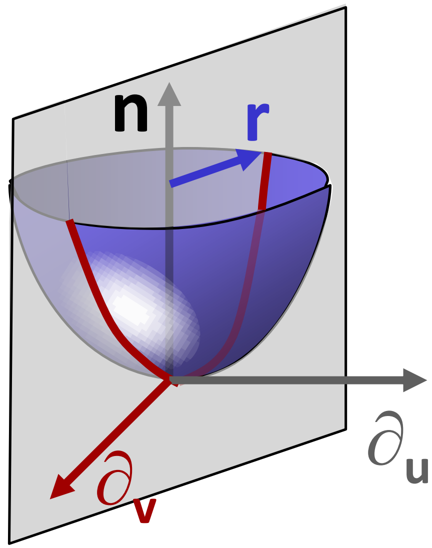
Statistical Geometry Processing

Winter Semester 2011/2012



Differential Geometry

UNIVERSITÄT DES SAARLANDES
M2CI CLUSTER OF EXCELLENCE
max planck institut informatik

Multi-Dimensional Derivatives

Derivative of a Function

Reminder: The *derivative* of a function is defined as

$$\frac{d}{dt}f(t) := \lim_{h\to 0}\frac{f(t+h)-f(t)}{h}$$

If limit exists: function is called *differentiable*.

Other notation:

$$\frac{d}{dt}f(t) = \underbrace{f'(t)}_{\substack{\text{variable}\\ \text{from context}}} = \underbrace{\dot{f}(t)}_{\substack{\text{time}\\ \text{variables}}}$$

$$\underbrace{\frac{d^k}{dt^k}f(t) = f^{(k)}(t)}_{\substack{\text{repeated differentiation}\\ \text{(higher order derivatives)}}}$$

Taylor Approximation

Smooth functions can be approximated locally:

- $$f(x) \approx f(x_0) + \frac{d}{dx} f(x_0)(x - x_0) + \frac{1}{2}\frac{d^2}{dx^2} f(x_0)(x - x_0)^2 + \ldots + \frac{1}{k!}\frac{d^k}{dx^k} f(x_0)(x - x_0)^k + O(x^{k+1})$$
- Convergence: holomorphic functions
- Local approximation for smooth functions

Rule of Thumb

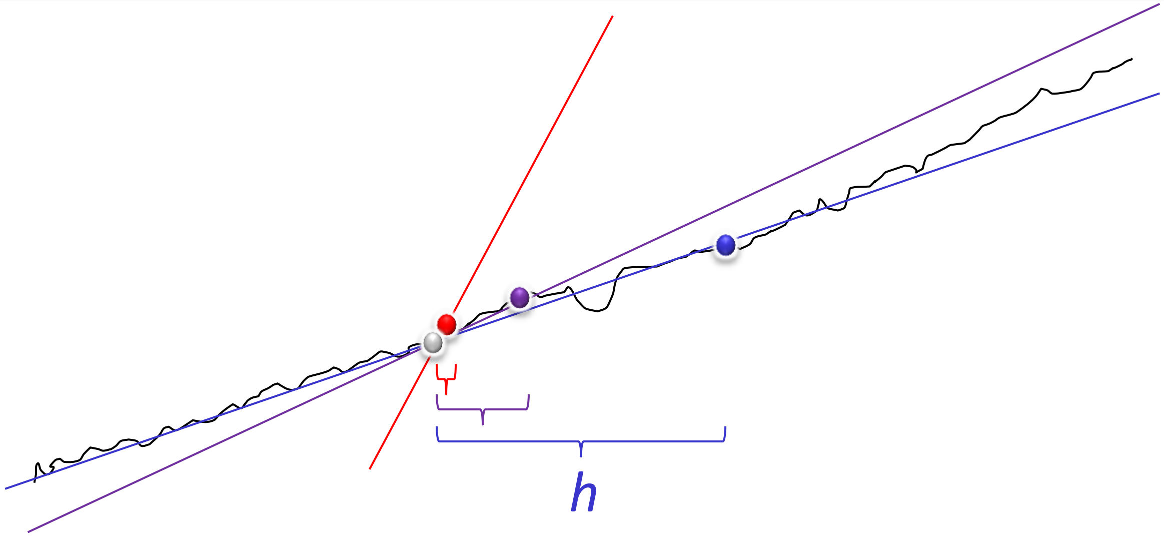
Derivatives and Polynomials

- Polynomial: $f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 \ldots$
  - 0th-order derivative: $f(0) = c_0$
  - 1st-order derivative: $f'(0) = c_1$
  - 2nd-order derivative: $f''(0) = 2c_2$
  - 3rd-order derivative: $f'''(0) = 6c_3$
  - ...

