

- On a plane: There's exactly one geodesic (straight line) between any two points.
- On a sphere: There's exactly one shortest geodesic (the shorter great circle arc), but potentially two paths if we consider the longer arc.
- On a torus: There can be multiple geodesics between two points, wrapping around the hole or tube in different ways.

6.4.4 Symmetry

Geodesics are symmetric: the geodesic from point A to point B is the same as the geodesic from B to A (just traversed in the opposite direction).

Remark 6.1: When are geodesics unique?

On a complete Riemannian manifold, for points within a *convex normal neighborhood* (i.e., before reaching the cut locus), there exists a *unique* minimizing geodesic between them. Non-uniqueness arises beyond the injectivity radius (e.g., antipodal points on a sphere admit infinitely many geodesics).

6.5 Connection to Distance Calculations

In previous chapters, we calculated distances on manifolds. Those distances are precisely the lengths of geodesics!

6.5.1 Relationship to Previous Chapters

- **1D Plane**: The distance $d = |x_2 - x_1|$ is the length of the straight line (geodesic) segment.
- **1D Circle**: The distance $d = r \cdot \theta$ is the length of the geodesic arc along the circle.
- **2D Plane**: The distance $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ is the length of the straight line (geodesic).
- **2D Sphere**: The great circle distance $d = R \cdot \arccos(\vec{v}_1 \cdot \vec{v}_2)$ is the length of the geodesic arc.

This connection is fundamental: **the distance between two points on a manifold equals the length of the geodesic connecting them**.

6.6 Mathematical Intuition

6.6.1 Minimizing Path Length

Geodesics can be thought of as paths that minimize the total length. Mathematically, if we have a path $\gamma(t)$ between two points, the length is:

Definition 6.2: Length functional

Given a smooth curve $\gamma : [t_1, t_2] \to \mathcal{M}$ with local coordinates $x^i(t)$, its length is

$$L[\gamma] = \int_{t_1}^{t_2} \sqrt{g_{ij}(\gamma(t))\,\dot{x}^i(t)\,\dot{x}^j(t)}\,dt.$$

where g_{ij} is the metric tensor that encodes the geometry of the manifold. A geodesic is a path that minimizes this length functional.

6.6.2 Variational Principle

Geodesics satisfy a variational principle: they're paths for which any small variation increases (or at least doesn't decrease) the length. This is analogous to how a taut string naturally takes the shortest path between two fixed points.

6.6.3 The Geodesic Equation

For those familiar with calculus of variations, geodesics satisfy the geodesic equation:

Definition 6.3: Geodesic equation (coordinate form)

In local coordinates $x^i(t)$ on $(\mathcal{M}, g)$, a geodesic satisfies

$$\frac{d^2x^i}{dt^2} + \Gamma^i_{jk}(x)\,\frac{dx^j}{dt}\,\frac{dx^k}{dt} = 0,$$

where Γ^i_{jk} are the Christoffel symbols of the Levi-Civita connection.

This equation ensures the path is "straight" in the curved space.

Don't worry if this looks complicated—the key intuition is that geodesics are paths that are "as straight as possible" given the curvature of the space.

6.7 Real-World Examples

Geodesics appear everywhere in our daily lives:

Example 6.3: Aviation and navigation

When airplanes fly long distances, they follow great circle routes—geodesics on the Earth's sphere. This is why flight paths appear curved on flat maps but are actually the shortest routes.

Example 6.4: GPS and mapping

GPS systems calculate shortest routes along roads, which approximate geodesics on the road network (a graph-like manifold). The navigation system finds geodesic-like paths within the network.

Example 6.5: Surface transportation

When planning routes on hilly terrain, the shortest path follows the geodesics of the terrain's surface. Hikers naturally follow geodesic-like paths when taking the most direct route.

Example 6.6: Physics

In general relativity, light rays and free-falling objects follow geodesics in spacetime. This is one of the most profound applications of geodesics in physics.

6.8 Relationship to Curvature

The curvature of a manifold affects how geodesics behave:

6.8.1 Zero Curvature (Flat Space)

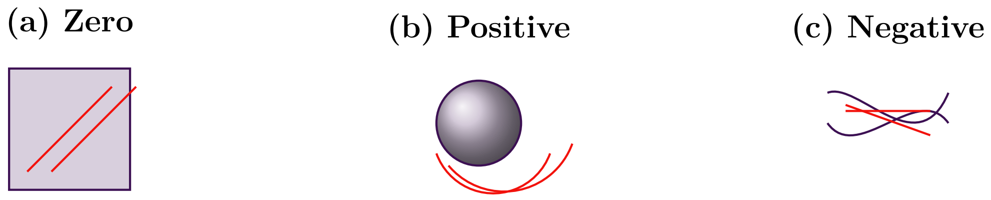
On a plane (zero curvature), geodesics are straight lines that never converge or diverge. Parallel geodesics remain parallel.

6.8.2 Positive Curvature (Sphere)

On a sphere (positive curvature), geodesics (great circles) tend to converge. Two initially parallel geodesics will eventually meet—think of lines of longitude on Earth meeting at the poles.

6.8.3 Negative Curvature (Saddle)

On a saddle surface (negative curvature), geodesics tend to diverge. Two initially parallel geodesics will spread apart.



(a) Zero

(b) Positive

(c) Negative

Parallel stay parallel

Converge

Diverge

Figure 6.3: How curvature affects geodesics: (a) Zero curvature: parallel geodesics stay parallel. (b) Positive curvature: geodesics converge. (c) Negative curvature: geodesics diverge.

6.9 Intuitive Understanding: The String Analogy

The best way to understand geodesics is to imagine pulling a string tight between two points on a surface:

- The string naturally takes the shortest path.
- It stays on the surface (can’t cut through it).
- It’s under tension, which minimizes its length.

- The path it takes is the geodesic!

This physical intuition helps us understand why geodesics are the natural "straight lines" on curved surfaces.

6.10 What's Next?

In the next chapter, we'll explore geodesics in detail:

- **1D Manifolds**: Geodesics on lines and circles.
- **2D Manifolds**: Geodesics on planes and spheres.
- **Calculations**: How to find and compute geodesics.
- **Examples**: Specific cases with numerical calculations.

We'll see how the abstract concept of geodesics plays out in concrete examples, building on the foundation we've established here.

Key Takeaways 4

- A **geodesic** is the shortest path between two points that stays on a manifold.
- Geodesics are the natural generalization of straight lines to curved spaces.
- The distance between two points on a manifold equals the length of the geodesic connecting them.
- Geodesics depend on the geometry (curvature) of the manifold—they're not just straight lines in the ambient space.
- Real-world examples include airplane routes (great circles), GPS navigation, and paths on terrain.
- Curvature affects how geodesics behave: zero curvature keeps them parallel, positive curvature makes them converge, negative curvature makes them diverge.
- The string analogy provides excellent intuition: geodesics are like the path a taut string would take on a surface.

6.11 Conclusion

Geodesics are fundamental to understanding geometry on manifolds. They provide the natural way to measure distances, define "straightness" on curved surfaces, and connect the abstract mathematical structure of manifolds to practical applications like navigation and physics.

By understanding geodesics, we gain insight into how geometry works in curved spaces—a concept that's essential not just in mathematics, but in physics (general relativity), computer graphics, robotics, and many other fields.

Chapter 7

Geodesics in 1D, 2D, and on a Sphere

In previous chapters, we calculated distances on manifolds and learned that these distances equal the lengths of geodesics. This chapter goes deeper: we'll derive geodesics systematically, solve the geodesic equations, and explore their properties. Instead of just stating what geodesics are, we'll learn *how* to find them and *why* they have their particular forms.

7.1 Introduction: From Distances to Paths

We've already established that:

- The distance between two points on a manifold equals the length of the geodesic connecting them.
- On a line, geodesics are straight segments.
- On a circle, geodesics are arcs.
- On a plane, geodesics are straight lines.
- On a sphere, geodesics are great circle arcs.

But *how* do we know these are the geodesics? And *how* can we find geodesics on more complex manifolds? This chapter answers these questions by introducing systematic methods for deriving geodesics.

7.2 Deriving Geodesics: The Variational Approach

7.2.1 The Path Length Functional

A fundamental principle is that geodesics minimize path length. Mathematically, if we have a path $\gamma(t)$ between two points P_1 and P_2 on a manifold, its length is given by (see Definition 6.6.1):

$$L[\gamma] = \int_{t_1}^{t_2} \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}} \, dt$$

where g_{ij} is the metric tensor encoding the geometry, and $x^i(t)$ are the coordinates of the path. A geodesic is a path that minimizes this functional.

7.2.2 The Euler-Lagrange Equations

To find paths that minimize $L[\gamma]$, we use the calculus of variations. The Euler-Lagrange equations tell us that if a path minimizes an integral $\int L(x, \dot{x}, t)\, dt$, then:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}^i}\right) - \frac{\partial L}{\partial x^i} = 0$$

Notes. $\dot{x}^i = \frac{dx^i}{dt}$ denotes time derivatives of the coordinates along the curve $\gamma(t)$; L is the Lagrangian (here derived from length/energy); t is a curve parameter; x^i are local coordinates on the manifold.

Remark 7.1: Euler–Lagrange refresher

For a scalar functional $\int L(x, \dot{x}, t)\, dt$ of a single coordinate $x(t)$ (and its velocity $\dot{x}$), stationary curves satisfy

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0.$$

In multiple coordinates $x^i(t)$, apply this to each component to obtain the vector form above.

For the path length functional, this leads to the geodesic equation. Let's see how this works for simple manifolds.

7.2.3 Applying Variational Calculus: 1D Line

On a 1D line, the metric is simply $g_{11} = 1$, so the path length is:

$$L = \int_{t_1}^{t_2} \left|\frac{dx}{dt}\right| dt = \int_{t_1}^{t_2} \sqrt{\left(\frac{dx}{dt}\right)^2}\, dt$$

The Lagrangian is $L = \sqrt{(\dot{x})^2} = |\dot{x}|$. For a smooth path, we can assume $\dot{x} > 0$ (or reverse the parameterization), so $L = \dot{x}$.

The Euler-Lagrange equation gives:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = \frac{d}{dt}(1) - 0 = 0$$

This is automatically satisfied! But we need to consider the full functional. Actually, for the path length, we should use $L = (\dot{x})^2$ (minimizing squared length is equivalent), which gives:

$$\frac{d}{dt}(2\dot{x}) = 0 \quad \Rightarrow \quad \ddot{x} = 0$$

This means $\dot{x}$ is constant, so $x(t) = at+b$—a straight line! This confirms that geodesics on a line are straight segments.

7.2.4 Applying Variational Calculus: 2D Plane

On a 2D plane with Cartesian coordinates, the metric is $g_{ij} = \delta_{ij}$ (the identity matrix), so:

$$L = \int \sqrt{(\dot{x})^2 + (\dot{y})^2}\, dt$$

Using the squared length approach (minimizing $(\dot{x})^2 + (\dot{y})^2$), the Euler-Lagrange equations give:

$$\ddot{x} = 0, \quad \ddot{y} = 0$$

This means both $x(t)$ and $y(t)$ are linear functions, so the path is a straight line: $\mathbf{r}(t) = \mathbf{p}_1 + t(\mathbf{p}_2 - \mathbf{p}_1)$.

7.3 Solving the Geodesic Equation

7.3.1 The Geodesic Equation in Detail

The geodesic equation is:

$$\frac{d^2x^i}{dt^2} + \Gamma^i_{jk}\frac{dx^j}{dt}\frac{dx^k}{dt} = 0$$

where Γ^i_{jk} are the **Christoffel symbols**, which encode how the metric changes with position. They are given by:

$$\Gamma^i_{jk} = \frac{1}{2}g^{il}\left(\frac{\partial g_{jl}}{\partial x^k} + \frac{\partial g_{kl}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^l}\right)$$

where g^{il} is the inverse of the metric tensor.

7.3.2 Solving for a 1D Line

On a line, the metric is constant: $g_{11} = 1$. Therefore, all partial derivatives vanish, so $\Gamma^1_{11} = 0$.

The geodesic equation becomes:

$$\frac{d^2x}{dt^2} = 0$$

Solving this: $\dot{x} =$ constant, so $x(t) = x_0 + vt$. This is a straight line with constant velocity, confirming our earlier result.

7.3.3 Solving for a 2D Plane

In Cartesian coordinates, the metric is $g_{ij} = \delta_{ij}$, which is constant. Therefore, all Christoffel symbols vanish: $\Gamma^i_{jk} = 0$.

The geodesic equations become:

$$\frac{d^2x}{dt^2} = 0, \quad \frac{d^2y}{dt^2} = 0$$

Solutions: $x(t) = x_0 + v_x t$, $y(t) = y_0 + v_y t$. This describes a straight line in the plane.

7.3.4 Solving for a 1D Circle

A circle of radius r can be parameterized by angle θ. The metric is $g_{\theta\theta} = r^2$ (constant), so $\Gamma^{\theta}_{\theta\theta} = 0$.

The geodesic equation becomes:

$$\frac{d^2\theta}{dt^2} = 0$$

Solution: $\theta(t) = \theta_0 + \omega t$, where ω is constant. This describes uniform motion along the circle—a geodesic arc.

7.3.5 Solving for a 2D Sphere

On a sphere of radius R, using spherical coordinates (ϕ, λ) (latitude, longitude), the metric is:

$$ds^2 = R^2(d\phi^2 + \sin^2\phi\, d\lambda^2)$$

The Christoffel symbols are non-zero. The geodesic equations become:

$$\frac{d^2\phi}{dt^2} - \sin\phi\cos\phi\left(\frac{d\lambda}{dt}\right)^2 = 0 \tag{7.1}$$

$$\frac{d^2\lambda}{dt^2} + 2\cot\phi\frac{d\phi}{dt}\frac{d\lambda}{dt} = 0 \tag{7.2}$$

These equations are more complex, but their solutions describe great circles. One can verify that paths with constant λ (lines of longitude) and the equator ($\phi = 0$) are solutions, representing great circles.

A more elegant approach uses the fact that great circles are intersections of the sphere with planes through the center, which we'll explore in the next section.

7.4 Geometric Construction of Geodesics

7.4.1 Construction Methods

Sometimes it's easier to construct geodesics geometrically rather than solving differential equations. Let's see how this works for each manifold.

1D Line

Trivial: the geodesic is simply the straight line segment connecting the two points.

1D Circle

Given two points on a circle, there are two arcs connecting them. The geodesic is the shorter arc. If the points are antipodal, both arcs are geodesics of equal length.

Remark 7.2: Antipodal points

Two points on a circle or sphere are *antipodal* if they are diametrically opposite (separated by half the circumference on the circle or by a central angle of π on the sphere). On a sphere, every great circle through one antipodal point also passes through the other.

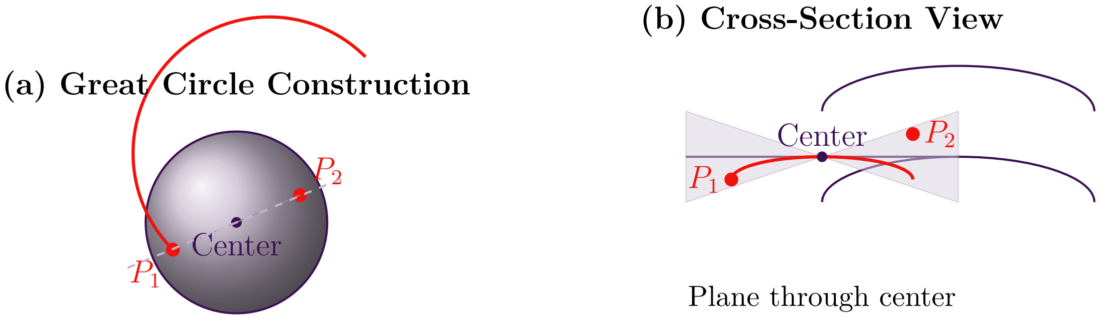
2D Plane

The geodesic is the straight line segment. This can be constructed using a ruler or by drawing the line through the two points.

2D Sphere: Great Circle Construction

For a sphere, geodesics are great circles. Here's how to construct the great circle through two points P_1 and P_2:

1. Consider the sphere centered at the origin.
2. The two points define vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ from the center.
3. The great circle lies in the plane containing the origin, P_1, and P_2.
4. This plane is perpendicular to the normal vector $\mathbf{n} = \mathbf{p}_1 \times \mathbf{p}_2$.
5. The intersection of this plane with the sphere is the great circle.
6. The geodesic is the shorter arc of this great circle connecting P_1 and P_2.



(a) Great Circle Construction

(b) Cross-Section View

Figure 7.1: Constructing a great circle geodesic on a sphere: (a) The great circle is the intersection of the sphere with a plane through the center and both points. (b) Cross-section showing how the plane cuts the sphere.

This geometric construction is often more intuitive than solving differential equations and directly shows why great circles are geodesics.

7.4.2 Coordinate-Free Description

Geodesics are **intrinsic** to the manifold—they depend only on the metric, not on how the manifold is embedded in higher-dimensional space. For example, great circles on a sphere are geodesics regardless of how we view the sphere in 3D space. This is a fundamental property of Riemannian geometry.

7.5 Multiple Geodesics and Special Cases

7.5.1 When Are There Multiple Geodesics?

Not all pairs of points have a unique geodesic. Let's examine when multiple geodesics exist:

- **1D Line**: Always exactly one geodesic (the straight line segment).
- **1D Circle**:
  - Usually one geodesic (the shorter arc).
  - If points are antipodal: two geodesics of equal length (both semicircles).
- **2D Plane**: Always exactly one geodesic (the straight line).
- **2D Sphere**:
  - Usually one shortest geodesic (the shorter great circle arc).
  - If points are antipodal: infinitely many geodesics (all great circles through them have the same length: πR).
  - The longer great circle arc is also a geodesic, but not the shortest one.

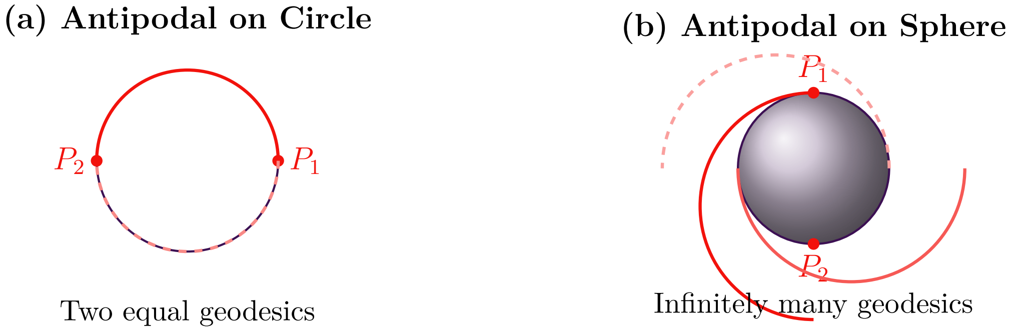


Figure 7.2: Multiple geodesics: (a) On a circle, antipodal points have two geodesics of equal length. (b) On a sphere, antipodal points have infinitely many geodesics (all great circles through them).

7.5.2 Conjugate Points and Cut Locus

The set of points where geodesics cease to be unique is called the **cut locus**.

Example 7.1: Cut locus examples

For example:

- On a circle, the cut locus of a point is its antipodal point.
- On a sphere, the cut locus of a point is its antipodal point.

At these special points, the geometry becomes degenerate in the sense that multiple geodesics have the same length.

7.6 Geodesics in Different Coordinate Systems

7.6.1 Coordinate Transformations

Geodesics are geometric objects—they exist independently of coordinate systems. However, their descriptions change with coordinates. Let's see how the same geodesic looks in different coordinate systems.

Example 7.2: Straight line in different coordinates

Consider a straight line in the plane. In Cartesian coordinates (x, y), it's simply:

$$y = mx + b$$

In polar coordinates (r, θ), the same line becomes:

$$r = \frac{b}{\sin\theta - m\cos\theta}$$

This is more complex, but it's the same geometric object.

Example 7.3: Great circle on a sphere

A great circle on a sphere can be described in:

- **Cartesian coordinates**: $ax + by + cz = 0$ (plane through origin)
- **Spherical coordinates**: More complex, but the geodesic equation we solved earlier describes it.