Rule of Thumb:

- Derivatives correspond to polynomial coefficients
- Estimate derivates ↔ polynomial fitting

Differentiation is Ill-posed!



Regularization

- Numerical differentiation needs regularization
  - Higher order is more problematic
- Finite differences (larger *h*)
- Averaging (polynomial fitting) over finite domain

Partial Derivative

Multivariate functions:

- Notation changes:

use curly-d

$$\frac{\partial}{\partial x_k} f(x_1,\dots,x_{k-1},x_k,x_{k+1},\dots,x_n) :=$$

$$\lim_{h\to 0} \frac{f(x_1,\dots,x_{k-1},x_k+h,x_{k+1},\dots,x_n) - f(x_1,\dots,x_{k-1},x_k,x_{k+1},\dots,x_n)}{h}$$

- Alternative notation:

$$\frac{\partial}{\partial x_k} f(\mathbf{x}) = \partial_k f(\mathbf{x}) = f_{x_k}(\mathbf{x})$$

Special Cases

Derivatives for:

- Functions f: $\mathbb{R}^n \to \mathbb{R}$ (“heightfield”)
- Functions f: $\mathbb{R} \to \mathbb{R}^n$ (“curves”)
- Functions f: $\mathbb{R}^n \to \mathbb{R}^m$ (general case)

Special Cases

Derivatives for:

- Functions f: $\mathbb{R}^n \to \mathbb{R}$ ("heightfield")
- Functions f: $\mathbb{R} \to \mathbb{R}^n$ ("curves")
- Functions f: $\mathbb{R}^n \to \mathbb{R}^m$ (general case)

Gradient

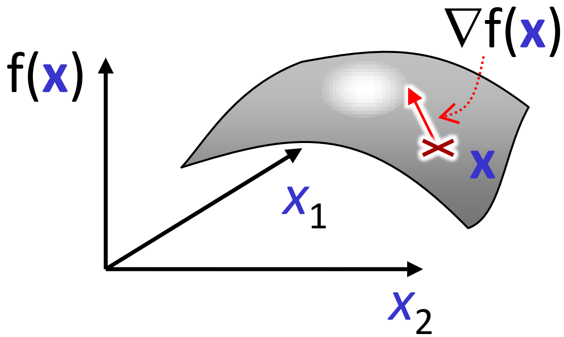
Gradient:

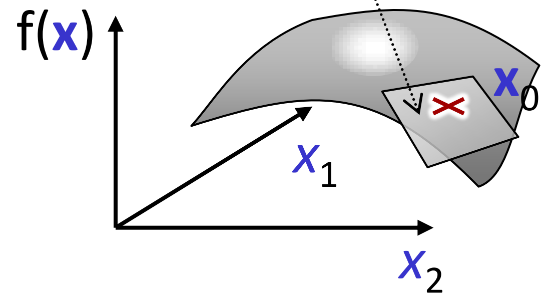
- Given a function f: $\mathbb{R}^n \rightarrow \mathbb{R}$ ("heightfield")
- The vector of all partial derivatives of f is called the *gradient*:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} f(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_n} f(\mathbf{x}) \end{pmatrix}$$

Gradient

Gradient:





- gradient: vector pointing in direction of steepest ascent.
- Local linear approximation (Taylor):

$$f(\mathbf{x}) \approx f(\mathbf{x}_0)+\nabla f(\mathbf{x}_0)\cdot(\mathbf{x}-\mathbf{x}_0)$$

Higher Order Derivatives

Higher order Derivatives:

- Can do all combinations: $\left(\frac{\partial}{\partial x_{i_1}}\frac{\partial}{\partial x_{i_2}}\dots\frac{\partial}{\partial x_{i_k}}\right)f$

- Order does not matter for $f \in C^k$

Hessian Matrix

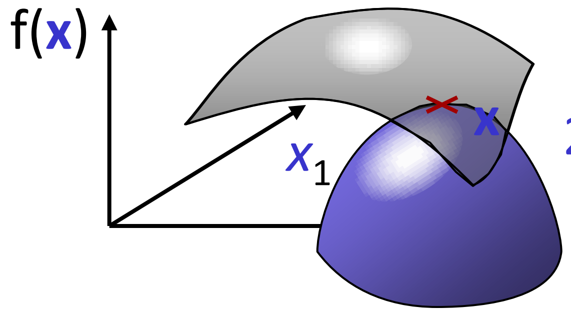
Higher order Derivatives:

- Important special case: Second order derivative

$$\begin{pmatrix} \frac{\partial^2}{\partial {x_1}^2} & \frac{\partial}{\partial x_2}\frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n}\frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2} & \frac{\partial^2}{\partial {x_1}^2} & & \frac{\partial}{\partial x_n}\frac{\partial}{\partial x_2} \\ \vdots & & \ddots & \vdots \\ \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_n} & \frac{\partial}{\partial x_2}\frac{\partial}{\partial x_n} & \cdots & \frac{\partial^2}{\partial {x_n}^2} \end{pmatrix} f(\mathbf{x}) =: H_f(\mathbf{x})$$

- "Hessian" matrix (symmetric for $f \in C^2$)
- Orthogonal Eigenbasis, full Eigenspectrum

Taylor Approximation



2nd order approximation (schematic)

Second order Taylor approximation:

- Fit a paraboloid to a general function

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^{\mathrm{T}} \cdot H_f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)$$

Special Cases

Derivatives for:

- Functions f: $\mathbb{R}^n \to \mathbb{R}$ (“heightfield”)
- Functions f: $\mathbb{R} \to \mathbb{R}^n$ (“curves”)
- Functions f: $\mathbb{R}^n \to \mathbb{R}^m$ (general case)

Derivatives of Curves

Derivatives of vector valued functions:

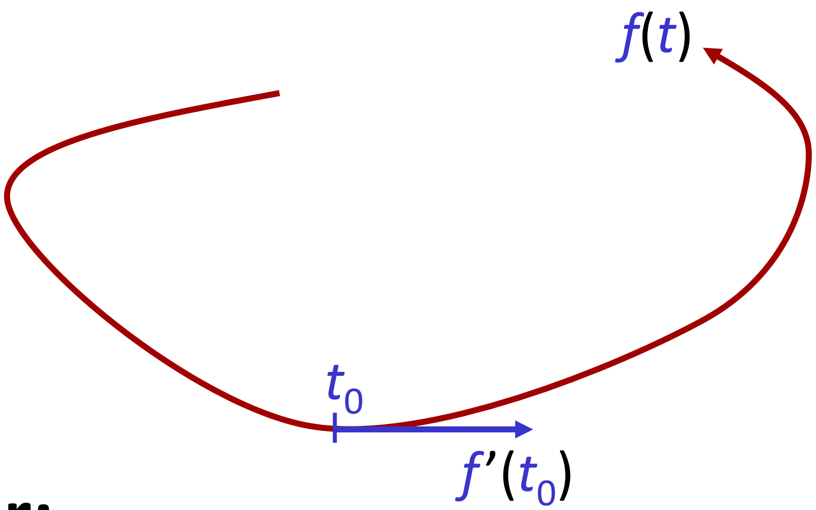
- Given a function f: $\mathbb{R} \rightarrow \mathbb{R}^n$ ("curve")

$$f(t) = \begin{pmatrix} f_1(t) \\ \vdots \\ f_n(t) \end{pmatrix}$$

- We can compute derivatives for every output dimension:

$$\frac{d}{dt} f(t) := \begin{pmatrix} \frac{d}{dt} f_1(t) \\ \vdots \\ \frac{d}{dt} f_n(t) \end{pmatrix} =: f'(t) =: \dot{f}(t)$$

Geometric Meaning



Tangent Vector:

- f': tangent vector
- Motion of physical particle: $\dot{f}$ = velocity.
- Higher order derivatives: Again vector functions
- Second derivative $\ddot{f}$ = acceleration

Special Cases

Derivatives for:

- Functions $f\colon \mathbb{R}^n \to \mathbb{R}$ ("heightfield")
- Functions $f\colon \mathbb{R} \to \mathbb{R}^n$ ("curves")
- Functions $f\colon \mathbb{R}^n \to \mathbb{R}^m$ (general case)

You can combine it...