The key insight: the geodesic equation transforms covariantly, so solving it in any coordinate system gives the same geometric geodesic.

7.6.2 Natural Parameterization

A particularly useful parameterization is the **arc length parameter** s. In this parameterization, the speed along the geodesic is constant (equal to 1), so:

$$\left|\frac{d\mathbf{r}}{ds}\right| = 1$$

For example:

- On a line: $x(s) = x_0 + s$ (if we choose appropriate orientation).
- On a circle: $\theta(s) = \theta_0 + s/r$ (uniform angular speed).
- On a plane: $\mathbf{r}(s) = \mathbf{r}_0 + s\mathbf{u}$ where $|\mathbf{u}| = 1$.

This parameterization simplifies many calculations and is often used in differential geometry.

7.7 Properties and Characteristics of Geodesics

7.7.1 Local vs Global Properties

Local Minimality

Geodesics are **locally shortest**: if you take a small enough neighborhood around any point on a geodesic, it's the shortest path between its endpoints in that neighborhood.

Global Minimality

On simply connected manifolds (like a plane or sphere), geodesics are also **globally shortest**—they minimize distance among all possible paths. However, there can be exceptions:

- On a sphere, the longer great circle arc is a geodesic but not the shortest path.
- On a torus, there can be multiple geodesics between two points, and the shortest one depends on how the geodesic wraps around.

7.7.2 Geodesic Completeness

A manifold is **geodesically complete** if every geodesic can be extended indefinitely (in both directions).

Example 7.4: Geodesic completeness examples

- **Line**: Complete—geodesics extend to $\pm\infty$.
- **Circle**: Complete—geodesics are closed (they loop around).
- **Plane**: Complete—geodesics extend infinitely.
- **Sphere**: Complete—geodesics are closed (great circles).

Geodesic completeness is an important property that relates to the global structure of the manifold.

7.7.3 Geodesic Curvature

Geodesics have **zero geodesic curvature**—they are "as straight as possible" on the manifold. This is an intrinsic property. However, they may appear curved when viewed from an embedding space:

- A great circle on a sphere has zero geodesic curvature (intrinsic).
- But it appears curved when viewed in 3D space (extrinsic curvature).

This distinction between intrinsic and extrinsic geometry is fundamental to understanding manifolds.

7.8 Worked Examples: Systematic Derivation

Example 7.5: Circle geodesic via variational method

Let's derive that arcs are geodesics on a circle using the variational approach.
Setup: Circle of radius r, parameterized by angle θ. The metric is $g_{\theta\theta} = r^2$.
Path length functional:

$$L = \int \sqrt{r^2 \left(\frac{d\theta}{dt}\right)^2}\, dt = r \int \left|\frac{d\theta}{dt}\right| dt$$

For smooth paths, we can use $L = r \int (\dot{\theta})^2 \, dt$ (minimizing squared length).
Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \frac{d}{dt}(2r^2\dot{\theta}) = 0$$

This gives $\ddot{\theta} = 0$, so $\theta(t) = \theta_0 + \omega t$—uniform motion along the circle, which describes an arc. This confirms that arcs are geodesics on a circle.

Example 7.6: Sphere geodesic via geometric construction

Let's construct the great circle through two points on a sphere.
Setup: Sphere of radius R, points P_1 and P_2 with position vectors $\mathbf{p}_1$ and $\mathbf{p}_2$.
Construction:

1. The normal to the plane containing the great circle is $\mathbf{n} = \mathbf{p}_1 \times \mathbf{p}_2$.
2. The plane equation is $\mathbf{n} \cdot \mathbf{r} = 0$ (passes through origin).
3. The intersection of this plane with the sphere $|\mathbf{r}| = R$ is the great circle.
4. The geodesic is the shorter arc of this great circle.

This geometric construction is often simpler than solving differential equations and directly shows why great circles are geodesics.

Example 7.7: Coordinate transformation

Let's verify that a straight line geodesic on a plane transforms correctly under coordinate changes.
In Cartesian coordinates: Geodesic is $y = 2x + 1$ (straight line).
In polar coordinates: $x = r\cos\theta$, $y = r\sin\theta$. Substituting:

$$r\sin\theta = 2r\cos\theta + 1$$

Solving for r:

$$r = \frac{1}{\sin\theta - 2\cos\theta}$$

This is the same geodesic, just described in different coordinates. The geodesic equation, when transformed to polar coordinates, will give this same result.

7.9 Comparison Across Manifolds: Deeper Insights

7.9.1 Systematic Comparison

Manifold	Curvature	Geodesic Type	Uniqueness	Completeness
1D Line	Zero	Straight line	Always unique	Complete
1D Circle	Positive	Arc	Unique (except antipodal)	Complete
2D Plane	Zero	Straight line	Always unique	Complete
2D Sphere	Positive	Great circle arc	Unique shortest	Complete

Table 7.1: Comparison of geodesic properties across different manifolds.

7.9.2 Unifying Principles

Despite their differences, all geodesics share fundamental properties:

1. **Minimization**: They minimize path length (or are critical points of the length functional).
2. **Differential equation**: They satisfy the geodesic equation $\ddot{x}^i + \Gamma^i_{jk}\dot{x}^j\dot{x}^k = 0$.
3. **Local straightness**: They have zero geodesic curvature—they're "as straight as possible" on the manifold.
4. **Coordinate independence**: They are geometric objects, independent of coordinate choices.
5. **Intrinsic nature**: They depend only on the metric (intrinsic geometry), not on embeddings.

These unifying principles allow us to study geodesics on any manifold, not just the simple ones we've explored here.

Key Takeaways 5

- **Systematic derivation**: Geodesics can be derived using variational calculus (minimizing path length) or by solving the geodesic differential equation.
- **Variational approach**: The Euler-Lagrange equations applied to the path length functional yield the geodesic equation.
- **Geodesic equation**: $\ddot{x}^i + \Gamma^i_{jk}\dot{x}^j\dot{x}^k = 0$ describes geodesics, where Christoffel symbols Γ^i_{jk} encode the geometry.
- **Geometric construction**: On spheres, great circles can be constructed as intersections with planes through the center—often simpler than solving equations.
- **Multiple geodesics**: Some manifolds (circle, sphere) have multiple geodesics between certain point pairs (antipodal points).
- **Coordinate independence**: Geodesics are geometric objects—their descriptions change with coordinates, but the geodesics themselves are invariant.
- **Unifying principles**: All geodesics minimize length, satisfy the geodesic equation, have zero geodesic curvature, and are intrinsic to the manifold.

7.10 What's Next?

The methods we've developed here extend to more complex manifolds:

- **Torus**: Geodesics can wrap around in multiple ways, leading to rich behavior.
- **General surfaces**: Numerical methods are often needed to find geodesics.
- **Riemannian geometry**: The general theory provides tools for any manifold with a metric.
- **Applications**: Geodesics appear in general relativity (spacetime paths), robotics (path planning), and computer graphics (surface rendering).

Understanding geodesics on simple manifolds provides the foundation for these advanced topics.

Chapter 8

Distance on Sphere

In Chapter 3, we learned the basic formula for calculating distances on a sphere using great circle arcs. In Chapter 6, we explored how geodesics (great circles) are derived and constructed. This chapter goes deeper into distance calculations: we'll explore alternative formulas, handle special cases, examine numerical considerations, and discuss Earth-specific applications.

8.1 Introduction: Beyond the Basic Formula

We've already established that:

- The distance between two points on a sphere equals the length of the great circle arc connecting them.
- The basic formula is $d = R \cdot \arccos(\vec{v}_1 \cdot \vec{v}_2)$ or, in spherical coordinates:

$$d = R \cdot \arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\lambda_2 - \lambda_1))$$

- We've seen a real-world example (Sydney to New York).

But there's more to learn: alternative formulas with better numerical properties, special cases that need careful handling, approximations for different scenarios, and practical considerations for Earth applications. This chapter covers these advanced topics.

8.2 Alternative Formulas for Sphere Distance

8.2.1 The Haversine Formula

The haversine formula is an alternative formulation that's often preferred for computational purposes due to better numerical stability.

The Haversine Function

Definition 8.1: Haversine function

The *haversine* of an angle θ is

$$\operatorname{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos\theta}{2}.$$

It is widely used in spherical distance formulas due to its numerical stability, especially for small angles.

Derivation of the Haversine Formula

Starting from the basic formula, we can derive the haversine version. For two points on a sphere with coordinates (ϕ_1, λ_1) and (ϕ_2, λ_2), the central angle θ between them satisfies:

$$\cos\theta = \sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\Delta\lambda)$$

where $\Delta\lambda = \lambda_2 - \lambda_1$.
Using the identity $\cos\theta = 1 - 2\sin^2(\theta/2)$, we can rewrite this in terms of haversines:

$$\operatorname{hav}(\theta) = \operatorname{hav}(\phi_2 - \phi_1) + \cos\phi_1 \cos\phi_2 \operatorname{hav}(\Delta\lambda)$$

Therefore:

$$\theta = 2 \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

And the distance is:

$$d = 2R \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

where $\Delta\phi = \phi_2 - \phi_1$.

Advantages of the Haversine Formula

The haversine formula offers several advantages:

- **Numerical stability**: Uses arcsin instead of arccos, avoiding precision issues when the argument is near -1 or 1.
- **Better for small distances**: The formula is more stable for nearby points, which is common in many applications.
- **Computational efficiency**: Avoids the need to compute arccos of values near boundaries, which can be problematic in floating-point arithmetic.
- **Widely used**: The haversine formula is the standard in many navigation and mapping applications.

8.2.2 Spherical Law of Cosines

The spherical law of cosines provides another approach to distance calculation, derived from spherical trigonometry.

Statement of the Law

For a spherical triangle with sides a, b, c (measured as angles) and opposite angles A, B, C:

$$\cos c = \cos a \cos b + \sin a \sin b \cos C$$

Application to Distance Calculation

For two points on a sphere, we can form a spherical triangle with:

- Side $a = \frac{\pi}{2} - \phi_1$ (co-latitude of first point)
- Side $b = \frac{\pi}{2} - \phi_2$ (co-latitude of second point)
- Angle $C = \Delta\lambda$ (longitude difference)
- Side $c = \frac{d}{R}$ (central angle, which is what we want to find)

Applying the spherical law of cosines:

$$\cos\left(\frac{d}{R}\right) = \cos\left(\frac{\pi}{2} - \phi_1\right)\cos\left(\frac{\pi}{2} - \phi_2\right) + \sin\left(\frac{\pi}{2} - \phi_1\right)\sin\left(\frac{\pi}{2} - \phi_2\right)\cos(\Delta\lambda)$$

Using trigonometric identities ($\cos(\pi/2 - x) = \sin x$ and $\sin(\pi/2 - x) = \cos x$):

$$\cos\left(\frac{d}{R}\right) = \sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\Delta\lambda)$$

This is exactly our basic formula! The spherical law of cosines provides the theoretical foundation for the distance formula.

8.2.3 Comparison of Formulas

Formula	**Stability**	**Best For**	**Complexity**
Basic (arccos)	Medium	General use	Simple
Haversine	High	Small distances, computation	Moderate
Spherical Law	Low	Theoretical derivation	Simple

Table 8.1: Comparison of different distance formulas on a sphere.

In practice:

- Use the **basic formula** for general understanding and when precision is not critical.

- Use the **haversine formula** for computational applications, especially with small distances or when numerical stability matters.
- Use the **spherical law of cosines** primarily for theoretical work and understanding the geometric foundation.

Remark 8.1: Distance formulas summary

Basic (arccos): $d = R\,\arccos\Big(\sin\phi_1\sin\phi_2 + \cos\phi_1\cos\phi_2\cos\Delta\lambda\Big)$.
Haversine: $d = 2R\,\arcsin\left(\sqrt{\sin^2(\frac{\Delta\phi}{2}) + \cos\phi_1\cos\phi_2\,\sin^2(\frac{\Delta\lambda}{2})}\right)$.
Spherical law of cosines (central angle): $\cos\frac{d}{R} = \sin\phi_1\sin\phi_2 + \cos\phi_1\cos\phi_2\cos\Delta\lambda$.

8.3 Special Cases and Edge Cases

8.3.1 Points on the Same Meridian

Example 8.1: Same meridian

When two points lie on the same meridian (same longitude), the formula simplifies significantly.
Condition: $\lambda_1 = \lambda_2$, so $\Delta\lambda = 0$.
The distance formula becomes:

$$d = R\cdot\arccos(\sin\phi_1\sin\phi_2 + \cos\phi_1\cos\phi_2\cdot 1)$$

Using the angle subtraction formula:

$$d = R\cdot\arccos(\cos(\phi_2 - \phi_1)) = R\cdot|\phi_2 - \phi_1|$$

Result: $d = R\cdot|\phi_2 - \phi_1|$ (latitude difference in radians).

Example 8.2: Same meridian: numerical example

If $\phi_1 = 30°$ and $\phi_2 = 60°$ on the same meridian, with $R = 6371$ km:

$$d = 6371 \times \left|\frac{\pi}{6} - \frac{\pi}{3}\right| = 6371 \times \frac{\pi}{6} \approx 3336 \text{ km}$$

8.3.2 Points on the Same Parallel (Latitude)

Example 8.3: Same parallel (latitude)

When two points have the same latitude, the distance depends on the longitude difference, but the effective radius is reduced.
Condition: $\phi_1 = \phi_2 = \phi$, so $\Delta\phi = 0$.
The distance formula simplifies to:

$$d = R \cdot \arccos(\sin^2\phi + \cos^2\phi\cos(\Delta\lambda))$$

Using $\sin^2\phi + \cos^2\phi = 1$ and trigonometric identities:

$$d = R \cdot \arccos(\cos^2\phi + \cos^2\phi\cos(\Delta\lambda) - \cos^2\phi + \sin^2\phi)$$

This can be simplified to:

$$d = R \cdot \cos\phi \cdot |\Delta\lambda|$$

Interpretation: The effective radius at latitude ϕ is $R\cos\phi$, which decreases as we move away from the equator. This makes sense because circles of latitude get smaller as we approach the poles.

Example 8.4: Same parallel: numerical example

At latitude 60° ($\phi = \pi/3$), the effective radius is $R\cos(\pi/3) = R/2$. So a longitude difference of 1° corresponds to a distance of approximately $R \cdot \frac{1}{2} \cdot \frac{\pi}{180} \approx 55.6$ km, compared to about 111 km at the equator.

8.3.3 Points Near Poles

Example 8.5: Near poles: numerical considerations

When points are near the poles, special numerical considerations are needed.
Issue: Near the poles, small changes in longitude can cause large changes in actual direction, and the formulas can become numerically unstable.
Handling: For points very close to the poles ($|\phi| > 85°$), it's often better to:

- Use the haversine formula for better numerical stability.
- Consider the distance along the parallel as an approximation when appropriate.
- Use special polar coordinate transformations.

Example 8.6: Near-pole numerical example

Two points near the North Pole at 89° N, with longitudes 0° and 180°. The great circle distance is approximately $2R \times (90° - 89°) = 2R \times \frac{\pi}{180} \approx 222$ km, which is much smaller than the distance along the parallel ($\pi R \approx 20,015$ km).

8.3.4 Antipodal Points

When two points are exactly opposite each other on the sphere (antipodal points), the distance is maximized.

Condition: Points are antipodal if $\phi_2 = -\phi_1$ and $\lambda_2 = \lambda_1 \pm \pi \pmod{2\pi}$.

Distance: $d = \pi R$ (half the circumference).

Numerical considerations: The basic formula gives $\arccos(-1) = \pi$, which is exact. However, numerical errors can occur if the coordinates are not exactly antipodal. The haversine formula is more robust in this case.

Multiple geodesics: As we saw in Chapter 6, antipodal points have infinitely many great circles connecting them, all with the same length.

8.3.5 Very Small Distances

For very small distances, we can use approximations that are simpler and more efficient.

Small angle approximation: When the central angle θ is small, we can approximate:

$$\sin\theta \approx \theta, \quad \cos\theta \approx 1 - \frac{\theta^2}{2}$$

For small distances on a sphere, we can use the flat Earth approximation:

$$d \approx R \cdot \sqrt{(\Delta\phi)^2 + (\cos\phi)^2(\Delta\lambda)^2}$$

where ϕ is the mean latitude.

When valid: This approximation is accurate to within about 1% for distances up to about 100 km, and within 0.1% for distances up to about 20 km.

Example 8.7: Flat Earth approximation accuracy

For two points 10 km apart at latitude 45°, the flat Earth approximation gives results accurate to within a few meters.

8.3.6 Very Large Distances

For very large distances (approaching half the circumference), numerical precision becomes important.

Issue: When calculating arccos of values near -1, floating-point precision can cause problems.

Solution: Use the haversine formula, which uses arcsin and is more stable. Alternatively, use:

$$d = R \cdot (\pi - \arccos(-\cos\theta)) = R \cdot (\pi - \arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\Delta\lambda)))$$

when the argument of arccos is near -1.

8.4 Earth-Specific Considerations

8.4.1 Sphere vs Ellipsoid

The Earth is not a perfect sphere—it's an ellipsoid (flattened at the poles). This affects distance calculations.

Earth's shape: The Earth is approximately an oblate spheroid with:

- Equatorial radius: $a \approx 6378.137$ km
- Polar radius: $b \approx 6356.752$ km
- Flattening: $f = \frac{a-b}{a} \approx 0.003353$

Sphere approximation: Using a mean radius $R = 6371$ km introduces errors:

- At the equator: Error is minimal (less than 0.1%)
- At mid-latitudes: Error is typically less than 0.5%
- At the poles: Error can reach about 1%

When sphere is sufficient: For most applications (navigation, flight planning for general aviation), the sphere approximation is adequate. For high-precision applications (surveying, geodesy), ellipsoidal models are needed.

8.4.2 Different Earth Models

Various Earth models are used for different purposes:

- **Mean radius**: $R = 6371$ km (simple, good for general use)
- **WGS84 ellipsoid**: Standard geodetic reference system
- **GRS80 ellipsoid**: Another common geodetic system
- **Local models**: Regional ellipsoids optimized for specific areas

Choosing a model:

- General calculations: Use mean radius $R = 6371$ km
- GPS applications: Use WGS84
- High-precision regional work: Use local ellipsoids

8.4.3 Altitude Considerations

When points are at different altitudes (e.g., aircraft at different flight levels), we need to adjust the radius.

Adjustment: For a point at altitude h above sea level, the effective radius is $R + h$.

Approximate formula: If points are at altitudes h_1 and h_2:

$$d \approx (R + \bar{h}) \cdot \theta$$

where $\bar{h} = \frac{h_1+h_2}{2}$ is the mean altitude and θ is the central angle calculated using sea-level coordinates.

When important: For aircraft navigation, altitude differences can be significant (10-12 km for commercial flights). However, for most ground-based calculations, altitude effects are negligible.

8.5 Numerical Methods and Approximations

8.5.1 Small Distance Approximation

For small distances, we can use simpler formulas that avoid trigonometric functions.

Flat Earth approximation:

$$d \approx R \cdot \sqrt{(\Delta\phi)^2 + (\cos\bar{\phi})^2(\Delta\lambda)^2}$$

where $\bar{\phi} = \frac{\phi_1+\phi_2}{2}$ is the mean latitude.

Accuracy: This approximation has errors of:

- Less than 0.1% for distances up to 20 km
- Less than 1% for distances up to 100 km
- Less than 10% for distances up to 1000 km

Use cases: Quick calculations, preliminary estimates, applications where exact precision isn't needed.

8.5.2 Precision and Floating-Point Issues

Floating-point arithmetic can introduce errors in distance calculations.

arccos issues: When the argument of arccos is near -1 or 1, small floating-point errors can cause large errors in the result.

Example 8.8: arccos precision issue

If $\cos\theta = -0.9999999999$ (due to rounding), then $\theta = \arccos(-0.9999999999) \approx 1.414 \times 10^{-5}$ radians, but the true value might be slightly different, leading to distance errors.

Solution: Use the haversine formula, which uses arcsin and is more stable. The arcsin function is well-behaved near 0 and $\pi/2$.