General case:

- Given a function *f*: $\mathbb{R}^n \rightarrow \mathbb{R}^m$ ("space warp")

$$f(\mathbf{x}) = f((x_1,\dots,x_n)) = \begin{pmatrix} f_1(x_1,\dots,x_n) \\ \vdots \\ f_m(x_1,\dots,x_n) \end{pmatrix}$$

- Maps points in space to other points in space
- First derivative: Derivatives of all *output components* of f w.r.t. all *input directions*.
- "Jacobian matrix": denoted by ∇f or $\mathbf{J}_f$

Jacobian Matrix

Jacobian Matrix:

$$\nabla f(\mathbf{x}) = J_f(\mathbf{x}) = \nabla f(x_1,\dots,x_n)$$

$$= \begin{pmatrix} \nabla f_1(x_1,\dots,x_n)^{\mathrm{T}} \\ \vdots \\ \nabla f_m(x_1,\dots,x_n)^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \partial_{x_1} f_1(\mathbf{x}) & \cdots & \partial_{x_n} f_1(\mathbf{x}) \\ \vdots & & \vdots \\ \partial_{x_1} f_m(\mathbf{x}) & \cdots & \partial_{x_n} f_m(\mathbf{x}) \end{pmatrix}$$

Use in a first-order Taylor approximation:

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + J_f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

↑ matrix / vector product

Coordinate Systems

Problem:

- What happens, if the coordinate system changes?
- Partial derivatives go into different directions then.
- Do we get the same result?

Total Derivative

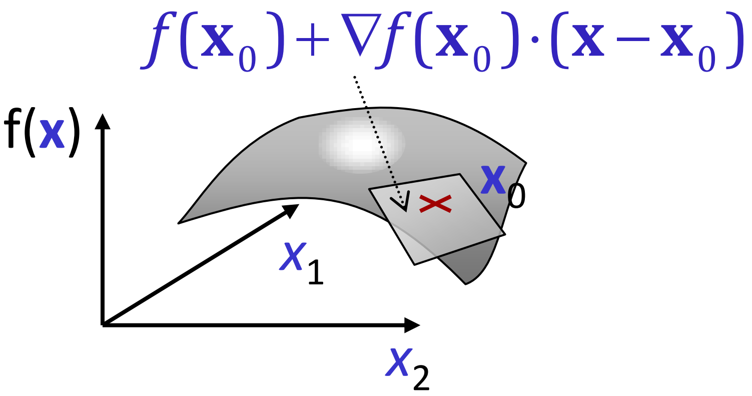
First order Taylor approx.:

- $f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) + R_{x_0}(\mathbf{x})$
- Converges for C^1 functions

  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$

  $$\lim_{\mathbf{x} \to \mathbf{x}_0} \frac{R_{x_0}(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_0\|} = 0,$$

  ("totally differentiable")



Partial Derivatives

Consequences:

- A linear function: fully determined by image of a basis
- Hence: Directions of partial derivatives do not matter – this is just a basis transform.
  - We can use any linear independent set of directions $\mathbf{T}$
  - Transform to standard basis by multiplying with $\mathbf{T}^{-1}$
- Similar argument for higher order derivatives

Directional Derivative

The directional derivative is defined as:

- Given $f: \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$, $||\mathbf{v}|| = 1$.
- Directional derivative:

$$\nabla_{\mathbf{v}} f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{v}}(\mathbf{x}) := \frac{d}{dt} f(\mathbf{x} + t\mathbf{v})$$

- Compute from Jacobian matrix

$$\nabla_{\mathbf{v}} f(\mathbf{x}) = \nabla f(\mathbf{x}) \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

(requires total differentiability)

Multi-Dimensional Optimization

Optimization Problems

Optimization Problem:

- Given a C^1 function $f\colon \mathbb{R}^n \rightarrow \mathbb{R}$ (general heightfield)
- We are looking for a local extremum (minimum / maximum) of this function

Theorem:

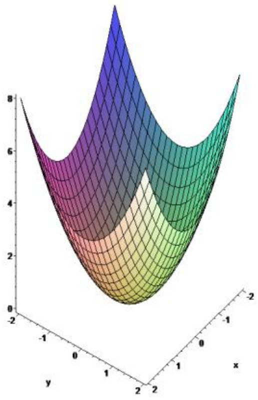
- $\mathbf{x}$ is a local extremum $\Rightarrow \nabla f(\mathbf{x}) = \mathbf{0}$

Sketch of a proof: If $\nabla f(\mathbf{x}) \neq 0$, we can walk a small step in gradient direction to improve the score further (in case of a maximum, minimum similar).

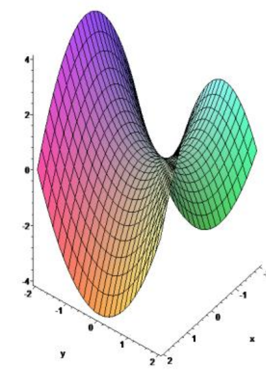
Critical Points

Critical points:

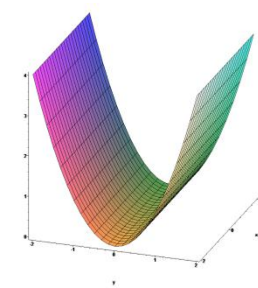
- $\nabla f(\mathbf{x}) = \mathbf{0}$ does not guarantee an extremum (saddle points)
- Points with $\nabla f(\mathbf{x}) = \mathbf{0}$ are called *critical points*.
- Final decision via *Hessian matrix*:
  - All eigenvalues > 0: local minimum
  - All eigenvalues < 0: local maximum
  - Mixed eigenvalues: saddle point
  - Some zero eigenvalues: critical line



$\lambda_i > 0$



$\lambda_0 > 0, \lambda_1 < 0$



$\lambda_0 = 0, \lambda_1 > 0$

Quadratic Optimization

Quadratic Case:

- f: $\mathbb{R}^n \rightarrow \mathbb{R}$
- Objective function: $f(\mathbf{x}) = \mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{x} + \mathbf{b}^\mathrm{T}\mathbf{x} + c$
  - symmetric $n \times n$ matrix $\mathbf{A}$
  - n-dim. vector $\mathbf{b}$
  - constant c
- Gradient: $\nabla f(\mathbf{x}) = 2\mathbf{A}\mathbf{x} + \mathbf{b}$
- Critical points: solution to $2\mathbf{A}\mathbf{x} = -\mathbf{b}$
- Solution: Solve system of linear equations

Example

Gradient computation example:

$$[x, y]\begin{pmatrix} a \\ b \end{pmatrix} = ax + by \rightarrow \nabla = \begin{pmatrix} a \\ b \end{pmatrix}$$

$$[x, y]\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = ax^2 + 2bxy + cy^2$$

$$\left.\begin{matrix} \partial_x \rightarrow 2ax + 2by \\ \partial_y \rightarrow 2bx + 2cy \end{matrix}\right\} \nabla \rightarrow 2\mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix}$$

Global Extrema of Quadratic Funcs.

Three cases:

- Eigenvalues of **A** ≥ 0: critical points are *global minima*
- Eigenvalues of **A** ≤ 0: critical points are *global maxima*
- Mixed eigenvalues: no global minimum/maximum exists (minimum and maximum at infinity)

Structure:

- Critical points form an affine subspace of $\mathbb{R}^n$.
- I.e.: Point, line, plane...

Non-Linear Optimization Algorithms

Non-Quadratic Optimization

Optimization Problems:

- Find (local/global) minimum of E: $\mathbb{R}^n \supseteq \Omega \to \mathbb{R}$.
- E for "energy" (motivated from physics)
- What to do if E is non-quadratic?

Gradient Descent

Gradient Descent:

- Gradient ∇E points into direction of steepest ascent.
- Walking a small step in direction $-\nabla E$ will decrease the energy.
- When $\nabla E = \mathbf{0}$, a critical point is found.