8.5.3 Fast Computation Methods

For applications requiring many distance calculations, optimization techniques can help:

- **Precomputation**: Calculate $\cos\phi$ and $\sin\phi$ once per point, reuse for multiple distance calculations.
- **Lookup tables**: For fixed precision, precompute common values.
- **Series expansions**: For very small distances, use Taylor series:

  $$\theta \approx \sqrt{(\Delta\phi)^2 + (\cos\phi)^2(\Delta\lambda)^2}$$

  with higher-order corrections if needed.
- **Vectorized computation**: Use vector operations when calculating distances for many point pairs simultaneously.

8.6 Applications Beyond Basic Distance

8.6.1 Finding Intermediate Points

Sometimes we need to find a point at a given distance along a great circle from a starting point.

Problem: Given point P_1 at (ϕ_1, λ_1), find point P_2 at distance d along a great circle in direction α (bearing).

Solution: Using spherical trigonometry:

$$\phi_2 = \arcsin(\sin\phi_1 \cos\frac{d}{R} + \cos\phi_1 \sin\frac{d}{R}\cos\alpha) \tag{8.1}$$

$$\lambda_2 = \lambda_1 + \arctan 2(\sin\alpha \sin\frac{d}{R}\cos\phi_1, \cos\frac{d}{R} - \sin\phi_1 \sin\phi_2) \tag{8.2}$$

Application: Finding waypoints along a flight path, interpolating points on a route.

8.6.2 Distance to a Great Circle

Calculating the shortest distance from a point to a great circle arc.

Problem: Given a point P and a great circle defined by two points P_1 and P_2, find the shortest distance from P to the great circle.

Solution: The distance is the arc length of the perpendicular from P to the great circle plane. This involves:

- Finding the normal vector to the great circle plane
- Calculating the angle between P and this plane
- Converting the angle to distance

Application: Navigation, determining closest approach to a route.

8.6.3 Bearing and Initial Heading

The **bearing** (or azimuth) is the initial direction from one point to another.

Formula: The bearing α from P_1 to P_2 is:

$$\alpha = \arctan 2(\sin \Delta\lambda \cos \phi_2, \cos \phi_1 \sin \phi_2 - \sin \phi_1 \cos \phi_2 \cos \Delta\lambda)$$

Interpretation:

- $0°$ or $360° =$ North
- $90° =$ East
- $180° =$ South
- $270° =$ West

Application: Navigation, determining initial heading for travel between two points.

8.7 Worked Examples: Complex Scenarios

Example 8.9: Multiple cities route

Calculate the total distance for a multi-city route: London → New York → Los Angeles → Tokyo.

Given coordinates:

- London: $(51.5074°, -0.1278°)$
- New York: $(40.7128°, -74.0060°)$
- Los Angeles: $(34.0522°, -118.2437°)$
- Tokyo: $(35.6762°, 139.6503°)$

Calculation: Calculate each leg separately using the haversine formula, then sum:

$$d_{\text{London}\to\text{NY}} \approx 5570 \text{ km} \tag{8.3}$$

$$d_{\text{NY}\to\text{LA}} \approx 3944 \text{ km} \tag{8.4}$$

$$d_{\text{LA}\to\text{Tokyo}} \approx 8767 \text{ km} \tag{8.5}$$

$$d_{\text{total}} \approx 18,281 \text{ km} \tag{8.6}$$

Note: This is the great circle route distance. Actual flight paths may differ due to air traffic control, weather, and other factors.

Example 8.10: Comparing formulas

Calculate the distance between Paris $(48.8566°, 2.3522°)$ and Tokyo $(35.6762°, 139.6503°)$ using different formulas.
Using basic formula:

$$d = 6371 \cdot \arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\Delta\lambda)) \approx 9714 \text{ km}$$

Using haversine formula:

$$d = 2 \cdot 6371 \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \approx 9714 \text{ km}$$

Both formulas give the same result (within numerical precision).
Using flat Earth approximation:

$$d \approx 6371 \cdot \sqrt{(\Delta\phi)^2 + (\cos\bar{\phi})^2(\Delta\lambda)^2} \approx 9710 \text{ km}$$

The approximation is quite good (error less than 0.1%) because the distance is large but not near the limits of the approximation.

Example 8.11: Near-polar distance

Calculate the distance between two points near the North Pole: (85° N, 0° E) and (85° N, 180° E).
Using the formula: Since both points are at the same latitude:

$$d = R \cdot \cos(85°) \cdot |180°| = 6371 \times 0.0872 \times \pi \approx 1744 \text{ km}$$

Note: This is much shorter than the distance along the parallel (which would be approximately $6371 \times \cos(85°) \times \pi \approx 1744$ km as well, since they're on the same parallel). However, the great circle distance is actually shorter, going "over the pole."
Using the full formula accounting for the polar route:

$$d = R{\cdot}\arccos(\sin^2(85°){+}\cos^2(85°)\cos(180°)) = R{\cdot}\arccos(0.9924{-}0.0076) = R{\cdot}\arccos(0.9848) \approx 222 \text{ k}$$

This is the correct great circle distance, going over the pole!

Key Takeaways 6

- **Multiple formulas**: The basic formula, haversine formula, and spherical law of cosines all calculate the same distance, but have different numerical properties.
- **Haversine preferred**: For computational applications, the haversine formula is generally preferred due to better numerical stability.
- **Special cases**: Points on the same meridian or parallel have simplified formulas. Near poles and antipodal points require special handling.
- **Approximations**: For small distances, the flat Earth approximation is accurate and efficient. For large distances, use exact formulas.
- **Earth considerations**: The sphere approximation is sufficient for most applications, but ellipsoidal models are needed for high precision.
- **Practical applications**: Distance calculations enable waypoint finding, bearing calculations, and route planning in navigation systems.

8.8 What's Next?

The concepts we've explored here extend to:

- **Ellipsoidal distance**: More accurate calculations using ellipsoidal models (Vincenty's formulae).
- **General surfaces**: Distance calculations on arbitrary curved surfaces.
- **Computational geometry**: Efficient algorithms for distance queries on large datasets.
- **Navigation systems**: Integration of distance calculations into GPS and other navigation technologies.

Understanding distance calculations on spheres provides the foundation for these advanced applications while being practically useful for many real-world navigation and mapping tasks.

Chapter 9

Open n-Ball

In Chapter 1, we learned that manifolds are locally homeomorphic to Euclidean space. But what exactly does "Euclidean space" mean locally? The answer is: **open n-balls**. These are the fundamental building blocks that define what it means for a space to "look like" Euclidean space locally. This chapter explores open n-balls in detail, showing why they're essential to the manifold definition.

9.1 Introduction: Why Open n-Balls Matter

Recall from Chapter 1 that a manifold is a space where every point has a neighborhood that looks like Euclidean space. More precisely, every point has a neighborhood that is **homeomorphic** to an open ball in $\mathbb{R}^n$.

But what is an open ball? And why is it the "standard" representation of Euclidean space locally? This chapter answers these questions by:

- Defining open n-balls formally
- Showing examples in different dimensions
- Explaining why "open" matters (no boundary)
- Connecting to the manifold definition
- Exploring properties and applications

Understanding open n-balls is crucial because they're what we map to when we say a manifold "locally looks like Euclidean space."

9.2 Definition of an Open n-Ball

9.2.1 Formal Definition

Definition 9.1: Open n-ball

An *open n-ball* in $\mathbb{R}^n$ of radius $r > 0$ centered at $\mathbf{p} \in \mathbb{R}^n$ is

$$B_r(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{p}\| < r\},$$

where $\|\mathbf{x} - \mathbf{p}\|$ is the Euclidean norm. In coordinates, for $\mathbf{p} = (p_1, \ldots, p_n)$ and $\mathbf{x} = (x_1, \ldots, x_n)$,

$$B_r(\mathbf{p}) = \left\{(x_1, \ldots, x_n) \in \mathbb{R}^n : \sum_{i=1}^{n}(x_i - p_i)^2 < r^2\right\}.$$

Components: center $\mathbf{p}$; radius r; strict inequality excludes the boundary.

9.2.2 Why "Open"?

The term "open" is crucial and has a specific mathematical meaning:

- **Open ball**: $\|\mathbf{x} - \mathbf{p}\| < r$ (boundary *not* included)
- **Closed ball**: $\|\mathbf{x} - \mathbf{p}\| \leq r$ (boundary *included*)

Why it matters for manifolds: When we say a manifold is locally homeomorphic to Euclidean space, we need neighborhoods without boundaries. Including boundaries would create problems:

- Boundaries are "special" points that don't have full neighborhoods
- Homeomorphisms would need to map boundaries, which is restrictive
- The local structure should be "smooth" without edge effects

Remark 9.1: A mapping thought experiment

Can you construct a one-to-one, continuous, onto map with continuous inverse from a 2D open disk $B_r(\mathbf{p}) \subset \mathbb{R}^2$ to an open ball in $\mathbb{R}^n$ for $n \neq 2$? *Hint*: Dimension is a topological invariant; by invariance of domain, such a homeomorphism cannot exist unless the dimensions match.

For example, if you're standing on Earth, your local neighborhood (the patch of ground you see) doesn't include a boundary—you can move in any direction. This is what "open" captures.

9.3 Visual Examples in Different Dimensions

9.3.1 1D Open Ball (Open Interval)

In one dimension, an open ball is simply an open interval on the real line.

Definition 9.2: Open 1D Ball

For $p \in \mathbb{R}$ and $r > 0$, the open 1D ball (interval) is

$$B_r(p) \;=\; \{x \in \mathbb{R} : |x - p| < r\} \;=\; (p - r,\, p + r).$$

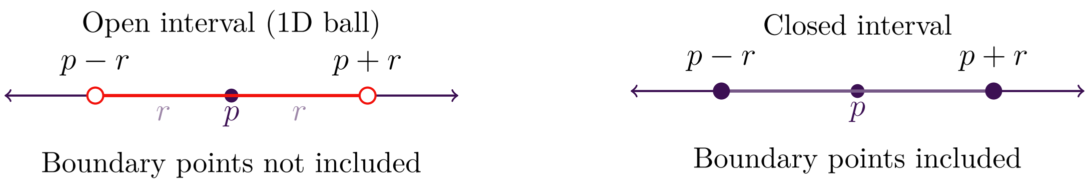


Figure 9.1: Comparison: (a) Open 1D ball (open interval) - boundary points not included. (b) Closed interval - boundary points included.

Example 9.1: 1D open ball

If $p = 0$ and $r = 2$, then $B_2(0) = (-2, 2)$ is the open interval from -2 to 2, excluding the endpoints.

9.3.2 2D Open Ball (Open Disk)

In two dimensions, an open ball is a disk (circle's interior) without its boundary.

Definition 9.3: Open 2D Ball

For $\mathbf{p} = (p_x, p_y) \in \mathbb{R}^2$ and $r > 0$, the open 2D ball (disk) is

$$B_r(\mathbf{p}) \;=\; \{(x, y) \in \mathbb{R}^2 : (x - p_x)^2 + (y - p_y)^2 < r^2\}.$$

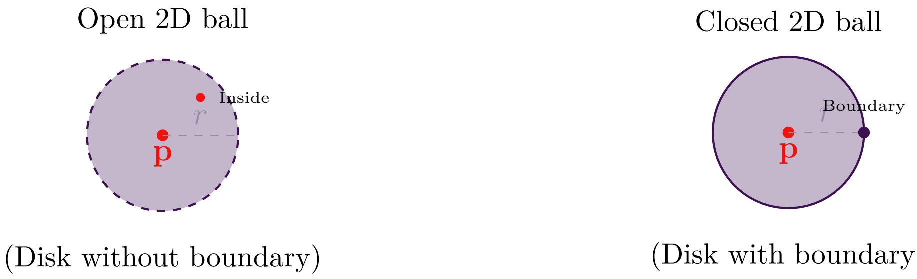


Figure 9.2: Comparison: (a) Open 2D ball (open disk) - boundary circle not included (shown dashed). (b) Closed disk - boundary included (solid circle).

Example 9.2: 2D open ball

The unit disk centered at the origin is $B_1(\mathbf{0}) = \{(x, y) : x^2 + y^2 < 1\}$.

9.3.3 3D Open Ball (Open Sphere)

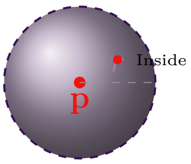
In three dimensions, an open ball is the interior of a sphere (without its surface).

Definition 9.4: Open 3D Ball

For $\mathbf{p} = (p_x, p_y, p_z) \in \mathbb{R}^3$ and $r > 0$, the open 3D ball is

$$B_r(\mathbf{p}) \;=\; \{(x, y, z) \in \mathbb{R}^3 : (x - p_x)^2 + (y - p_y)^2 + (z - p_z)^2 < r^2\}.$$

Open 3D ball



(Sphere interior, surface not included)

Figure 9.3: Open 3D ball: the interior of a sphere. The surface (boundary) is not included, shown here with a dashed outline.

Example 9.3: 3D open ball

The unit ball centered at the origin is $B_1(\mathbf{0}) = \{(x, y, z) : x^2 + y^2 + z^2 < 1\}$.

9.3.4 n-Dimensional Open Ball

The concept extends naturally to any dimension n:

$$B_r(\mathbf{p}) = \left\{(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : \sum_{i=1}^{n} (x_i - p_i)^2 < r^2\right\}$$

Key insight: Even though we can't visualize dimensions beyond 3, the mathematical structure is the same: all points within distance r of the center, with the boundary excluded.

Notation: When the center is the origin $\mathbf{0}$, we often write simply B_r instead of $B_r(\mathbf{0})$.

9.4 Connection to Manifold Definition

9.4.1 The Role of Open Balls in Manifolds

The formal definition of a manifold requires that every point has a neighborhood that is homeomorphic to an open ball in $\mathbb{R}^n$. This is what we mean when we say a manifold "locally looks like Euclidean space."

Formal statement: A space M is an n-dimensional manifold if for every point $p \in M$, there exists:

- An open neighborhood U of p in M
- A homeomorphism $\phi : U \to B_r(\mathbf{0}) \subset \mathbb{R}^n$ (an open ball)

The pair (U, ϕ) is called a **chart**, and ϕ maps the local neighborhood on the manifold to an open ball in Euclidean space.

9.4.2 Visual Connection

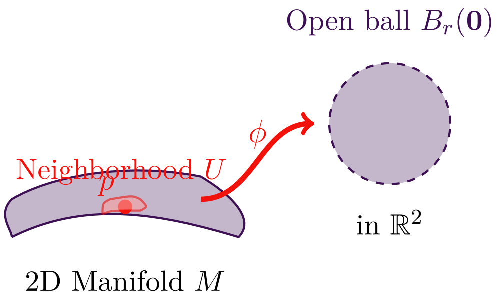


Figure 9.4: The manifold definition: Every point p on a manifold M has a neighborhood U that maps via a homeomorphism ϕ to an open ball in $\mathbb{R}^n$. This is what "locally looks like Euclidean space" means precisely.

9.4.3 Dimension Matching

Crucially, the dimension of the open ball must match the dimension of the manifold:

- A 1D manifold (like a curve) requires neighborhoods homeomorphic to open 1D balls (intervals).
- A 2D manifold (like a sphere's surface) requires neighborhoods homeomorphic to open 2D balls (disks).
- An n-dimensional manifold requires neighborhoods homeomorphic to open n-dimensional balls.

This dimension matching is essential: you can't map a 2D surface patch to a 1D interval without losing information.

9.5 Distance Calculations in Open n-Balls

9.5.1 Euclidean Distance Formula

Within an open n-ball, we use the standard Euclidean distance formula. For two points $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ in $\mathbb{R}^n$:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} = \|\mathbf{p} - \mathbf{q}\|$$

This is the same formula we've used in previous chapters for distances on planes and in Euclidean space.

9.5.2 Examples in Different Dimensions

1D Distance

In a 1D open ball (interval), the distance is simply:

$$d(p, q) = |p - q|$$

Example 9.4: Distance in $(-2, 2)$

In the open interval $(-2, 2)$, the distance between $p = -1$ and $q = 1.5$ is:

$$d(-1, 1.5) = |1.5 - (-1)| = 2.5$$

2D Distance

In a 2D open ball (disk), the distance is:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

Example 9.5: Distance in the unit disk

In the unit disk centered at the origin, for points $\mathbf{p} = (0.3, 0.4)$ and $\mathbf{q} = (0.7, 0.2)$:

$$d = \sqrt{(0.7 - 0.3)^2 + (0.2 - 0.4)^2} = \sqrt{0.16 + 0.04} = \sqrt{0.2} \approx 0.447$$

3D Distance

In a 3D open ball, the distance is:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$$

Example 9.6: Distance in the unit ball

In the unit ball, for points $\mathbf{p} = (0.5, 0.5, 0.5)$ and $\mathbf{q} = (0.8, 0.3, 0.6)$:

$$d = \sqrt{(0.3)^2 + (-0.2)^2 + (0.1)^2} = \sqrt{0.09 + 0.04 + 0.01} = \sqrt{0.14} \approx 0.374$$

n-Dimensional Distance

The general formula works for any dimension:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

9.5.3 Properties of Distance in Open Balls

Maximum Distance Within a Ball

Definition 9.5: Diameter via supremum

The maximum distance between any two points in an open ball $B_r(\mathbf{p})$ is strictly less than $2r$ (the diameter). Since the ball is open, we can approach but never attain $2r$. Formally,

$$\sup\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in B_r(\mathbf{p})\} = 2r.$$

Here sup (*supremum*) is the least upper bound of a set: the smallest number that is greater than or equal to every element in the set, not necessarily achieved by any element.

Note: The supremum is $2r$, but this maximum is never actually achieved because the ball is open (boundary excluded).

Triangle Inequality

Theorem 9.1: Triangle inequality

For any three points $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$ in an open ball,

$$d(\mathbf{p}, \mathbf{r}) \leq d(\mathbf{p}, \mathbf{q}) + d(\mathbf{q}, \mathbf{r}).$$

This is a fundamental property of Euclidean distance that holds in any dimension.

9.6 Properties of Open n-Balls

9.6.1 Topological Properties

Open n-balls have several important topological properties:

Open Set

An open ball is itself an **open set**: every point in the ball has a neighborhood entirely contained within the ball. This means:

- For any point $\mathbf{q} \in B_r(\mathbf{p})$, there exists $\epsilon > 0$ such that $B_\epsilon(\mathbf{q}) \subset B_r(\mathbf{p})$.
- You can always "shrink" around any interior point and stay inside.

Connected

Definition 9.6: Connected set

A set is *connected* if it cannot be written as the disjoint union of two nonempty open sets in the subspace topology. Equivalently, any two points can be joined by a continuous path lying in the set. An open ball is connected.

Remark 9.2: Convex vs. concave

A set is *convex* if for any two points in it, the entire line segment between them lies in the set. “Concave” typically describes functions or shapes that curve inward; in set terms we usually contrast convex with nonconvex rather than “concave.”

Simply Connected

An open ball is **simply connected**: any loop (closed path) can be continuously shrunk to a point while staying in the ball. There are no "holes" in an open ball.

9.6.2 Geometric Properties

Convex

An open ball is **convex**: the line segment between any two points in the ball is entirely contained in the ball. In other words, if $\mathbf{p}, \mathbf{q} \in B_r(\mathbf{c})$, then for all $t \in [0, 1]$:

$$(1 - t)\mathbf{p} + t\mathbf{q} \in B_r(\mathbf{c})$$

This is a key geometric property that makes balls "round" and symmetric.

Bounded

An open ball is **bounded**: all points are within a finite distance from the center. Specifically, every point $\mathbf{x} \in B_r(\mathbf{p})$ satisfies:

$$\|\mathbf{x} - \mathbf{p}\| < r < \infty$$

Symmetric

An open ball is **symmetric**: it looks the same in all directions from the center. This symmetry is a consequence of the Euclidean metric, which treats all directions equally.

9.6.3 Comparison with Closed Balls

It’s instructive to compare open and closed balls:

Why open balls for manifolds?: The manifold definition requires neighborhoods without boundaries because:

- Boundaries create "edge effects" that complicate local structure
- Homeomorphisms are cleaner without boundary constraints

Property	Open Ball	Closed Ball
Boundary	Not included	Included
Definition	$\|\mathbf{x} - \mathbf{p}\| < r$	$\|\mathbf{x} - \mathbf{p}\| \leq r$
Open set?	Yes	No
For manifolds	Required	Not suitable

Table 9.1: Comparison of open and closed balls.

- The local structure should be "smooth" and unrestricted

9.7 Open Balls as Neighborhoods

9.7.1 Neighborhood Concept

A **neighborhood** of a point $\mathbf{p}$ is any open set containing $\mathbf{p}$. Open balls are the most fundamental type of neighborhood.

Key fact: Any neighborhood of a point contains an open ball around that point. This means open balls are the "smallest" or "most basic" neighborhoods we can use.

9.7.2 Why This Matters for Manifolds

The manifold definition states: "Every point has a neighborhood homeomorphic to an open ball." This is equivalent to saying "locally looks like Euclidean space" because:

- Open balls are the standard neighborhoods in Euclidean space
- Any neighborhood in Euclidean space contains an open ball
- So being homeomorphic to a neighborhood is equivalent to being homeomorphic to an open ball

This is why open balls are so fundamental—they're what we mean by "Euclidean space locally."

9.7.3 Visual: Zooming In

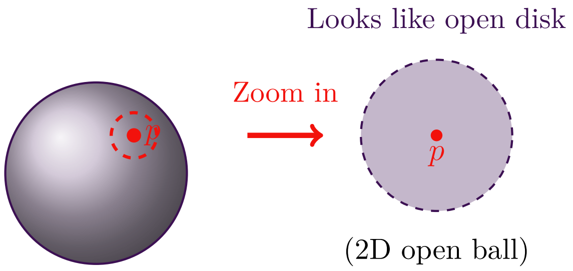


Figure 9.5: Zooming in on a manifold: As you zoom in on any point, the local neighborhood looks more and more like an open ball in Euclidean space. This is the essence of local flatness.

This visualization captures the intuitive idea: no matter where you are on a manifold, if you zoom in enough, your local view looks like a flat open ball.

9.8 Special Cases and Examples

9.8.1 Unit Ball

The **unit ball** is the open ball of radius 1 centered at the origin:

$$B_1(\mathbf{0}) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$$

In different dimensions:

- **1D**: $(-1, 1)$ - the open interval from -1 to 1
- **2D**: $\{(x, y) : x^2 + y^2 < 1\}$ - the unit disk
- **3D**: $\{(x, y, z) : x^2 + y^2 + z^2 < 1\}$ - the unit sphere (interior)
- **nD**: $\{\mathbf{x} : \|\mathbf{x}\| < 1\}$ - the n-dimensional unit ball

The unit ball is often used as a standard reference because it's simple and symmetric.

9.8.2 Balls at Different Centers

Any open ball $B_r(\mathbf{p})$ is just a translation of the unit ball:

$$B_r(\mathbf{p}) = \mathbf{p} + r \cdot B_1(\mathbf{0})$$

This means all open balls are geometrically equivalent (homeomorphic) regardless of their center or radius.

9.8.3 Relationship to Other Shapes

Open balls are not the only open sets that could work for manifolds, but they're the standard choice:

- **Open ball**: Round, symmetric, standard choice
- **Open rectangle/cube**: Also works, but not symmetric
- **Open ellipse**: Also works, but more complex

Why balls?:

- Symmetry makes them natural
- Simplicity in calculations
- Standard in mathematics
- All open neighborhoods contain balls anyway

9.9 Worked Examples

Example 9.7: Is a point in an open ball?

Question: Is the point $(2, 3)$ in the open ball centered at $(1, 1)$ with radius $r = 2.5$?
Solution: Check if the distance is less than the radius:

$$d((2,3),(1,1)) = \sqrt{(2-1)^2 + (3-1)^2} = \sqrt{1+4} = \sqrt{5} \approx 2.236$$

Since $2.236 < 2.5$, the point $(2, 3)$ is inside the open ball.

Example 9.8: Distance between two points in a ball

Question: Calculate the distance between points $(0.5, 0.5)$ and $(0.8, 0.7)$ in the unit ball $B_1(\mathbf{0})$.
Solution: First, verify both points are in the unit ball:

- For $(0.5, 0.5)$: $0.5^2 + 0.5^2 = 0.5 < 1$ (inside)
- For $(0.8, 0.7)$: $0.8^2 + 0.7^2 = 0.64 + 0.49 = 1.13 > 1$ (outside!)

Since $(0.8, 0.7)$ is outside the unit ball, let's use $(0.6, 0.6)$ instead, which is inside $(0.6^2 + 0.6^2 = 0.72 < 1)$.
Now calculate the distance between $(0.5, 0.5)$ and $(0.6, 0.6)$:

$$d = \sqrt{(0.6-0.5)^2 + (0.6-0.5)^2} = \sqrt{0.01+0.01} = \sqrt{0.02} \approx 0.141$$

Both points are in the unit ball, and their distance is approximately 0.141.

Example 9.9: Largest contained ball

Question: Given a rectangular region $R = \{(x, y) : 0 < x < 3, 0 < y < 2\}$, what's the largest open ball that fits entirely in R?
Solution: The largest ball will be limited by the closest boundary. The center should be at the center of the rectangle: $(1.5, 1.0)$. The distance to the nearest boundary is $\min(1.5, 1.0, 1.5, 1.0) = 1.0$.
So the largest open ball is $B_1((1.5, 1.0))$ with radius $r = 1.0$.

Key Takeaways 7

- **Definition**: An open n-ball $B_r(\mathbf{p})$ is the set of points within distance r of center $\mathbf{p}$, with the boundary excluded.
- **Role in manifolds**: Manifolds are locally homeomorphic to open n-balls—this is the precise meaning of "locally looks like Euclidean space."
- **Dimension matching**: An n-dimensional manifold requires neighborhoods homeomorphic to n-dimensional open balls.
- **Why "open"?**: Excluding the boundary is essential for manifolds—it ensures smooth local structure without edge effects.
- **Distance**: Within open balls, we use standard Euclidean distance: $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum(p_i - q_i)^2}$.
- **Properties**: Open balls are open sets, connected, convex, bounded, and symmetric.
- **Standard neighborhoods**: Open balls are the fundamental neighborhoods in Euclidean space, making them the natural choice for manifold definitions.

9.10 What's Next?

Understanding open n-balls provides the foundation for:

- **Charts and atlases**: Collections of open ball mappings that cover a manifold
- **Smooth manifolds**: Manifolds with differentiable structure
- **Differential geometry**: The study of geometric structures on manifolds
- **Advanced topics**: Riemannian manifolds, Lie groups, and more

Open n-balls are the building blocks that make the manifold concept precise and mathematically rigorous.

Chapter 10

Riemannian Manifolds

10.1 From Manifolds to Riemannian Manifolds

Up to now, we have treated manifolds as spaces that locally look like Euclidean space (via charts) and we have computed distances and geodesics in concrete cases (lines, circles, planes, spheres). To make these ideas fully precise and unify all previous formulas, we introduce a **Riemannian metric**: a smoothly varying inner product on each tangent space that tells us how to measure lengths, angles, and distances on a manifold.

At an intuitive level, a Riemannian metric is a field of "local rulers" that may vary from point to point. These rulers determine how long a small displacement is and how two directions compare via an angle. Once a metric is given, the manifold becomes a **Riemannian manifold**.

10.2 Tangent Spaces and Vectors on a Manifold

To measure lengths and angles on a manifold, we first need a notion of direction at a point. The **tangent space** $T_p\mathcal{M}$ at a point p collects all possible "velocities" of curves passing through p.

Definition 10.1: Tangent Space

Let $\mathcal{M}$ be a smooth manifold and $p \in \mathcal{M}$. The *tangent space* $T_p\mathcal{M}$ is the vector space of equivalence classes of smooth curves γ with $\gamma(0) = p$, where $\gamma_1 \sim \gamma_2$ if, in any chart, $\frac{d}{dt}(x \circ \gamma_1)(0) = \frac{d}{dt}(x \circ \gamma_2)(0)$. Equivalently, $T_p\mathcal{M}$ is spanned by the coordinate derivations $\{\partial/\partial x^i|_p\}$.

10.2.1 Tangent Vectors as Velocities of Curves

Let $\gamma : (-\epsilon, \epsilon) \to \mathcal{M}$ be a smooth curve with $\gamma(0) = p$. The derivative $\dot{\gamma}(0)$ is a tangent vector at p. Two curves define the same tangent vector if their derivatives agree in any (hence every) chart. The set of all such velocities forms a vector space $T_p\mathcal{M}$.

10.2.2 Coordinate Bases and Change of Coordinates

Given a chart $(U, x^1, \ldots, x^n)$ with $p \in U$, the partial derivatives $\left\{ \frac{\partial}{\partial x^1}\Big|_p, \ldots, \frac{\partial}{\partial x^n}\Big|_p \right\}$ form a basis of $T_p\mathcal{M}$. A tangent vector can be written

$$v\Big|_p = v^i \frac{\partial}{\partial x^i}\Big|_p, \quad v^i \in \mathbb{R}.$$

Under a coordinate change $x \mapsto y$, components transform by the Jacobian: $v^i = \frac{\partial x^i}{\partial y^j} v^j$. The vector itself is geometric (coordinate-independent); only its components change.

10.2.3 Differential (Pushforward) and the Jacobian

For a smooth map $F : \mathcal{M} \to \mathcal{N}$, the **differential** at p, $dF_p : T_p\mathcal{M} \to T_{F(p)}\mathcal{N}$, pushes tangent vectors forward by $dF_p(\dot\gamma(0)) = \frac{d}{dt}\Big(F \circ \gamma\Big)(0)$. In coordinates, dF_p is represented by the Jacobian matrix $\Big(\partial F^\alpha / \partial x^i\Big)$. This notion underlies pullback metrics and the geometry of learned manifolds (Chapter 10).

10.2.4 Examples

- $\mathbb{R}^n$: $T_p\mathbb{R}^n \cong \mathbb{R}^n$ with the standard basis.
- **Circle** S^1: At angle θ, T_pS^1 is the line tangent to the circle, spanned by $\partial/\partial\theta$; velocities are perpendicular to the radius.
- **Sphere** S^2: T_pS^2 is the plane tangent to the sphere at p; great-circle motion has velocity in this plane.

10.3 The Riemannian Metric

10.3.1 Definition (Intuition first)

For each point p on a manifold $\mathcal{M}$, consider the tangent space $T_p\mathcal{M}$. A Riemannian metric assigns to each p an inner product $g_p(\cdot,\cdot)$ on $T_p\mathcal{M}$ that varies smoothly with p. This inner product makes it possible to measure:

- **Lengths** of tangent vectors and curves
- **Angles** between tangent directions
- **Distances** between points (as minimal curve lengths)

10.3.2 Coordinates and the Metric Tensor

In a local coordinate chart $(x^1, \ldots, x^n)$, the metric is represented by a symmetric, positive-definite matrix of smooth functions $g_{ij}(x)$, and the squared infinitesimal length is

$$ds^2 = g_{ij}(x)\, dx^i\, dx^j, \quad g_{ij} = g_{ji}, \quad (g_{ij}) \text{ positive definite.}$$

Under a coordinate change, the components g_{ij} transform as a $(0,2)$-tensor, ensuring ds^2 is coordinate-independent.

10.3.3 Metric Tensor: Coordinate Form and Properties

The metric is a smoothly varying *(0,2)-tensor field* g, meaning that at each point p it takes two tangent vectors and returns a scalar $g_p(u, v)$, bilinear, symmetric, and positive-definite. In coordinates:

- **Bilinearity and symmetry**: $g_{ij} = g_{ji}$; linear in each argument.
- **Positive-definite**: $g_{ij}v^iv^j > 0$ for any nonzero v.
- **Line element**: $ds^2 = g_{ij}\, dx^idx^j$ defines squared infinitesimal length.
- **Coordinate change**: If $y = y(x)$, then $g'_{\alpha\beta} = \frac{\partial x^i}{\partial y^\alpha} \frac{\partial x^j}{\partial y^\beta}\, g_{ij}$, leaving ds^2 invariant.

Important constructions:

- **Inverse metric**: g^{ij} satisfies $g^{ik}g_{kj} = \delta^i_j$; used to raise/lower indices and map vectors to covectors.
- **Volume element**: $dV = \sqrt{\det g}\, dx^1 \cdots dx^n$ gives area/volume in coordinates.
- **Levi-Civita connection**: The unique torsion-free connection compatible with g has Christoffel symbols $\Gamma^i_{\ jk}$ built from g and its first derivatives; geodesics and curvature derive from this.
- **Pullback metric**: For $F : \mathcal{Z} \to \mathcal{M}$, the induced metric on $\mathcal{Z}$ is $g_{\mathcal{Z}} = J_F^\top\, g_{\mathcal{M}}\, J_F$; with ambient Euclidean $g = I$, this reduces to $J_F^\top J_F$ (learned manifolds in Chapter 10).

10.4 Examples of Riemannian Metrics

10.4.1 Euclidean Space

On $\mathbb{R}^n$ with Cartesian coordinates, the standard metric is $g_{ij} = \delta_{ij}$. Then $ds^2 = \sum_i (dx^i)^2$, which recovers familiar Euclidean geometry.

10.4.2 The Circle

On a circle of radius r with angle coordinate θ, the metric is $ds^2 = r^2\, d\theta^2$. Curve length along the circle is the usual arc length $L = \int r\, |\dot\theta|\, dt$.

10.4.3 The Sphere

On a sphere of radius R with spherical coordinates (ϕ, λ) (latitude, longitude),

$$ds^2 = R^2 \Big(d\phi^2 + \sin^2\!\phi\, d\lambda^2 \Big),$$

the *round metric*, consistent with great-circle distances studied earlier.

10.5 Distances from the Metric

Given a smooth curve $\gamma : [t_1, t_2] \to \mathcal{M}$, its length with respect to g is

$$L[\gamma] = \int_{t_1}^{t_2} \sqrt{g_{ij}(\gamma(t))\, \dot{x}^i(t)\, \dot{x}^j(t)}\, dt.$$

The **distance** between two points $p, q \in \mathcal{M}$ is the infimum of $L[\gamma]$ over all smooth curves γ with $\gamma(t_1) = p$ and $\gamma(t_2) = q$. This construction unifies all earlier distance formulas:

- On a line ($g = 1$), $d(p, q) = |x_2 - x_1|$.
- On a circle ($g = r^2$), $d = r\,\theta$ (shorter arc).
- On a sphere (round metric), d is the great-circle (central-angle) distance.

10.6 Affine Connection and Covariant Derivative

To differentiate vector fields along curves on a manifold, we need a rule that compares vectors at nearby points. An **affine connection** (or simply a connection) ∇ provides the *covariant derivative* $\nabla_X Y$ of a vector field Y along a vector field X.

Definition 10.2: Affine Connection

An *affine connection* ∇ assigns to vector fields X, Y a vector field $\nabla_X Y$ such that: (i) it is $\mathbb{R}$-bilinear, (ii) C^∞-linear in X, (iii) obeys the Leibniz rule $\nabla_X(fY) = X[f]Y + f\,\nabla_X Y$, and (iv) is tensorial in Y.

10.6.1 Concept and properties

The covariant derivative obeys linearity in X and Y, the Leibniz rule $\nabla_X(fY) = X[f]\,Y + f\,\nabla_X Y$, and reduces to ordinary differentiation in flat coordinates. It enables us to describe how vectors change along curves intrinsically.

10.6.2 Christoffel symbols in coordinates

In a chart with coordinate vector fields ∂_i, the connection is encoded by **Christoffel symbols** Γ^k_{ij} via

$$\nabla_{\partial_i} \partial_j \;=\; \Gamma^k_{ij}\, \partial_k.$$

These symbols are not tensor components (they do not transform tensorially), but they assemble to a geometric object: the connection itself.

10.6.3 Levi-Civita connection from the metric

For a Riemannian metric g, there exists a unique connection that is *torsion-free* ($\Gamma^k_{ij} = \Gamma^k_{ji}$) and *metric-compatible* ($\nabla g = 0$). This **Levi-Civita connection** has Christoffel symbols

$$\Gamma^i_{jk} \;=\; \tfrac{1}{2}\, g^{i\ell}\Big(\partial_j g_{k\ell} + \partial_k g_{j\ell} - \partial_\ell g_{jk}\Big),$$

expressing how g determines differentiation, parallel transport, and geodesics.

10.6.4 Parallel Transport

Definition and intuition. Given a curve $\gamma : [0,1] \to \mathcal{M}$ and an initial vector $V(0) \in T_{\gamma(0)}\mathcal{M}$, the **parallel transport** of V along γ is the vector field $V(t)$ along γ satisfying the ODE

$$\nabla_{\dot\gamma(t)} V(t) \;=\; 0, \quad V(0) \text{ given.}$$

Under the Levi-Civita connection, transport preserves inner products, so lengths and angles of transported vectors remain constant.

Definition 10.3: Parallel Transport

On a manifold with connection $(\mathcal{M}, \nabla)$, a vector field V along a curve γ is *parallel* if $\nabla_{\dot\gamma} V = 0$. The map $P_{0\to t} : T_{\gamma(0)}\mathcal{M} \to T_{\gamma(t)}\mathcal{M}$ taking $V(0)$ to $V(t)$ is the *parallel transport* along γ.

Coordinate form. In local coordinates, with components $V^i(t)$ and $\dot\gamma^j(t)$,

$$\frac{dV^i}{dt} + \Gamma^i_{jk}\Big(\gamma(t)\Big)\,\dot\gamma^j(t)\,V^k(t) \;=\; 0,$$

an initial value problem with unique smooth solution for smooth data.

Properties. For the Levi-Civita connection: (i) inner products are preserved, (ii) transport depends on the path in general (path independence holds in flat regions/coordinates like Cartesian $\mathbb{R}^n$), (iii) concatenation of paths composes transports.

Examples.

- **Plane**: In Cartesian coordinates, $\Gamma = 0$, so $dV/dt = 0$ and V is constant; transport is path-independent.
- **Sphere**: Transporting a tangent vector along a latitude or a spherical triangle generally changes its direction upon return, revealing positive curvature.

Relation to geodesics. Geodesics are *auto-parallel*: their velocity is parallel transported along themselves, $\nabla_{\dot\gamma}\dot\gamma = 0$. Thus geodesics are "straightest" curves with respect to the connection.

Holonomy and curvature (brief). Transporting a vector around a closed loop typically produces a rotated vector at the base point; the net rotation (holonomy) encodes curvature. On flat manifolds (or regions), the holonomy is trivial.

Note for learned manifolds. For manifolds parameterized by generators G, parallel transport under the pullback metric can be approximated by integrating the coordinate ODE, enabling semantics-preserving traversals across the data manifold.

10.6.5 Exponential and Log Maps; Normal Coordinates

Exponential map. For $v \in T_p\mathcal{M}$, let γ_v be the unique geodesic with $\gamma_v(0) = p$ and $\dot{\gamma}_v(0) = v$. The **exponential map** is

$$\exp_p(v) \;=\; \gamma_v(1),$$

defined at least on a neighborhood of $0 \in T_p\mathcal{M}$. It converts a tangent vector into a point by "shooting" along its geodesic for unit time.

Log map (where defined). On a normal neighborhood U of p, the inverse map $\log_p : U \to T_p\mathcal{M}$ takes a point q to the initial velocity of the minimizing geodesic from p to q.

Normal coordinates. Using $\exp_p$, one defines **normal coordinates** centered at p; in these coordinates, geodesics through p are straight lines, Christoffel symbols vanish at p, and $g_{ij}(p) = \delta_{ij}$, so the metric is Euclidean up to first order.

10.6.6 Geodesic Deviation and Curvature (Jacobi Fields)

Consider a smooth family of geodesics $\gamma_s(t)$ with variation field $J(t) = \partial\gamma_s/\partial s\big|_{s=0}$. Then J satisfies the (coordinate-free) **Jacobi equation**

$$\nabla_{\dot{\gamma}}\nabla_{\dot{\gamma}}J \;+\; R(J, \dot{\gamma})\,\dot{\gamma} \;=\; 0,$$

where R is the Riemann curvature tensor. Qualitatively: positive curvature tends to focus (converge) nearby geodesics, negative curvature defocuses (diverges), and zero curvature keeps separation linear.