Properties:

- For sufficiently small steps, this algorithm is guaranteed to converge
- Generally slow convergence
- Does not work in practice for ill-conditioned problems

Newton Optimization

Newton Optimization

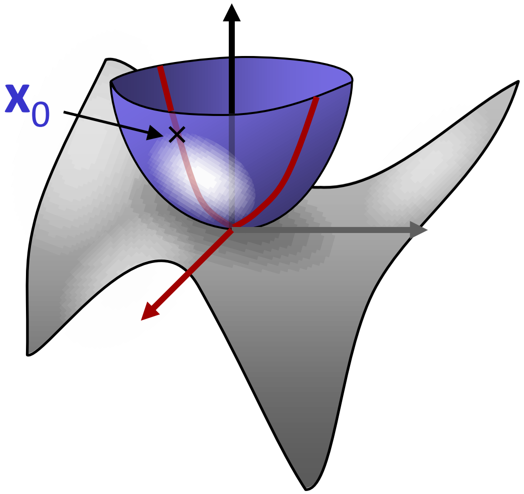
- Basic idea: Local quadratic approximation of E:

$$E(\mathbf{x}) \approx E(\mathbf{x}_0) + \nabla E(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^{\mathrm{T}} \cdot H_E(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)$$

- Solve for vertex (critical point) of the fitted parabola
- Iterate until a minimum is found ($\nabla E = 0$)

Properties:

- Typically much faster convergence, more stable
- No convergence guarantee

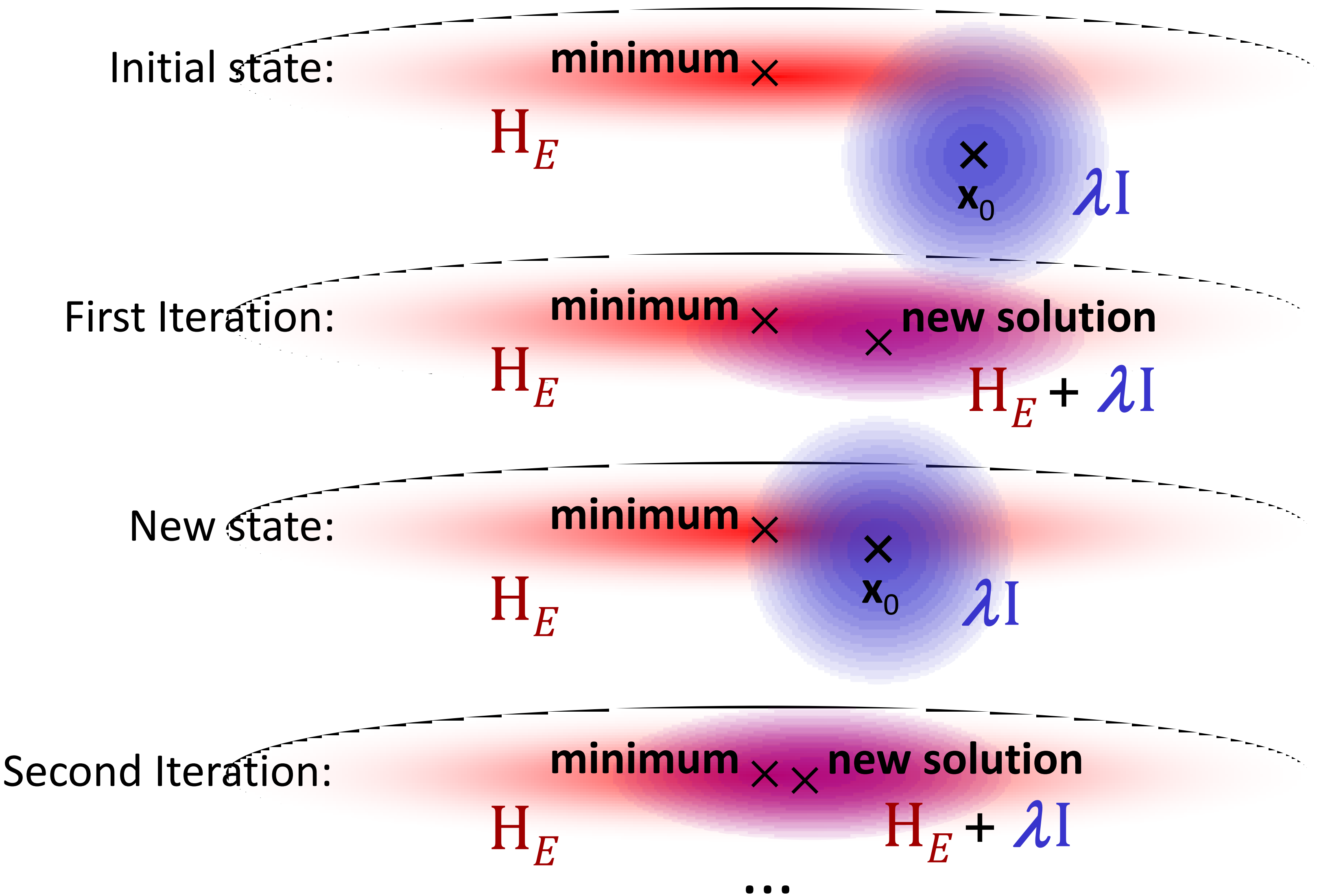


Newton Optimization - Divergence

Regularization:

- Hessian matrix: for negative eigenvalues, steps might point uphill
- (Near-) zero eigenvalues make problem ill-conditioned.
- Simple solution: Add $\lambda \mathbf{I}$ to the Hessian for a small λ.
- Sum of two quadrics: $\lambda \mathbf{I}$ keeps solution at $\mathbf{x}_0$.
- This is an example of regularization

Handling Indefinite Situations



Further Algorithms

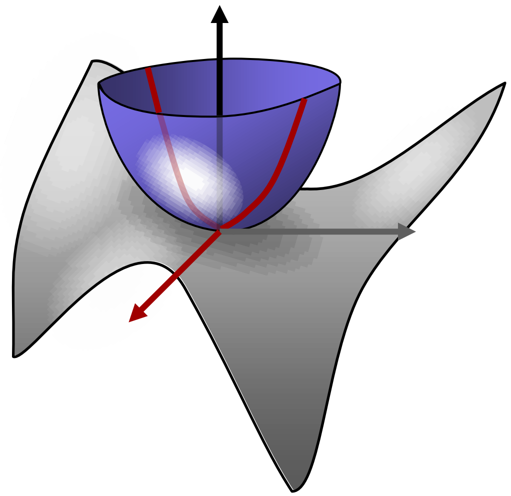
Gradient descent line search:

- Optimize step size for gradient descent
  - Fit 1D parabola to *E* in gradient direction
  - Perform 1D Newton search
  - If E does not decrease at the new position:
    - Try to half step width (say up to 10-20 times).
    - If this still does not decrease *E*, stop and output local minimum.

Further Algorithms

Line search for Newton-optimization:

- Following the quadratic fit might overshoot
- Line search:
  - Test value of E at new position
  - Half step width until error decreases (say 10-20 iterations)
  - Switch to gradient descent, if this does not work



Convex Problems

General Classification:

- Non-linear optimization problems can be hard to solve.
- What is definitely “easy”?

Convex Problems:

- *Convex functions* on a *convex domain* can be optimized “easily” using a generic algorithm.
- Other problems *might* be hard to solve.

Convex Problems

Convex Function:

- A C^2 function E is convex, if $H_E > 0$ (all eigenvalues of the Hessian are strictly positive everywhere)
- A set Ω is convex if every line connecting two points from Ω is also contained in Ω.
- A convex function has at most one local minimum

Problem Properties:

- Assume a global minimum exists
- Will be the only local minimum
- Can be reached on a straight line from any point in Ω

Convex Problems

Generic Optimization Algorithm (Sketch):

- Gradient descent
- Start at any point $\mathbf{p} \in \Omega$
- Perform gradient descent in "small enough" steps
- In case of hitting the domain boundary, project on boundary surface (follow the wall)
- When the gradient becomes zero, the minimum is found

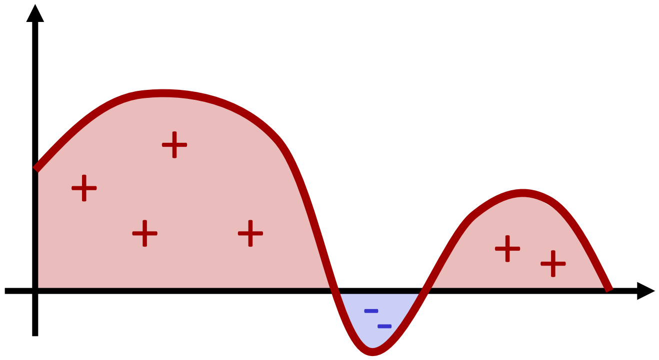
There are more efficient algorithms...