10.6.7 Curvature: Types and Intuition

- **Gaussian curvature** (2D): intrinsic measure of bending; triangles have angle sum $\pi + (\text{area}) \cdot K$ (small triangles). Sphere: $K > 0$; saddle: $K < 0$; plane: $K = 0$.
- **Sectional curvature** (general): curvature of a 2D plane in $T_p\mathcal{M}$; reduces to Gaussian curvature in 2D.
- **Ricci/scalar curvature** (brief): averages of sectional curvature controlling volume distortion and heat flow; useful summaries in higher dimensions.

Visualization: on a sphere, initially parallel geodesics (longitudes) meet; on a saddle, they spread apart; on a plane, they stay parallel.

10.6.8 Cut Locus, Conjugate Points, and Injectivity Radius

Geodesics minimize distance only up to certain limits. The **cut locus** of p is where minimizing geodesics from p first fail to be unique or to minimize. **Conjugate points** are points along a geodesic where nearby geodesics reconverge (nontrivial Jacobi fields vanish). The **injectivity radius** at p is the largest radius for which $\exp_p$ is a diffeomorphism onto its image; within it, $\log_p$ and geodesic uniqueness/minimality hold.

10.6.9 Computation and Examples

- **Sphere (S^2)**: Great circles are geodesics. With unit vectors $u, v \in S^2$, the central angle $\theta = \arccos(u \cdot v)$ gives the distance $d = R\theta$. The log map at u can be written using the component of v orthogonal to u; exp is its inverse along great circles.
- **Plane ($\mathbb{R}^n$)**: $\Gamma = 0$ in Cartesian coordinates; exp/log are $\exp_p(v) = p + v$, $\log_p(q) = q - p$; transport and geodesics are path-independent straight lines.
- **Cylinder** (flat): Locally like a plane; unroll-and-roll constructions provide geodesics and transport.

Numerics: Boundary-value geodesics (given endpoints) can be solved with shooting methods; stability improves in normal coordinates and with good initial guesses.

10.6.10 Key Takeaways: Geodesics and Curvature

- **exp/log**: Bridge between tangent spaces and points; normal coordinates make geodesics look straight near their base.
- **Curvature**: Determines how geodesics deviate, how transport around loops rotates vectors, and where geodesics stop minimizing.
- **Global limits**: Cut locus and injectivity radius bound geodesic uniqueness/minimality.

10.6.11 Geodesics via the connection

Geodesics are the curves whose velocity vectors are *auto-parallel*: $\nabla_{\dot\gamma}\dot\gamma = 0$. In coordinates, this condition yields the geodesic ODE with Christoffel symbols. This connects the variational and "straightest-possible" viewpoints.

10.7 Geodesics Determined by the Metric

10.7.1 Variational Characterization

Geodesics are curves that are locally length-minimizing for the metric g. Equivalently, they are stationary points of the length functional. In coordinates, geodesics satisfy the **geodesic equation**

$$\frac{d^2x^i}{dt^2} + \Gamma^i_{jk}(x)\,\frac{dx^j}{dt}\,\frac{dx^k}{dt} = 0,$$

where the Christoffel symbols are determined by the metric via

$$\Gamma^i_{jk} \;=\; \tfrac{1}{2}\,g^{i\ell}\Big(\partial_j g_{k\ell} + \partial_k g_{j\ell} - \partial_\ell g_{jk}\Big).$$

This shows the metric fully determines geodesics.

10.7.2 Consistency with Previous Chapters

- **Plane**: $g_{ij} = \delta_{ij} \Rightarrow \Gamma^i_{jk} = 0$, geodesics are straight lines.
- **Circle**: $ds^2 = r^2 d\theta^2$ gives uniform motion in θ (arcs).
- **Sphere**: The geodesic equations yield great circles, matching our geometric construction.

10.8 Curvature: How Metrics Encode Shape

The metric encodes how the space bends intrinsically. In two dimensions, the Gaussian curvature distinguishes flat ($K = 0$), positively curved (sphere-like, geodesics converge), and negatively curved (saddle-like, geodesics diverge) behavior—precisely the phenomena observed in earlier chapters.

10.9 Why Riemannian Structure Matters

- It provides the rigorous framework for length, distance, and angle on manifolds.
- It explains and unifies geodesics across all examples we studied.
- It prepares us for applications: learned manifolds in generative AI induce metrics (via pullbacks), affecting interpolation and distance.

Key Takeaways 8

- A **Riemannian manifold** is a manifold equipped with a smoothly varying inner product on tangent spaces.
- The **metric tensor** g_{ij} defines local lengths and angles and determines distances and geodesics.
- Classical cases (line, circle, plane, sphere) fit naturally into this framework.
- Curvature (from the metric) governs how geodesics converge or diverge.

10.10 What Comes Next

In the next chapter, we connect these ideas to generative AI: neural generators endow data manifolds with *pullback metrics*, enabling geodesic interpolation and principled distance notions that align better with perception than raw Euclidean distances.

Chapter 11

Manifolds in Generative AI

This chapter bridges the mathematical foundations we've built—manifolds, geodesics, distances, and local structure—with modern generative artificial intelligence. We'll see how the abstract concepts of differential geometry become concrete tools for understanding and building AI systems that generate images, text, audio, and other complex data.

Example 11.1: Text generation (e.g., ChatGPT)

Viewing model parameters as a *parameter manifold* with the Fisher metric yields *natural gradient* updates that follow geodesics in information space, often converging faster than Euclidean gradients. Inference traverses a low-dimensional *data manifold* of plausible token sequences, where curvature helps explain why small Euclidean steps can produce large semantic shifts and why geometry-aware sampling (e.g., temperature, nucleus) stabilizes outputs.

Example 11.2: Image generation (diffusion, GANs, flows)

Diffusion models learn score fields that align with the manifold of natural images; denoising follows integral curves tangent to this manifold. Normalizing flows learn *diffeomorphisms* between a simple latent manifold and image space; the pullback metric clarifies conditioning and mode coverage. In GANs, geodesic distances in latent space better preserve semantics than Euclidean pixel metrics, guiding interpolation and editing.

Example 11.3: Video generation

Video lives on a manifold with temporal constraints (smooth trajectories). Geodesics in latent space produce temporally coherent frames; parallel transport transfers motion styles between clips. Information geometry (Fisher) improves training stability for sequence models by respecting curvature induced by autoregressive likelihoods, reducing exposure bias and drift.

11.1 Introduction: From Mathematics to AI

Throughout this book, we've explored:

- **Manifolds**: Spaces that locally look like Euclidean space, with charts providing local coordinate systems
- **Geodesics**: Shortest paths on curved surfaces, generalizing straight lines
- **Distances**: Arc lengths along geodesics connecting points
- **Open n-balls**: Local neighborhoods that define manifold structure

Definition 11.1: Parameter Manifold

A *parameter manifold* is a differentiable manifold Θ of model parameters endowed with a metric (e.g., the Fisher metric or a pullback metric), which induces lengths/angles of parameter displacements and defines geometry-aware optimization.

These mathematical concepts are not just abstract theory—they directly describe the structure of real-world data and provide the foundation for modern generative AI systems. In this chapter, we'll see how.

11.2 The Data Manifold Hypothesis

11.2.1 High-Dimensional Data on Low-Dimensional Manifolds

Consider a 256×256 grayscale image. In its raw form, this is a vector in $\mathbb{R}^{65536}$ (one dimension per pixel). However, not every point in this 65,536-dimensional space corresponds to a meaningful image. Most points would be random noise. The set of all possible natural images forms a much smaller subset—a **manifold** embedded in this high-dimensional space.

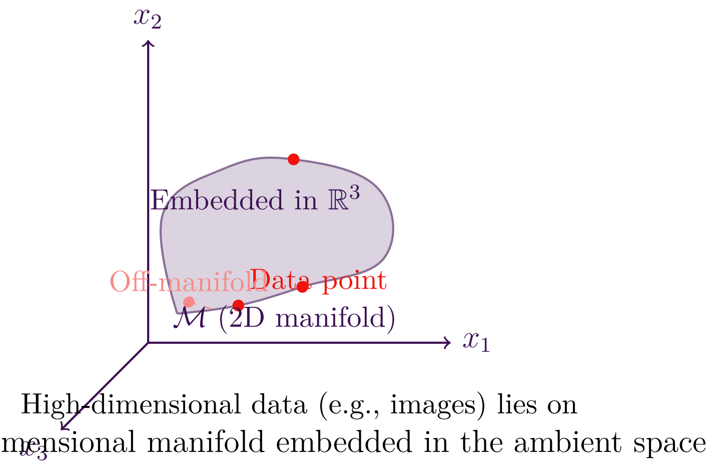

x_2
Embedded in $\mathbb{R}^3$
Off-manifold
Data point
$\mathcal{M}$ (2D manifold)
x_1
x_3
High-dimensional data (e.g., images) lies on
a low-dimensional manifold embedded in the ambient space

Figure 11.1: The data manifold hypothesis: high-dimensional data points (like images) lie on a low-dimensional manifold $\mathcal{M}$ embedded in the ambient space. Points off the manifold are typically invalid or meaningless.

Remark 11.1: Ambient Space

The ambient space is the high-dimensional space in which the data lies. For example, the ambient space for an image is $\mathbb{R}^{256\times256\times3}$.

Remark 11.2: Latent Space

The latent space is the low-dimensional space in which the data is represented. For example, the latent space for an image is $\mathbb{R}^{50-200}$.

11.2.2 Mathematical Formulation

Formally, the **data manifold hypothesis** states:

Definition 11.2: Data Manifold

Given a dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ where each $\mathbf{x}_i \in \mathbb{R}^D$ (high-dimensional space), there exists a lower-dimensional manifold $\mathcal{M} \subset \mathbb{R}^D$ of intrinsic dimension $d \ll D$ such that:

$$\mathcal{D} \approx \{\mathbf{x} \in \mathcal{M} : \mathbf{x} = f(\mathbf{z}), \mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^d\} \tag{11.1}$$

where $f : \mathcal{Z} \to \mathcal{M}$ is a smooth mapping from the latent space $\mathcal{Z}$ to the data manifold $\mathcal{M}$.

The intrinsic dimension d is typically much smaller than the ambient dimension D. For example:

- **Images**: $D = 65536$ (256×256), but $d \approx 50 - 200$ for natural images
- **Text embeddings**: $D = 768$ (BERT), but $d \approx 100 - 300$ for semantic structure
- **Audio**: $D = 16000$ (1 second at 16kHz), but $d \approx 20 - 100$ for speech
- **Video**: $D = 256 \times 256 \times 3 \times 30$ (256x256x3 for each frame, 30 frames), but $d \approx 50 - 200$ for natural videos

11.2.3 Why This Matters for AI

Understanding the manifold structure enables:

1. **Dimensionality reduction**: Work in d dimensions instead of D
2. **Efficient generation**: Sample from the latent space $\mathcal{Z}$ and map to $\mathcal{M}$
3. **Better generalization**: Learn the manifold structure rather than memorizing high-dimensional noise
4. **Meaningful interpolation**: Move along the manifold rather than through empty space

11.3 Statistical Manifolds and Information Geometry

Many objects in generative modeling are not points in Euclidean space but *probability distributions*. A **statistical manifold** models a parametric family of distributions $\{p(x \mid \boldsymbol{\theta})\}$ as a differentiable manifold with coordinates $\boldsymbol{\theta}$. This equips model spaces with geometry that reflects statistical distinguishability.

11.3.1 Tangent space and scores

At $\boldsymbol{\theta}$, the tangent space is spanned by the score functions (velocity of log-likelihood):

$$\frac{\partial}{\partial \theta_i} \log p(x \mid \boldsymbol{\theta}), \quad i = 1, \dots, k.$$

These directions describe how the distribution changes under infinitesimal parameter moves.

11.3.2 Fisher information metric

The canonical Riemannian metric on a statistical manifold is the **Fisher information**:

$$g_{ij}(\boldsymbol{\theta}) \;=\; \mathbb{E}_{x\sim p(\cdot|\boldsymbol{\theta})}\Big[\partial_{\theta_i} \log p \, \partial_{\theta_j} \log p\Big] \;=\; -\,\mathbb{E}\Big[\partial_{\theta_i}\partial_{\theta_j} \log p\Big].$$

It is symmetric, positive definite (under regularity), and *invariant to reparameterization*, making lengths/angles of parameter steps coordinate-independent.

11.3.3 KL divergence and local geometry

Around $\boldsymbol{\theta}$, the KL divergence admits the quadratic expansion (see also Bregman Divergence)

$$\mathrm{KL}\Big(p(\cdot \mid \boldsymbol{\theta}) \,\|\, p(\cdot \mid \boldsymbol{\theta} + d\boldsymbol{\theta})\Big) \;=\; \tfrac{1}{2}\, d\boldsymbol{\theta}^\top \mathbf{F}(\boldsymbol{\theta})\, d\boldsymbol{\theta} \;+\; o(\|d\boldsymbol{\theta}\|^2),$$

where $\mathbf{F}$ is the Fisher matrix. Thus Fisher is the *local* metric induced by KL.

11.3.4 Geodesics and affine structures (brief)

Exponential families admit special affine coordinates (mixture vs exponential). While full α-connections are beyond scope, it suffices to note: straight lines in natural parameters correspond to *exponential geodesics*, and straight lines in mean parameters correspond to *mixture geodesics*.

11.3.5 Exponential family examples

- Bernoulli(π): $\theta = \mathrm{logit}(\pi)$; Fisher $= \pi(1-\pi)$ in mean parameter, or 1 in natural parameter.
- Univariate Gaussian with mean μ and fixed variance σ^2: Fisher for μ is $1/\sigma^2$.
- Full Gaussian (mean and covariance): Fisher couples mean and covariance; natural coordinates yield block structure.

11.3.6 Optimization: natural gradient

The **natural gradient** preconditions the Euclidean gradient by $\mathbf{F}^{-1}$:

$$\tilde{\nabla}_{\boldsymbol{\theta}}\mathcal{L} \;=\; \mathbf{F}(\boldsymbol{theta})^{-1}\,\nabla_{\boldsymbol{\theta}}\mathcal{L}.$$

This is the steepest-descent direction under the Fisher metric, often improving conditioning and invariance. Practical estimates include empirical Fisher, diagonal Fisher, and K-FAC.

11.3.7 Relevance to generative models

- **VAEs**: Likelihood/decoder and encoder families live on statistical manifolds; Fisher links ELBO curvature to stable updates.
- **Normalizing flows**: Invertible maps reparameterize distributions; Fisher transforms via pushforward/pullback.
- **Diffusion**: Score $\nabla_x \log p_t$ relates to local geometry of intermediate p_t; small-step KL controls schedule design.

Key Takeaways 9

- A **statistical manifold** treats parametric distributions as a Riemannian manifold.
- The **Fisher metric** is the intrinsic metric; locally, KL equals its quadratic form.
- **Natural gradient** is steepest descent under Fisher, aiding stable and invariant learning.

11.3.8 Information Geometry Primer

KL local quadratic and Fisher. Around $\boldsymbol{\theta}$, $\mathrm{KL}\Big(p(\cdot \mid \boldsymbol{\theta}) \,\|\, p(\cdot \mid \boldsymbol{\theta} + d\boldsymbol{\theta})\Big) \approx \frac{1}{2} d\boldsymbol{\theta}^{\top}\mathbf{F}(\boldsymbol{\theta})d\boldsymbol{\theta}$. Thus Fisher is the statistical manifold's metric.

Exponential families and log-partition. For $p(x \mid \theta) = \exp\{\langle \theta, T(x)\rangle - A(\theta)\}$, the log-partition A is convex and $\nabla^2 A(\theta) = \mathbf{F}(\theta)$. Moreover, $\mathrm{KL}(\theta_1 \,\|\, \theta_2) = D_A(\theta_2 \,\|\, \theta_1)$ (Bregman of A).

Bregman duality. For a convex potential φ, $D_{\varphi}(p \,\|\, q) = D_{\varphi^*}(\nabla\varphi(q) \,\|\, \nabla\varphi(p))$. Negative entropy yields KL on the simplex; $\frac{1}{2}\|\cdot\|^2$ yields squared Euclidean.

Mirror descent vs. natural gradient. Mirror descent uses the mirror map $\nabla\varphi$ and Bregman projections; natural gradient uses a Riemannian metric $\mathbf{G}$ (e.g., Fisher) and updates $\mathbf{G}^{-1}\nabla L$. When $\mathbf{G} = \nabla^2\varphi$, the two align locally.

11.4 Fisher Information Matrix

11.4.1 Definition and equivalent forms

Definition 11.3: Fisher Information Matrix

For a model $p(x \mid \boldsymbol{\theta})$ with score $s = \nabla_{\boldsymbol{\theta}} \log p$, the *Fisher information* is $\mathbf{F}(\boldsymbol{\theta}) = \mathbb{E}[ss^\top]$ (equivalently, $-\mathbb{E}[\nabla^2_{\boldsymbol{\theta}} \log p]$ under regularity). It is the canonical metric of the statistical manifold.

Let the *score* be $s(x; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log p(x \mid \boldsymbol{\theta})$. The **Fisher information matrix** is

$$\mathbf{F}(\boldsymbol{\theta}) \;=\; \mathbb{E}_{x \sim p(\cdot \mid \boldsymbol{\theta})}\Big[\, s\, s^\top \Big] \;=\; -\,\mathbb{E}_{x \sim p(\cdot \mid \boldsymbol{\theta})}\Big[\nabla^2_{\boldsymbol{\theta}} \log p(x \mid \boldsymbol{\theta})\Big] \quad \text{(under regularity)}.$$

It quantifies local sensitivity of the distribution to parameter changes.

11.4.2 Core properties (see also Fisher Information Matrix)

- Symmetric positive semidefinite; positive definite under identifiability.
- Additive across i.i.d. samples: $\mathbf{F}_N = N\,\mathbf{F}$.
- Invariant under smooth reparameterizations.
- Cramér–Rao lower bound: for any unbiased estimator $\hat{\boldsymbol{\theta}}$, $\mathrm{Cov}(\hat{\boldsymbol{\theta}}) \succeq \mathbf{F}(\boldsymbol{\theta})^{-1}$.

11.4.3 Fisher as geometry

The Fisher matrix is the canonical Riemannian metric on a statistical manifold: locally

$$\mathrm{KL}\Big(p(\cdot \mid \boldsymbol{\theta}) \,\|\, p(\cdot \mid \boldsymbol{\theta} + d\boldsymbol{\theta})\Big) \;\approx\; \tfrac{1}{2}\, d\boldsymbol{\theta}^\top \mathbf{F}(\boldsymbol{\theta})\, d\boldsymbol{\theta},$$

so lengths and geodesics in parameter space are determined by $\mathbf{F}$.

11.4.4 Coordinate and computational views

True Fisher uses expectation over the model; the *empirical Fisher* replaces it with a data average. Mini-batch estimates with damping improve stability. Diagonal and block approximations are common for scale.

11.4.5 Relationships to other curvature notions (see also Natural Gradient, Parameter Manifolds)

When the output-space metric is Euclidean and the model is well specified, Fisher often aligns with Gauss–Newton ($\approx J^\top J$). Fisher equals the negative expected Hessian in regular exponential families, but can differ from the true Hessian in general.

11.4.6 Worked examples (see also Natural Gradient, Parameter Manifolds)

- Bernoulli/logistic regression: $\mathbf{F}$ scales with predictive variance $\pi(1-\pi)$, improving step isotropy.
- Gaussian mean (known variance σ^2): $\mathbf{F} = \sigma^{-2}\,\mathbf{I}$; with unknown variance, blocks couple mean/variance.
- Softmax regression: Fisher reflects class probabilities; label smoothing modifies curvature.

11.4.7 Practical estimation and approximations (see also Natural Gradient)

- True vs. empirical Fisher: bias/variance and computational cost trade-offs.
- Diagonal Fisher, K-FAC (Kronecker factored blocks), Shampoo: scalable curvature approximations.
- Use damping/trust-region strategies (NPG/TRPO) to control step sizes.

11.4.8 Applications in generative modeling

- VAEs: Fisher links ELBO curvature to stable, invariant updates.
- Normalizing flows: invertible reparameterizations transform Fisher by pushforward/pullback.
- Diffusion: KL-based terms exhibit local Fisher geometry along the noise schedule.

11.4.9 Key takeaways

- Fisher is the intrinsic metric of statistical manifolds via local KL geometry.
- It bounds estimator variance (CRLB) and guides geometry-aware optimization (natural gradient).
- Scalable approximations make Fisher usable in modern deep generative models.

11.5 Parameter Manifolds

11.5.1 Motivation: Why geometry in parameter space?

Training deep generative models involves optimizing parameters $\boldsymbol{\theta}$ in very high dimensions. Euclidean geometry on $\mathbb{R}^P$ is sensitive to reparameterizations and can be poorly conditioned. Treating Θ as a **parameter manifold** with an appropriate metric makes step sizes, directions, and conditioning meaningful and often invariant.

11.5.2 Definition and views (see also Fisher, Gauss–Newton)

Θ is a differentiable manifold (often $\mathbb{R}^P$ with charts). Useful geometries:

- **Euclidean**: baseline; sensitive to coordinate scalings.
- **Fisher metric on** Θ: via the model likelihood $p(x \mid \boldsymbol{\theta})$.
- **Pullback metrics**: via mappings $F(\boldsymbol{\theta})$ (e.g., outputs, features): $g_\Theta = J_F^\top g_{\text{out}} J_F$.

11.5.3 Metrics on parameter space

Fisher on Θ. $\mathbf{F}(\boldsymbol{\theta}) = \mathbb{E}\Big[\nabla_{\boldsymbol{\theta}} \log p \, \nabla_{\boldsymbol{\theta}} \log p^\top\Big]$ defines a Riemannian metric. Locally, $\mathrm{KL}(\theta, \theta + d\theta) \approx \frac{1}{2} d\theta^\top \mathbf{F} d\theta$.

Output-space pullback. If F maps parameters to outputs, then $g_\Theta = J_F^\top g_{\text{out}} J_F$. With $g_{\text{out}} = I$, this reduces to $J_F^\top J_F$ (Gauss–Newton structure).

11.5.4 Natural gradient and preconditioning

The **natural gradient** takes steepest descent under the chosen metric: $\tilde{\nabla}_{\boldsymbol{\theta}} \mathcal{L} = \mathbf{G}(\boldsymbol{\theta})^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}$ (e.g., $\mathbf{G} = \mathbf{F}$). Approximations:

- Empirical/diagonal Fisher (cheap, invariant approximations)
- K-FAC / block-diagonal curvature (layer-wise, scalable)
- Shampoo / Kronecker factorizations (improved conditioning)

11.5.5 Reparameterization and invariance

Under Euclidean metrics, simple reparameterizations (scales, normalizations) distort optimization. Riemannian metrics (Fisher/pullbacks) yield steps that are invariant to smooth reparameterizations, stabilizing training across equivalent parameterizations (e.g., logits rescaling, batchnorm, weight normalization).

11.5.6 Examples and case studies (see also Fisher, Bregman Divergence)

- **Logistic regression**: closed-form Fisher; natural gradient rescales by variance of features under the model.
- **Shallow nets**: blockwise/K-FAC approximations track layerwise curvature; improved conditioning and convergence.
- **Gauss–Newton link**: pullback metrics align with GN approximations for squared-error objectives.

11.5.7 Triangle of spaces

Parameter manifold (Θ), statistical manifold (distributions), and data/latent manifolds interact via pushforward/pullback of metrics and Jacobians; many practical "second-order" methods can be viewed as choosing a geometry on one space and pulling it back to Θ.

11.5.8 Practical guidance (see also Natural Gradient)

When to use natural gradient/approximations: ill-conditioned training, sensitivity to reparameterization, unstable step sizes. Tune damping, use minibatch Fisher, and prefer scalable factorizations for large models.

11.5.9 Key takeaways

- **Parameter manifolds** give a principled geometry for optimization.
- **Fisher/pullback metrics** produce invariant, better-conditioned steps.
- **Approximations** (diagonal/K-FAC/Shampoo) make geometry practical at scale.

11.6 Natural Gradient

11.6.1 Motivation

Euclidean gradients are coordinate-dependent and can be ill-conditioned in high-dimensional models. We seek updates that respect the geometry of the parameter manifold and are invariant to reparameterization.

11.6.2 Definition from Riemannian steepest descent

Definition 11.4: Natural Gradient

Given a Riemannian metric $\mathbf{G}(\boldsymbol{\theta})$ on parameter space, the *natural gradient* of $\mathcal{L}$ is $\tilde{\nabla}_{\boldsymbol{\theta}}\mathcal{L} = \mathbf{G}(\boldsymbol{\theta})^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}$. With $\mathbf{G} = \mathbf{F}$, this yields reparameterization-invariant updates.

Let $\mathbf{G}(\boldsymbol{\theta})$ be a Riemannian metric on the parameter manifold (e.g., Fisher or a pullback metric). The **natural gradient** is

$$\tilde{\nabla}_{\boldsymbol{\theta}}\mathcal{L} \;=\; \mathbf{G}(\boldsymbol{\theta})^{-1}\,\nabla_{\boldsymbol{\theta}}\mathcal{L},$$

the steepest descent direction under $\mathbf{G}$. Trust-region view: minimize $\mathcal{L}$ subject to a local constraint $\mathrm{KL}(\theta, \theta + d\theta) \le \varepsilon$ yields $d\theta \propto \mathbf{F}^{-1}\nabla\mathcal{L}$.

11.6.3 Choice of metric

- **Fisher metric**: from the statistical manifold; locally, $\mathrm{KL} \approx \frac{1}{2} d\theta^\top \mathbf{F}\, d\theta$.
- **Prediction/output pullback**: $\mathbf{G} = J_F^\top g_{\text{out}} J_F$; with $g_{\text{out}} = I$ this aligns with Gauss–Newton.

Choice depends on objective, likelihood modeling, and computational budget.

11.6.4 Invariance properties

Natural gradient is invariant to smooth reparameterizations of $\boldsymbol{\theta}$. Practical effects: robustness to scaling of logits, normalization layers, and alternate parameterizations, unlike Euclidean SGD with ad-hoc preconditioners.

11.6.5 Practical approximations and algorithms

- **Empirical / diagonal Fisher**: inexpensive, improves conditioning; limited coupling.
- **K-FAC / blockwise curvature**: layerwise Kronecker factorizations; scalable to large nets.
- **Shampoo / second-moment factorizations**: better conditioning with manageable cost.
- Damping, trust-region variants (e.g., NPG/TRPO), line search for stability.

11.6.6 Connections to second-order methods

Natural gradient often coincides with Gauss–Newton for squared-error models; both differ from full Newton (which uses the Hessian). These links clarify when curvature is about sensitivity to predictions vs. to parameters directly.

11.6.7 Worked examples (concise)

- **Logistic/softmax regression**: closed-form Fisher; $\tilde{\nabla}$ rescales by predictive variance, improving step isotropy.
- **Small MLP/CNN block**: K-FAC captures input/output second moments, approximating natural steps layerwise.

11.6.8 Usage guidance and caveats (see also Fisher, Parameter Manifolds)

Use when training is ill-conditioned or sensitive to reparameterization. Select approximations by model size; add damping; estimate Fisher on minibatches and maintain running averages. Watch for noisy Fisher estimates, metric–objective mismatch, and overly aggressive steps.

11.6.9 Key takeaways

- Natural gradient = geometry-aware steepest descent on parameter/statistical manifolds.
- Fisher/pullback metrics provide invariance and improved conditioning.
- Scalable approximations make it practical for modern generative models.

11.7 Bregman Divergence

11.7.1 Definition and intuition

Given a strictly convex, differentiable potential φ, the **Bregman divergence** is

$$D_\varphi(p \,\|\, q) \;=\; \varphi(p) - \varphi(q) - \langle \nabla\varphi(q),\, p - q \rangle.$$

It measures the gap between $\varphi(p)$ and the tangent plane of φ at q; asymmetric and zero iff $p = q$.

Definition 11.5: Bregman Divergence

For strictly convex differentiable φ, the *Bregman divergence* is $D_\varphi(p \,\|\, q) = \varphi(p) - \varphi(q) - \langle \nabla\varphi(q), p - q \rangle$. It is generally asymmetric and not a metric.

11.7.2 Core properties

Nonnegative and convex in the first argument; not a metric (asymmetry, no triangle inequality). **Duality:** with the Legendre transform φ^*,

$$D_\varphi(p \,\|\, q) = D_{\varphi^*}(\nabla\varphi(q) \,\|\, \nabla\varphi(p)).$$

Projections: Bregman projections minimize $D_\varphi(\cdot \,\|\, \mathcal{C})$ and satisfy a generalized Pythagorean theorem.

11.7.3 Important examples

- **Squared Euclidean:** $\varphi(x) = \frac{1}{2}\|x\|^2 \Rightarrow D = \frac{1}{2}\|p - q\|^2$.
- **KL on the simplex:** $\varphi(p) = \sum_i p_i \log p_i \Rightarrow D = \sum_i p_i \log \frac{p_i}{q_i}$.
- **Itakura–Saito:** $\varphi(x) = -\log x$ on $x > 0$.
- Logistic and generalized I-divergences arise from suitable φ.

11.7.4 Exponential families and Fisher geometry

For exponential families with natural parameter θ and log-partition $A(\theta)$,

$$\mathrm{KL}\Big(p(\cdot \mid \theta_1) \,\|\, p(\cdot \mid \theta_2)\Big) = D_A(\theta_2 \,\|\, \theta_1).$$

The local quadratic expansion of KL yields Fisher, linking Bregman structure to statistical manifolds and the geometry used by natural gradients.

11.7.5 Optimization links (see also Natural Gradient, Parameter Manifolds)

Mirror descent performs gradient steps in the dual space with mirror map $\nabla\varphi$ and updates via Bregman projection. Natural gradient aligns locally when the metric $\mathbf{G}$ matches $\nabla^2\varphi$. Euclidean GD is mirror descent with $\varphi = \frac{1}{2}\|\cdot\|^2$.

11.7.6 Projections, averaging, and means

The Bregman centroid minimizes average divergence and recovers familiar means: arithmetic (Euclidean), entropy-induced means on the simplex (geometric/softmax-like). Used in Bregman k-means and information-theoretic clustering.

11.7.7 Applications in generative modeling

- **Loss design:** choose φ to match output domains (simplex, positive reals).
- **Variational objectives:** ELBO's KL is Bregman; affects regularization and curvature.
- **Flows/diffusion:** divergence choices shape training dynamics and consistency terms.

11.7.8 Practical guidance (see also Fisher, Natural Gradient)

Pick φ by domain and invariances; mind asymmetry ($D_\varphi(p \,\|\, q)$ vs $D_\varphi(q \,\|\, p)$). Ensure numerical stability near boundaries (e.g., probabilities), and use appropriate smoothing/-damping.

11.7.9 Key takeaways

- Bregman divergences unify many losses via convex potentials.
- KL is a Bregman divergence; ties to exponential families and Fisher geometry.
- Mirror descent and natural gradient are complementary geometry-aware optimization views.

11.8 Manifolds in Variational Autoencoders (VAEs)

11.8.1 Architecture Overview

Variational Autoencoders (VAEs) explicitly model the data manifold structure. The key insight is that the encoder and decoder act as **charts** and **inverse charts** between the data manifold and a flat latent space.

11.8.2 Mathematical Formulation

A VAE consists of:

1. **Encoder** $q_\phi(\mathbf{z}|\mathbf{x})$: Approximates the posterior distribution over latent codes given data
2. **Decoder** $p_\theta(\mathbf{x}|\mathbf{z})$: Generates data from latent codes
3. **Latent prior** $p(\mathbf{z})$: Typically $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in $\mathbb{R}^d$

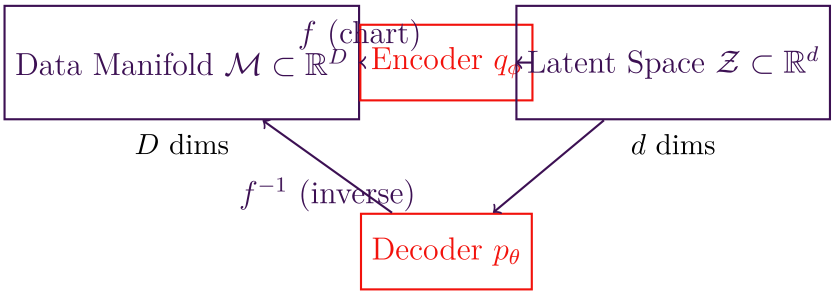

Data Manifold $\mathcal{M} \subset \mathbb{R}^D$
f (chart)
Encoder q_ϕ
Latent Space $\mathcal{Z} \subset \mathbb{R}^d$
D dims
d dims
f^{-1} (inverse)
Decoder p_θ

Figure 11.2: VAE architecture: The encoder maps data from the manifold to a flat latent space (chart), and the decoder maps back (inverse chart). This is exactly the manifold definition from Chapter 1!

The encoder acts as a **chart** mapping the data manifold to the latent space:

$$q_\phi : \mathcal{M} \to \mathcal{Z}, \quad \mathbf{x} \mapsto \mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon} \tag{11.2}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is element-wise multiplication.

The decoder acts as an **inverse chart** mapping the latent space back to the manifold:

$$p_\theta : \mathcal{Z} \to \mathcal{M}, \quad \mathbf{z} \mapsto \mathbf{x} = f_\theta(\mathbf{z}) \tag{11.3}$$

where f_θ is a neural network.

11.8.3 The Variational Objective

The VAE optimizes the **Evidence Lower BOund (ELBO)**:

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \tag{11.4}$$

Breaking this down:

- **Reconstruction term** $\mathbb{E}[\log p_\theta(\mathbf{x}|\mathbf{z})]$: Encourages the decoder to accurately map latent codes back to data (preserves manifold structure)
- **Regularization term** $\text{KL}(q_\phi \| p)$: Encourages the latent distribution to match the prior (ensures the latent space is well-structured)

11.8.4 Connection to Manifold Theory

The VAE's encoder-decoder structure directly implements the manifold definition:

- **Local flatness**: The latent space $\mathcal{Z}$ is flat (Euclidean), and the decoder maps it to the curved data manifold
- **Charts**: Each encoder q_ϕ defines a chart from a neighborhood on $\mathcal{M}$ to $\mathcal{Z}$
- **Atlas**: Multiple encoders (or stochastic samples) create an atlas covering the manifold
- **Dimension reduction**: The latent dimension d captures the intrinsic dimension of the data manifold

11.9 Manifolds in Generative Adversarial Networks (GANs)

11.9.1 The Generator as a Manifold

In GANs, the generator network $G : \mathcal{Z} \to \mathbb{R}^D$ implicitly learns the data manifold. The generator maps from a latent space (typically uniform or Gaussian) directly to the data space:

$$\mathbf{x} = G(\mathbf{z}), \quad \mathbf{z} \sim p(\mathbf{z}) \tag{11.5}$$

The set $\{G(\mathbf{z}) : \mathbf{z} \in \mathcal{Z}\}$ forms the learned manifold $\mathcal{M}_G$.

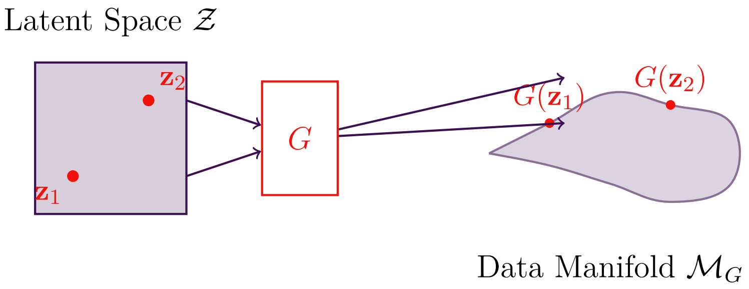


Figure 11.3: GAN generator as a manifold: The generator G maps the latent space to the data manifold $\mathcal{M}_G = \{G(\mathbf{z}) : \mathbf{z} \in \mathcal{Z}\}$.

11.9.2 Adversarial Training Objective

The GAN training objective is:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{11.6}$$

The discriminator D learns to distinguish real data from generated data. This adversarial process forces the generator to learn the true data manifold structure.

11.9.3 Manifold Geometry in GANs

The generator G defines a **parameterized manifold**:

$$\mathcal{M}_G = \{\mathbf{x} \in \mathbb{R}^D : \mathbf{x} = G(\mathbf{z}), \mathbf{z} \in \mathcal{Z}\} \tag{11.7}$$

The **Jacobian** of the generator:

$$\mathbf{J}_G(\mathbf{z}) = \frac{\partial G}{\partial \mathbf{z}} \in \mathbb{R}^{D\times d} \tag{11.8}$$

defines the tangent space at each point on the manifold. The columns of $\mathbf{J}_G$ span the tangent space.

11.9.4 Issues with GAN Manifolds

GANs can suffer from:

- **Mode collapse**: The generator only covers a subset of the true manifold
- **Non-smooth manifolds**: The learned manifold may have discontinuities or kinks
- **Off-manifold generation**: Generated samples may lie slightly off the true data manifold

These issues relate to the manifold structure not being properly learned or regularized.

11.10 Manifolds in Diffusion Models

11.10.1 The Forward and Reverse Processes

Diffusion models learn to generate data by reversing a diffusion process that gradually adds noise. This process moves data **off** the manifold and then learns to bring it back **onto** the manifold.

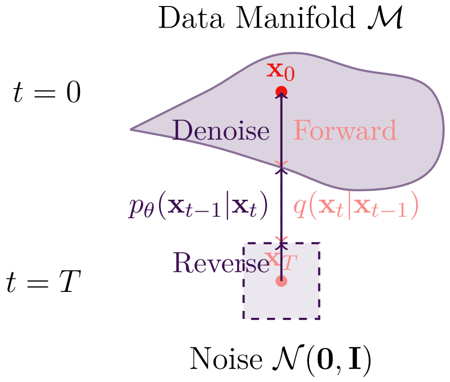

Data Manifold $\mathcal{M}$
$t = 0$
$\mathbf{x}_0$
Denoise
Forward
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
Reverse
$\mathbf{x}_T$
$t = T$
Noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$

Figure 11.4: Diffusion process: Forward process moves data off the manifold to noise, reverse process learns to bring it back onto the manifold.

11.10.2 Mathematical Formulation

The forward diffusion process is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{11.9}$$

where β_t is a noise schedule. This gradually moves data off the manifold.

After T steps:

$$q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_T; \sqrt{\bar{\alpha}_T}\mathbf{x}_0, (1-\bar{\alpha}_T)\mathbf{I}) \tag{11.10}$$

where $\bar{\alpha}_T = \prod_{s=1}^{T}(1-\beta_s)$. For large T, $\mathbf{x}_T \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$ (pure noise, off-manifold).

The reverse process learns to denoise:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{11.11}$$

The model $\boldsymbol{\mu}_\theta$ learns to move noisy samples back onto the data manifold.

11.10.3 Manifold Structure in Diffusion

The diffusion model implicitly learns the data manifold because:

- **Forward process**: $\mathbf{x}_0 \in \mathcal{M} \to \mathbf{x}_T \notin \mathcal{M}$ (off-manifold)
- **Reverse process**: $\mathbf{x}_T \notin \mathcal{M} \to \mathbf{x}_0 \in \mathcal{M}$ (onto-manifold)
- The learned denoising function $\boldsymbol{\mu}_\theta$ essentially defines a vector field that points toward the manifold

The manifold is the **attractor** of the reverse diffusion process.

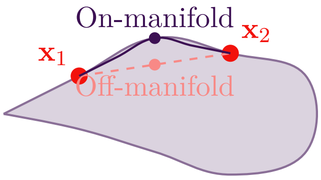
11.11 Geodesics in Generative AI

11.11.1 The Interpolation Problem

A fundamental task in generative AI is **interpolation**: given two data points $\mathbf{x}_1$ and $\mathbf{x}_2$, generate a smooth sequence of intermediate points. Naive linear interpolation in the data space often fails:

$$\mathbf{x}(t) = (1-t)\mathbf{x}_1 + t\mathbf{x}_2, \quad t \in [0,1] \tag{11.12}$$

This linear path may leave the data manifold, resulting in unrealistic or invalid samples.



Linear interpolation (dashed) vs. geodesic (solid)

Figure 11.5: Linear interpolation in data space can leave the manifold, while geodesic interpolation stays on the manifold.

11.11.2 Geodesic Interpolation in Latent Space

In generative models, we can interpolate in the **latent space** and then map to the data space:

$$\mathbf{z}(t) = (1-t)\mathbf{z}_1 + t\mathbf{z}_2, \quad \mathbf{x}(t) = G(\mathbf{z}(t)) \tag{11.13}$$

However, even linear interpolation in latent space may not correspond to geodesics on the data manifold. True geodesic interpolation requires solving the geodesic equation on the learned manifold.

11.11.3 Computing Geodesics on Learned Manifolds

Given a generator $G : \mathcal{Z} \to \mathcal{M}$, the geodesic between $\mathbf{x}_1 = G(\mathbf{z}_1)$ and $\mathbf{x}_2 = G(\mathbf{z}_2)$ can be found by:

1. Finding the geodesic $\gamma(t)$ in latent space that minimizes:

$$L[\gamma] = \int_0^1 \sqrt{\mathbf{g}_{ij}(\gamma(t))\dot{\gamma}^i(t)\dot{\gamma}^j(t)}\, dt \tag{11.14}$$

   where $\mathbf{g}_{ij}$ is the **pullback metric**:

$$\mathbf{g}_{ij}(\mathbf{z}) = \sum_{k=1}^{D} \frac{\partial G_k}{\partial z^i}\frac{\partial G_k}{\partial z^j} = (\mathbf{J}_G^T \mathbf{J}_G)_{ij} \tag{11.15}$$

2. Mapping to data space: $\mathbf{x}(t) = G(\gamma(t))$

This gives a geodesic on the data manifold that stays on the manifold throughout.

11.11.4 Applications of Geodesics

Geodesic interpolation is used for:

- **Style transfer**: Smooth transitions between styles
- **Attribute editing**: Changing facial features, object properties
- **Data augmentation**: Generating realistic variations
- **Exploration**: Understanding the manifold structure

11.12 Distance Metrics on Learned Manifolds

11.12.1 Learned vs. Euclidean Distance

In the ambient space $\mathbb{R}^D$, we have Euclidean distance:

$$d_{\text{Euclidean}}(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \tag{11.16}$$

But on the data manifold $\mathcal{M}$, the true distance is the geodesic distance:

$$d_{\mathcal{M}}(\mathbf{x}_1, \mathbf{x}_2) = \inf_{\gamma} \int_0^1 \sqrt{\mathbf{g}_{ij}(\gamma(t))\dot{\gamma}^i(t)\dot{\gamma}^j(t)}\, dt \tag{11.17}$$

where the infimum is over all paths γ connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ on $\mathcal{M}$.

11.12.2 Perceptual Distance

In practice, **perceptual distance** (how humans perceive similarity) often aligns better with geodesic distance on the data manifold than with Euclidean distance. This is why:

- Images that are geodesically close look similar to humans
- Euclidean distance can be misleading (e.g., small pixel changes can create very different images)
- Generative models that respect manifold structure produce more realistic results

11.12.3 Latent Space Distance

In the latent space of a VAE or GAN, we can approximate manifold distance:

$$d_{\text{latent}}(\mathbf{x}_1, \mathbf{x}_2) \approx \|\mathbf{z}_1 - \mathbf{z}_2\|_2 \tag{11.18}$$

where $\mathbf{z}_i$ are the latent codes. This is exact if the latent space is isometric to the manifold, but approximate in general.

11.13 Manifold Learning Algorithms

11.13.1 Principal Component Analysis (PCA)

PCA finds a linear subspace (flat manifold) that best approximates the data:

$$\mathbf{x} \approx \mathbf{U}_d\mathbf{z} + \boldsymbol{\mu} \tag{11.19}$$

where $\mathbf{U}_d \in \mathbb{R}^{D\times d}$ contains the top d principal components, $\mathbf{z} \in \mathbb{R}^d$ are the coordinates, and $\boldsymbol{\mu}$ is the mean.

This is a **linear manifold**—it works well when the true manifold is approximately flat.

11.13.2 Isomap

Isomap learns a nonlinear manifold by:

1. Building a graph of nearest neighbors
2. Computing shortest paths (approximate geodesics) on the graph
3. Embedding into lower dimensions preserving these geodesic distances

This directly uses the geodesic distance concept from Chapter 5!

11.13.3 Local Linear Embedding (LLE)

LLE assumes the manifold is locally linear. Each point is reconstructed as a linear combination of its neighbors:

$$\min_{\mathbf{W}} \sum_i \left\| \mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j \right\|^2 \tag{11.20}$$

subject to constraints. This preserves local manifold structure.

11.13.4 Neural Manifold Learning

Modern deep learning approaches learn manifolds implicitly:

- **Autoencoders**: Learn encoder-decoder mappings (like VAEs)
- **Generative models**: Learn data distributions on manifolds
- **Contrastive learning**: Learn representations that respect manifold structure

11.14 Advanced Topics: Curvature and Topology

11.14.1 Curvature of Learned Manifolds

The **Riemannian curvature tensor** measures how the manifold curves. In learned manifolds:

- **Positive curvature**: Data clusters, geodesics converge (like a sphere)
- **Negative curvature**: Data spreads out, geodesics diverge (like a saddle)
- **Zero curvature**: Flat regions (like a plane)

The curvature affects:

- How interpolation behaves
- The complexity needed to represent the manifold
- Generalization properties of generative models

11.14.2 Topological Properties

The topology of the data manifold (e.g., number of holes, connected components) affects generative models:

- **Disconnected manifolds**: Multiple modes, requires special handling
- **Non-orientable manifolds**: Rare but possible in some data
- **High genus**: Many holes, complex structure

Understanding topology helps design better generative models.

11.15 Challenges and Future Directions

11.15.1 Current Challenges

Estimating Intrinsic Dimension

Determining the true intrinsic dimension d of the data manifold is difficult:

- Methods: Correlation dimension, nearest neighbor methods, neural network approaches
- The dimension may vary across different regions of the manifold
- High-dimensional data makes estimation challenging

Non-Uniform Manifolds

Real data manifolds are often:

- **Non-uniform**: Different regions have different densities
- **Non-smooth**: May have discontinuities or sharp transitions
- **Multi-scale**: Different levels of detail at different scales

This complicates learning and generation.

Geodesic Computation in High Dimensions

Computing exact geodesics on high-dimensional learned manifolds is computationally expensive:

- The geodesic equation requires solving a system of ODEs
- Numerical methods are needed for most cases
- Approximations trade off accuracy for speed

11.15.2 Future Research Directions

Geometric Deep Learning

Incorporating explicit geometric structure into neural networks:

- **Riemannian neural networks**: Operations that respect manifold geometry
- **Geometric attention**: Attention mechanisms on manifolds
- **Manifold regularization**: Explicit constraints on learned representations

Discrete Manifolds

Extending beyond smooth manifolds:

- **Graph manifolds**: Discrete structures with manifold-like properties
- **Hybrid models**: Combining discrete and continuous representations
- **Singularities**: Handling points where the manifold structure breaks down

Manifold Alignment

Aligning manifolds across domains:

- **Cross-modal learning**: Aligning image and text manifolds
- **Transfer learning**: Transferring manifold structure across tasks
- **Multi-manifold learning**: Learning multiple related manifolds

Theoretical Foundations

Better understanding of:

- **Manifold capacity**: How much data can a manifold represent?
- **Generalization bounds**: How does manifold structure affect learning?
- **Convergence properties**: Do generative models converge to the true manifold?

Key Takeaways 10

This chapter has shown how the mathematical concepts from earlier chapters directly apply to generative AI:

1. **Manifolds**: Data lies on low-dimensional manifolds embedded in high-dimensional spaces. Generative models learn to represent and sample from these manifolds.
2. **Charts and Atlases**: VAEs explicitly use encoder-decoder pairs as charts mapping between the data manifold and flat latent spaces.
3. **Geodesics**: The shortest paths on manifolds provide natural interpolation methods that stay on the manifold, crucial for realistic generation.
4. **Distances**: Geodesic distances on manifolds align better with perceptual similarity than Euclidean distances, explaining why manifold-aware models perform better.
5. **Local Structure**: Open n-balls and local neighborhoods enable efficient computation and learning, even on complex curved manifolds.
6. **Geometry Matters**: Understanding the curvature, topology, and structure of learned manifolds helps design better generative models.

11.16 Conclusion: The Geometric Foundation of AI

Throughout this book, we've journeyed from the basic definition of a manifold to understanding geodesics, distances, and local structure. We've seen how these abstract mathematical concepts describe real-world phenomena—from navigation on Earth to the structure of data in AI systems.

The connection between manifolds and generative AI is profound:

- **Data has structure**: Natural data lies on manifolds, not randomly in high-dimensional space
- **Geometry guides generation**: Understanding manifold geometry enables better generative models
- **Mathematical tools apply**: Concepts from differential geometry directly solve AI problems

- **Intuition matters**: Geometric intuition helps design and understand AI systems

As generative AI continues to advance, a deep understanding of manifolds, geodesics, and geometric structure will become increasingly important. The mathematical foundations we've built here provide the tools needed to understand, analyze, and improve the next generation of AI systems.

The journey from abstract mathematics to practical AI is not just possible—it's essential. The manifold structure of data is not a mathematical curiosity; it's a fundamental property that shapes how we build and understand artificial intelligence.

Appendix A

Summary Table

A comprehensive summary table comparing key properties, formulas, and characteristics across different manifolds and concepts explored in this book.

A.1 Manifold Comparison

Manifold	Dimension	Curvature	Type	Key Property
1D Line	1	Zero	Open	Infinitely extendable
1D Circle	1	Positive	Closed	Loops back on itself
2D Plane	2	Zero	Open	Flat everywhere
2D Sphere	2	Positive	Closed	Constant positive curvature
2D Torus	2	Mixed	Closed	Positive and negative regions

Table A.1: Comparison of basic manifold properties.

A.2 Distance Formulas

Manifold	Distance Formula
1D Line	$d = \|x_2 - x_1\|$
1D Circle	$d = r \cdot \theta$
2D Plane	$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
2D Sphere	$d = R \cdot \arccos(\vec{v}_1 \cdot \vec{v}_2)$
2D Sphere (coords)	$d = R \cdot \arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\Delta\lambda))$
2D Sphere (haversine)	$d = 2R \cdot \arcsin\left(\sqrt{\sin^2(\Delta\phi/2) + \cos\phi_1 \cos\phi_2 \sin^2(\Delta\lambda/2)}\right)$
2D Torus	$d \approx \sqrt{(R + r\cos\theta)^2(\Delta\phi)^2 + r^2(\Delta\theta)^2}$

Table A.2: Distance formulas for different manifolds.

Manifold	Geodesic Type	Uniqueness	Completeness	Construction
1D Line	Straight line	Always unique	Complete	Direct
1D Circle	Arc	Unique (except antipodal)	Complete	Direct
2D Plane	Straight line	Always unique	Complete	Direct
2D Sphere	Great circle arc	Unique shortest	Complete	Plane intersection
2D Torus	Complex paths	Multiple possible	Complete	Numerical methods

Table A.3: Comparison of geodesic properties across different manifolds.

A.3 Geodesic Properties

A.4 Geodesic Equations and Methods

Manifold	Geodesic Equation	Solution Method
1D Line	$\ddot{x} = 0$	Direct integration
1D Circle	$\ddot{\theta} = 0$	Direct integration
2D Plane	$\ddot{x} = 0$, $\ddot{y} = 0$	Direct integration
2D Sphere	$\ddot{\phi} - \sin\phi\cos\phi\dot{\lambda}^2 = 0$	Geometric construction
	$\ddot{\lambda} + 2\cot\phi\dot{\phi}\dot{\lambda} = 0$	or numerical methods

Table A.4: Geodesic equations and solution methods.

A.5 Open n-Ball Properties

Dimension	Name	Definition
$n = 1$	Open interval	$B_r(p) = \{x : \|x - p\| < r\}$
$n = 2$	Open disk	$B_r(\mathbf{p}) = \{(x, y) : (x - p_x)^2 + (y - p_y)^2 < r^2\}$
$n = 3$	Open ball	$B_r(\mathbf{p}) = \{(x, y, z) : (x - p_x)^2 + (y - p_y)^2 + (z - p_z)^2 < r^2\}$
n	Open n-ball	$B_r(\mathbf{p}) = \{\mathbf{x} : \\|\mathbf{x} - \mathbf{p}\\| < r\}$

Table A.5: Open n-balls in different dimensions.

A.6 Generative AI Models and Manifolds

Model	Manifold Connection	Key Component
VAE	Encoder/decoder as charts	Maps data ↔ latent space
GAN	Generator defines manifold	$\mathcal{M}_G = \{G(\mathbf{z}) : \mathbf{z} \in \mathcal{Z}\}$
Diffusion	Forward/reverse on manifold	Data → noise → data
Normalizing Flow	Manifold-to-manifold maps	Preserves volume/structure

Table A.6: How generative models relate to manifolds.

A.7 Curvature Effects

Curvature Type	Geodesic Behavior	Example	Visual
Zero	Parallel geodesics stay parallel	Plane, line	Parallel lines
Positive	Geodesics converge	Sphere	Meridians meet at poles
Negative	Geodesics diverge	Saddle	Hyperbolic geometry
Mixed	Varies by region	Torus	Combination

Table A.7: How curvature affects geodesic behavior.

A.8 Special Cases: Sphere Distance

Special Case	Simplified Formula
Same meridian	$d = R \cdot \lvert\phi_2 - \phi_1\rvert$
Same parallel	$d = R \cdot \cos\phi \cdot \lvert\Delta\lambda\rvert$
Near poles	Use haversine formula
Antipodal points	$d = \pi R$ (half circumference)
Very small distance	Flat Earth: $d \approx R\sqrt{(\Delta\phi)^2 + (\cos\bar{\phi})^2(\Delta\lambda)^2}$

Table A.8: Special cases for sphere distance calculations.

A.9 Unifying Principles

All geodesics share these fundamental properties:

1. **Minimization**: They minimize path length (or are critical points of the length functional)
2. **Differential equation**: They satisfy $\ddot{x}^i + \Gamma^i_{jk}\dot{x}^j\dot{x}^k = 0$
3. **Local straightness**: Zero geodesic curvature—"as straight as possible" on the manifold
4. **Coordinate independence**: Geometric objects, independent of coordinate choices
5. **Intrinsic nature**: Depend only on the metric (intrinsic geometry), not on embeddings

A.10 Key Relationships

- **Distance = Geodesic Length**: The distance between two points equals the length of the geodesic connecting them
- **Manifold = Locally Open n-Ball**: Every point has a neighborhood homeomorphic to an open n-ball

- **Chart = Local Coordinate System**: Charts provide mappings from manifolds to flat spaces
- **Geodesic = Minimized Path Length**: Geodesics minimize the path length functional
- **Curvature = Geodesic Convergence**: Positive curvature causes convergence, negative causes divergence

Appendix B

Metaphors and Intuition

This appendix collects useful metaphors and intuitive explanations from throughout the book to help build geometric intuition. These analogies make abstract mathematical concepts more concrete and accessible.

B.1 Manifolds

B.1.1 The Earth Analogy

The metaphor: Imagine standing on Earth's surface. To you, the ground appears flat—you can use a local map as if you were on a plane. Yet we know Earth is a sphere globally. This intuition—that curved spaces look flat when you zoom in—is the essence of a manifold.

Why it helps: This is the most fundamental metaphor for manifolds. It captures the key property: **local flatness with global curvature**. Just as a small patch of Earth looks like a flat plane, any small neighborhood on a manifold looks like Euclidean space.

B.1.2 The Bug on a Donut

The metaphor: Like a small bug walking on a giant, oddly shaped donut—to the bug, the surface feels flat locally, but really it loops back and curves globally.

Why it helps: Emphasizes that manifolds can have complex global structure (like a torus) while still being locally flat. The bug's perspective is local, but the donut's shape is global.

B.1.3 Zooming In

The metaphor: If you zoom in very closely on any small neighborhood in a manifold, it looks like regular n-dimensional space. Think of zooming in on a curved surface until it looks flat.

Why it helps: Captures the mathematical definition: manifolds are spaces that are locally homeomorphic to Euclidean space. The "zooming in" action represents taking a small neighborhood.

B.2 Charts and Atlases

B.2.1 The Map Analogy

The metaphor: Just as an atlas of Earth contains multiple maps covering different regions, a mathematical atlas contains multiple charts covering different neighborhoods of the manifold. Each chart (map) shows a local view that looks flat.

Why it helps: Charts are like maps—they provide local coordinate systems. The atlas analogy explains why we need multiple charts to cover a whole manifold, just like we need multiple maps to cover the entire Earth.

B.2.2 Local Coordinate Systems

The metaphor: Charts allow us to work with the manifold locally as if it were flat, just like a street map lets you navigate a city as if it were a flat grid.

Why it helps: Explains the practical purpose of charts—they let us use familiar flat-space mathematics (calculus, geometry) locally on curved manifolds.

B.3 Geodesics

B.3.1 The String Analogy

The metaphor: The best way to understand geodesics is to imagine pulling a string tight between two points on a surface:

- The string naturally takes the shortest path
- It stays on the surface (can't cut through it)
- It's under tension, which minimizes its length
- The path it takes is the geodesic!

Why it helps: This physical intuition is perhaps the most powerful metaphor for geodesics. It captures all key properties: shortest path, staying on the surface, and the minimization principle.

B.3.2 The Tightrope Analogy

The metaphor: Like the path of a tightrope between two points on a curved dome, always "the shortest rope" route that sticks to the surface.

Why it helps: Emphasizes that geodesics must stay on the surface (like a tightrope on a dome) rather than cutting through space.

B.3.3 Great Circle Routes

The metaphor: Airplanes follow great circle routes—the shortest paths on Earth's surface. These are geodesics: they follow the curvature of Earth rather than appearing as straight lines on a flat map.

Why it helps: Provides a real-world example that readers can relate to. Explains why flight paths look curved on flat maps but are actually the shortest routes.

B.3.4 Generalization of Straight Lines

The metaphor: Geodesics are the natural generalization of "straight lines" to curved spaces. On a flat plane, geodesics are straight lines. On curved surfaces, they're the "straightest possible" paths while staying on the surface.

Why it helps: Connects the familiar (straight lines) to the new (geodesics), showing that geodesics are the natural extension of straightness to curved spaces.

B.4 Distance

B.4.1 Flight Routes vs. Tunnels

The metaphor: Distance on a sphere is like the distance between two cities on Earth measured by flight routes that follow arcs, not straight tunnels through Earth.

Why it helps: Clarifies that we measure distance *along* the surface, not through space. A flight route follows the surface; a tunnel would cut through it.

B.4.2 Arc Length vs. Chord Distance

The metaphor: On a circle, the arc distance (following the curve) is always greater than or equal to the chord distance (straight line through space). Like walking around the edge of a circle versus cutting straight across.

Why it helps: Explains why geodesic distance (arc length) differs from Euclidean distance (chord) on curved manifolds.

B.5 Open n-Balls

B.5.1 Neighborhoods Without Boundaries

The metaphor: An open n-ball is like a neighborhood that doesn't include its boundary—think of a city that extends right up to but doesn't include the city limits. This "openness" is essential for manifolds.

Why it helps: Explains why we use "open" balls (excluding boundaries) rather than "closed" balls. The boundary exclusion ensures smooth local structure.

B.5.2 Zooming In Reveals a Disk

The metaphor: When you zoom in on a manifold, you reveal an open disk (2D ball). This is what "locally looks like Euclidean space" means precisely.

Why it helps: Connects the zooming-in metaphor to the mathematical definition using open n-balls.

B.6 Curvature

B.6.1 Parallel Lines Behavior

The metaphor:

- **Zero curvature** (flat): Parallel lines stay parallel forever—like train tracks on flat ground
- **Positive curvature** (sphere): Parallel lines converge—like lines of longitude meeting at the poles
- **Negative curvature** (saddle): Parallel lines diverge—like paths spreading apart on a saddle

Why it helps: Provides intuitive ways to understand different types of curvature by visualizing how parallel geodesics behave.

B.7 Manifolds in AI

B.7.1 The Twisted Sheet

The metaphor: Imagine a twisted 2D sheet bent inside 3D space (a manifold). Learning the manifold means understanding the shape and rules of this sheet so the AI can generate or manipulate data along it sensibly, rather than arbitrarily in the whole 3D space.

Why it helps: Explains the data manifold hypothesis—data lies on a low-dimensional structure embedded in high-dimensional space. The AI needs to learn this structure.

B.7.2 The Data Manifold

The metaphor: High-dimensional data like images is like a 65,536-dimensional space where most points are random noise. The set of all possible natural images forms a much smaller subset—a manifold embedded in this space. Like a 2D surface floating in 3D space.

Why it helps: Makes concrete the idea that data has structure—not all points in high-dimensional space are valid. The manifold represents the valid data.

B.7.3 VAEs as Maps

The metaphor: In a VAE, the encoder and decoder act like maps (charts). The encoder is like creating a flat map of a curved region, and the decoder is like using that map to navigate back to the curved space.

Why it helps: Connects the abstract concept of charts (from Chapter 1) to the concrete architecture of VAEs, showing how mathematical concepts directly apply to AI.

B.7.4 Geodesic Interpolation

The metaphor: Linear interpolation in data space is like drawing a straight line between two cities—it might go through the Earth! Geodesic interpolation is like following the flight route—it stays on the surface.

Why it helps: Explains why geodesic interpolation produces more realistic results in generative AI—it respects the manifold structure.

B.8 General Principles

B.8.1 Local vs. Global

The metaphor: Manifolds are like neighborhoods that look flat locally but curve globally. Like your local area feeling flat, but Earth being a sphere.

Why it helps: Emphasizes the key distinction between local properties (flatness) and global properties (curvature).

B.8.2 Intrinsic vs. Extrinsic

The metaphor: Intrinsic geometry is like measurements made by someone living on the surface (like a bug on a sphere). Extrinsic geometry is like measurements made from outside (like viewing a sphere in 3D space). Geodesics are intrinsic—they depend only on the surface itself.

Why it helps: Clarifies that geodesics are properties of the manifold itself, not of how it's embedded in higher-dimensional space.

B.9 Why These Metaphors Matter

These metaphors serve several purposes:

- **Building intuition**: They make abstract concepts concrete and visualizable
- **Bridging concepts**: They connect familiar ideas to new mathematical concepts
- **Aiding memory**: Memorable analogies help recall definitions and properties
- **Enabling insight**: Visual metaphors help understand why theorems and formulas work

As you progress through the book, return to these metaphors when concepts feel abstract. They provide the intuitive foundation that makes the mathematical rigor accessible.

References

This bibliography provides a comprehensive list of references covering the mathematical foundations of manifolds and geodesics, as well as their applications in generative artificial intelligence. The references are organized by topic to help readers find relevant materials for deeper study.

How to Use This Bibliography

The references cover several key areas:

- **Differential Geometry and Manifolds**: Foundational textbooks and introductions to smooth manifolds, Riemannian geometry, and differential topology
- **Geodesics and Distance Calculations**: Methods for computing geodesics and distances on curved surfaces, including spherical geometry
- **Manifold Learning**: Algorithms and theoretical foundations for discovering low-dimensional structure in high-dimensional data
- **Generative AI Models**: Original papers and key developments in VAEs, GANs, diffusion models, and normalizing flows
- **Manifolds in Machine Learning**: Research connecting geometric foundations to modern deep learning

Each entry includes a brief note describing its relevance to the topics covered in this book.

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. *arXiv preprint*, 2017. WGAN using Wasserstein distance for improved training stability. URL: https://arxiv.org/abs/1701.07875, arXiv:1701.07875, doi:10.48550/arXiv.1701.07875.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. Laplacian eigenmaps for manifold learning. doi:10.1162/089976603321780317.

[3] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velicjkovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint*, 2021. Comprehensive survey on geometric deep learning. URL: https://arxiv.org/abs/2104.13478, arXiv:2104.13478, doi:10.48550/arXiv.2104.13478.

[4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. Early survey on geometric deep learning. doi:10.1109/MSP.2017.2693418.

[5] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. *arXiv preprint*, 2018. Neural ODEs for continuous normalizing flows. URL: https://arxiv.org/abs/1806.07366, arXiv:1806.07366, doi:10.48550/arXiv.1806.07366.

[6] Manfredo P. do Carmo. *Riemannian Geometry*. Birkhäuser, 1st edition, 1992. Classic textbook on Riemannian geometry, including geodesics and curvature. doi:10.1007/978-1-4757-2201-7.

[7] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016. Rigorous mathematical treatment of the manifold hypothesis in data. doi:10.1090/jams/852.

[8] Mevlana C. Gemici, Danilo Rezende, and Shakir Mohamed. Normalizing flows on riemannian manifolds. *arXiv preprint*, 2016. Extends normalizing flows to Riemannian manifolds. URL: https://arxiv.org/abs/1611.02304, arXiv:1611.02304, doi:10.48550/arXiv.1611.02304.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *arXiv preprint*, 2014. Original GAN paper introducing the adversarial training

framework. URL: https://arxiv.org/abs/1406.2661, arXiv:1406.2661, doi:10.48550/arXiv.1406.2661.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint*, 2020. Foundational paper on diffusion models for image generation. URL: https://arxiv.org/abs/2006.11239, arXiv:2006.11239, doi:10.48550/arXiv.2006.11239.

[11] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint*, 2018. GLOW architecture for normalizing flows. URL: https://arxiv.org/abs/1807.03039, arXiv:1807.03039, doi:10.48550/arXiv.1807.03039.

[12] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, 2013. Foundational paper on Variational Autoencoders. URL: https://arxiv.org/abs/1312.6114, arXiv:1312.6114, doi:10.48550/arXiv.1312.6114.

[13] John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 2012. Comprehensive introduction to smooth manifolds and differential geometry. doi:10.1007/978-1-4419-9982-5.

[14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint*, 2015. DCGAN architecture for image generation. URL: https://arxiv.org/abs/1511.06434, arXiv:1511.06434, doi:10.48550/arXiv.1511.06434.

[15] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint*, 2015. Introduces normalizing flows for variational inference. URL: https://arxiv.org/abs/1505.05770, arXiv:1505.05770, doi:10.48550/arXiv.1505.05770.

[16] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint*, 2014. Alternative derivation of VAEs with reparameterization trick. URL: https://arxiv.org/abs/1401.4082, arXiv:1401.4082, doi:10.48550/arXiv.1401.4082.

[17] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. Presents LLE algorithm for manifold learning. doi:10.1126/science.290.5500.2323.

[18] Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The riemannian geometry of deep generative models. *arXiv preprint*, 2018. Explores Riemannian geometry in the context of deep generative models. URL: https://arxiv.org/abs/1711.08014, arXiv:1711.08014, doi:10.48550/arXiv.1711.08014.

[19] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint*, 2021. DDIM for faster sampling from diffusion models. URL: https://arxiv.org/abs/2010.02502, arXiv:2010.02502, doi:10.48550/arXiv.2010.02502.

[20] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint*, 2020. Unifies diffusion models through stochastic differential equations. URL: `https://arxiv.org/abs/2011.13456`, `arXiv:2011.13456`, `doi:10.48550/arXiv.2011.13456`.

[21] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. Introduces Isomap algorithm for manifold learning. `doi:10.1126/science.290.5500.2319`.

[22] Thaddeus Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, 23(176):88–93, 1975. Accurate algorithm for geodesic calculations on ellipsoids. `doi:10.1179/sre.1975.23.176.88`.

Glossary

A glossary of key terms and concepts introduced throughout the book, with plain-language definitions for quick recall. Terms are organized alphabetically for easy reference.

A

Antipodal Points Two points on a sphere that are exactly opposite each other. There are infinitely many geodesics (great circle arcs) connecting antipodal points.

Arc Length The length of a curve on a manifold. For a geodesic, the arc length equals the distance between its endpoints.

Atlas A collection of charts that cover the entire manifold. Just as an atlas of Earth contains multiple maps covering different regions, a mathematical atlas contains multiple charts covering different neighborhoods of the manifold.

B

Bregman Divergence A divergence generated by a strictly convex potential φ: $D_\varphi(p \,\|\, q) = \varphi(p) - \varphi(q) - \langle \nabla\varphi(q), p - q\rangle$. Includes squared Euclidean and KL (via negative entropy). Asymmetric; central to mirror descent.

Bounded A set is bounded if it fits within some finite region. Open balls are bounded.

C

Calculus of Variations The branch of mathematics concerned with finding functions that minimize (or extremize) functionals. Used to derive geodesics by minimizing path length.

Chart (or **Coordinate Chart**) A mapping from a neighborhood on a manifold to a flat Euclidean space. Charts allow us to work with the manifold locally as if it were flat. The inverse mapping allows us to transfer coordinates back to the manifold.

Christoffel Symbols Quantities Γ^i_{jk} that encode how the metric tensor changes with position on a manifold. They appear in the geodesic equation and are computed from the metric and its partial derivatives.

Circle A closed 1D manifold. Geodesics on a circle are arcs, and the distance between two points is $d = r \cdot \theta$ where r is the radius and θ is the angle in radians.

Closed Set A set that includes its boundary. The complement of an open set.

Complete A manifold is geodesically complete if all geodesics can be extended indefinitely. Lines, circles, planes, and spheres are complete.

Computer Graphics The application of manifold and geodesic concepts to rendering surfaces, calculating lighting, and creating realistic animations.

Connected A space is connected if it cannot be divided into two disjoint non-empty open sets. Manifolds are typically connected.

Conjugate Points Points on a geodesic where nearby geodesics intersect. On a sphere, antipodal points are conjugate.

Convex A set is convex if the line segment between any two points lies entirely within the set. Open balls are convex.

Coordinate Transformation Changing from one coordinate system to another while preserving geometric properties. Geodesics are coordinate-independent—their descriptions change with coordinates, but the geodesics themselves are invariant.

Cut Locus The set of points where geodesics cease to be unique. On a sphere, the cut locus of a point is its antipodal point.

Curvature A measure of how much a manifold deviates from being flat. Positive curvature (like a sphere) causes geodesics to converge; negative curvature causes them to diverge; zero curvature (flat space) keeps geodesics parallel.

D

Data Manifold The low-dimensional manifold on which high-dimensional data actually lies. The data manifold hypothesis states that natural data (images, text, etc.) lies on a much lower-dimensional manifold embedded in high-dimensional space.

Data Manifold Hypothesis The principle that high-dimensional data like images, text embeddings, or audio actually lies on a low-dimensional manifold of intrinsic dimension $d \ll D$, where D is the ambient dimension.

Decoder In a VAE, the network that maps latent codes back to the data manifold. It acts as an inverse chart from the latent space to the data manifold.

Diffusion Model A generative model that learns to generate data by reversing a diffusion process. The forward process moves data off the manifold to noise, and the reverse process learns to bring it back onto the manifold.

Dimension The number of coordinates needed to describe a point locally on a manifold. A 1D manifold (like a circle) requires one coordinate, a 2D manifold (like a sphere) requires two coordinates.

Dimensionality Reduction Techniques for finding lower-dimensional representations of high-dimensional data, often based on the data manifold hypothesis.

Distance The length of the shortest path (geodesic) between two points on a manifold. On a flat plane, this is Euclidean distance; on a sphere, it's the great circle distance.

E

ELBO (Evidence Lower BOund) The objective function optimized by VAEs, consisting of a reconstruction term (encouraging accurate decoding) and a regularization term (ensuring the latent distribution matches the prior). The ELBO ensures the learned manifold structure is preserved.

Encoder In a VAE, the network that maps data from the manifold to a flat latent space. It acts as a chart from the data manifold to the latent space.

Euclidean Distance The standard distance in flat Euclidean space: $d = \sqrt{\sum(x_i - y_i)^2}$ for points with coordinates (x_i) and (y_i).

Euler-Lagrange Equations The necessary conditions for a path to minimize (or extremize) a functional. Applied to the path length functional, they give the geodesic equation.

Extrinsic Geometry Geometric properties that depend on how a manifold is embedded in a higher-dimensional space.

G

Gauss–Newton Matrix An approximation to the Hessian for least-squares-type objectives: $\mathbf{G} \approx J^\top J$ (with J a Jacobian). Closely related to pullback metrics and often aligns locally with Fisher.

GAN (Generative Adversarial Network) A generative model where a generator network learns to map from a latent space to the data manifold. The generator implicitly defines the learned manifold $\mathcal{M}_G = \{G(\mathbf{z}) : \mathbf{z} \in \mathcal{Z}\}$.

Geodesic The shortest path between two points on a curved surface or manifold. It's the natural generalization of "straight lines" to curved spaces. On a flat plane, geodesics are straight lines; on a sphere, geodesics are arcs of great circles.

Geodesic Curvature A measure of how much a curve deviates from being a geodesic. Geodesics have zero geodesic curvature—they're "as straight as possible" on the manifold.

Geodesic Equation The differential equation that geodesics satisfy: $\ddot{x}^i + \Gamma^i_{jk}\dot{x}^j\dot{x}^k = 0$, where Γ^i_{jk} are the Christoffel symbols encoding the geometry of the manifold.

Geodesic Interpolation Interpolation between two data points along a geodesic path on the manifold, ensuring the interpolated points stay on the manifold rather than leaving it.

General Relativity The theory of gravity where spacetime is a curved manifold and particles follow geodesics in spacetime.

Great Circle The largest circle that can be drawn on a sphere, formed by the intersection of the sphere with a plane passing through its center. Great circle arcs are geodesics on spheres.

Great Circle Distance The distance between two points on a sphere measured along the arc of a great circle. For a sphere of radius R, it's $d = R \cdot \arccos(\vec{v}_1 \cdot \vec{v}_2)$ where $\vec{v}_1$ and $\vec{v}_2$ are unit vectors to the points.

H

Haversine Formula A numerically stable formula for calculating great circle distances on a sphere, particularly useful for computational applications. It uses arcsin instead of arccos to avoid numerical issues near the poles.

Homeomorphism A continuous bijection with a continuous inverse. Two spaces are homeomorphic if one can be continuously deformed into the other without cutting or gluing. Manifolds are locally homeomorphic to Euclidean space.

I

Injectivity Radius The largest radius around a point p for which the exponential map $\exp_p$ is a diffeomorphism onto its image. Within this radius, geodesics are unique and minimizing.

Intrinsic Dimension The true dimensionality of a manifold, which may be much smaller than the ambient dimension in which it's embedded. For example, a sphere is intrinsically 2D even though it's embedded in 3D space.

Intrinsic Geometry Geometric properties of a manifold that depend only on the manifold itself, not on how it's embedded in a higher-dimensional space. Geodesics are intrinsic—they depend only on the metric, not on the embedding.

Isometry A mapping that preserves distances. An isometry between manifolds preserves all geometric properties.

J

Jacobian The matrix of partial derivatives of a function. For a generator $G : \mathcal{Z} \to \mathcal{M}$, the Jacobian $\mathbf{J}_G$ defines the tangent space to the manifold at each point.

Jacobi Field A vector field along a geodesic describing how nearby geodesics deviate; satisfies $\nabla_{\dot{\gamma}} \nabla_{\dot{\gamma}} J + R(J, \dot{\gamma})\dot{\gamma} = 0$.

L

Latent Space A lower-dimensional space (often Euclidean) used to represent data in a compressed form. In generative models, the latent space often corresponds to the intrinsic coordinates on the data manifold.

Local Flatness The property that every point on a manifold has a neighborhood that looks like flat Euclidean space. This is the fundamental characteristic of manifolds.

Local vs. Global Local properties hold in small neighborhoods, while global properties hold for the entire manifold. Manifolds are locally flat but may be globally curved.

M

Manifold A space that locally looks like flat Euclidean space, even if the overall shape is curved or complex. Every point has a neighborhood that can be mapped to a flat Euclidean space. Examples include lines, circles, spheres, and tori.

Manifold Learning Algorithms and techniques for learning the structure of data manifolds from data, including dimensionality reduction and representation learning.

Metric (or **Metric Tensor**) A mathematical object g_{ij} that encodes the geometry of a manifold. It defines how to measure distances and angles locally. In Euclidean space, the metric is the identity matrix; on curved manifolds, it varies with position.

Mirror Descent An optimization method using a mirror map $\nabla\varphi$ to take gradient steps in dual space and project back via a Bregman divergence; generalizes gradient descent to non-Euclidean geometries.

Mode Collapse In generative models, when the learned manifold only covers a subset of the true data manifold, failing to capture all modes of the data distribution.

N

Natural Gradient The steepest descent direction under a Riemannian metric on parameter space, typically the Fisher metric: $\tilde{\nabla} = \mathbf{F}^{-1}\nabla$. Invariant to smooth reparameterizations; improves conditioning.

Navigation The application of geodesics and distance calculations to finding routes, particularly on Earth's surface. Great circle routes are used in aviation and shipping.

Negative Curvature Curvature that causes geodesics to diverge. Examples include saddle surfaces and hyperboloids.

Numerical Stability The property of algorithms to produce accurate results even with floating-point arithmetic errors. The haversine formula is more numerically stable than the basic arccos formula for sphere distances.

O

Open n-Ball The set of all points within a certain distance r (the radius) from a center point $\mathbf{p}$ in n-dimensional Euclidean space, excluding the boundary. Denoted $B_r(\mathbf{p}) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{p}\| < r\}$. Manifolds are locally homeomorphic to open n-balls.

Open Set A set that doesn't include its boundary. Open n-balls are open sets, which is essential for the manifold definition.

P

Path Length Functional The mathematical expression for the length of a path on a manifold. For a path $\gamma(t)$, it's given by $L[\gamma] = \int \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}}\, dt$, where g_{ij} is the metric tensor.

Plane A flat 2D manifold with zero curvature. Geodesics on a plane are straight lines.

Positive Curvature Curvature that causes geodesics to converge. Examples include spheres. On a sphere, initially parallel geodesics (like lines of longitude) converge at the poles.

Pullback Metric The metric induced on a latent space by a generator mapping. If $G : \mathcal{Z} \to \mathcal{M}$ is a generator, the pullback metric is $\mathbf{g}_{ij}(\mathbf{z}) = (\mathbf{J}_G^T \mathbf{J}_G)_{ij}$ where $\mathbf{J}_G$ is the Jacobian of G.

R

Robotics The use of geodesics for path planning on curved surfaces and in configuration spaces.

S

Singularity A point where the manifold structure breaks down. For example, the tip of a cone is a singularity—it cannot be locally mapped to a flat disk.

Sphere A 2D manifold that is the set of all points equidistant from a center. A sphere has constant positive curvature.

Style Transfer The application of geodesic interpolation in generative AI to create smooth transitions between different styles or attributes.

Symmetric A set is symmetric if it looks the same from all directions. Open balls are symmetric about their center.

T

Tangent Space The space of all possible velocity vectors (tangent vectors) at a point on a manifold. On an n-dimensional manifold, the tangent space is n-dimensional.

Torus A 2D manifold shaped like a doughnut. It can be constructed by identifying opposite edges of a square or as the surface of revolution of a circle.

V

VAE (Variational Autoencoder) A generative model that learns to encode data into a latent space and decode it back. The encoder and decoder act as charts mapping between the data manifold and the flat latent space. VAEs optimize the Evidence Lower BOund (ELBO) to learn the manifold structure.

Variational Approach A method for finding geodesics by minimizing the path length functional. This leads to the Euler-Lagrange equations, which yield the geodesic equation.

Z

Zero Curvature Flat space where geodesics remain parallel. Examples include lines, planes, and cylinders.

1D and 2D Manifolds

1D Manifold A one-dimensional manifold. Examples include lines (open manifolds) and circles (closed manifolds).

2D Manifold A two-dimensional manifold. Examples include planes, spheres, and tori.

Contact Information

For questions, feedback, or collaboration opportunities, please feel free to reach out:

- **Website**: `https://vuhung16au.github.io/`
- **GitHub**: `https://github.com/vuhung16au/`
- **LinkedIn**: `https://www.linkedin.com/in/nguyenvuhung/`

License

Citation: If you use this work, please cite it appropriately. A suggested citation format is:

Nguyen, V. H. (2025). *Manifolds and Generative AI: A Mathematical Foundation.* Available at `https://github.com/vuhung16au/`