Multi-Dimensional Integrals

Integral

Integral of a function

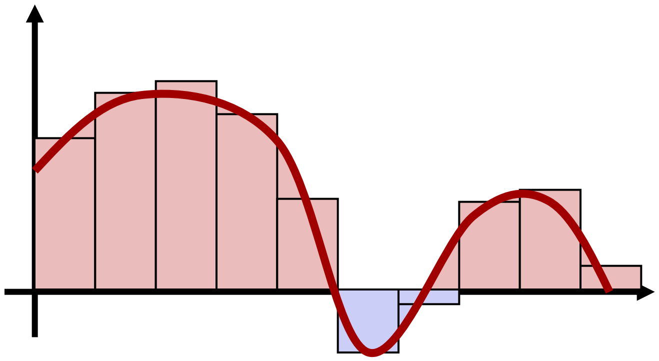
- Function f: $\mathbb{R} \to \mathbb{R}$
- Integral $\int_a^b f(t)dt$ measures signed area under curve:



Integral

Numerical Approximation

- Sum up a series of approximate shapes

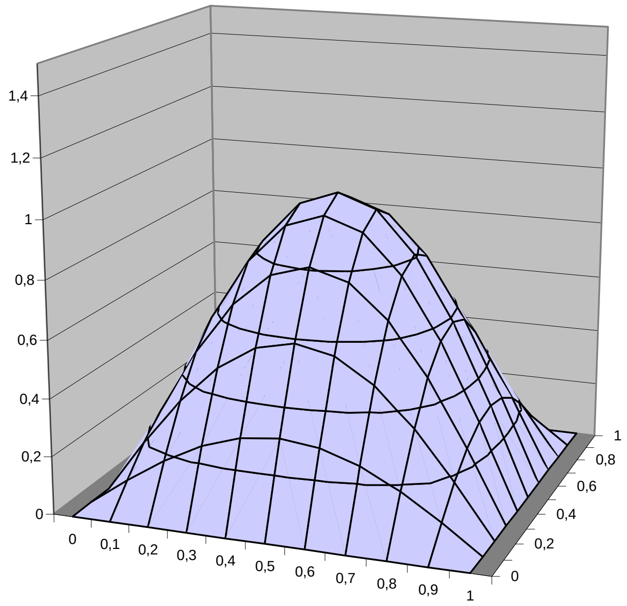


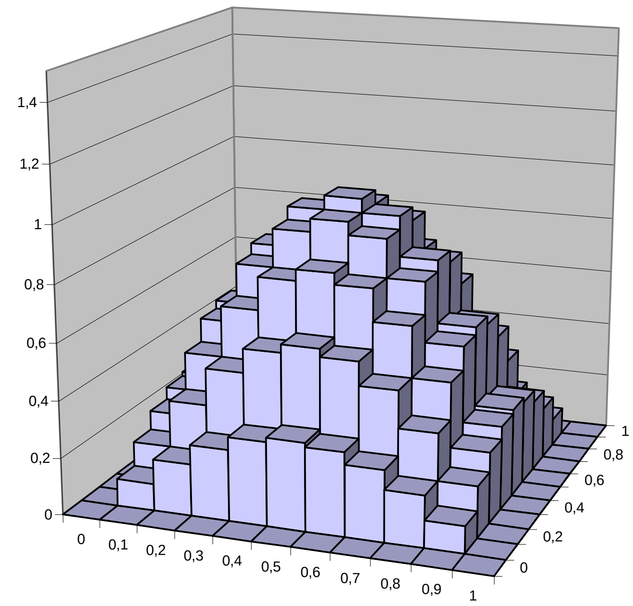
- (Riemannian) Definition: limit for baseline $\rightarrow$ zero

Multi-Dimensional Integral

Integration in higher dimensions

- Functions f: $\mathbb{R}^n \to \mathbb{R}$
- Tessellate domain and sum up volume of cuboids



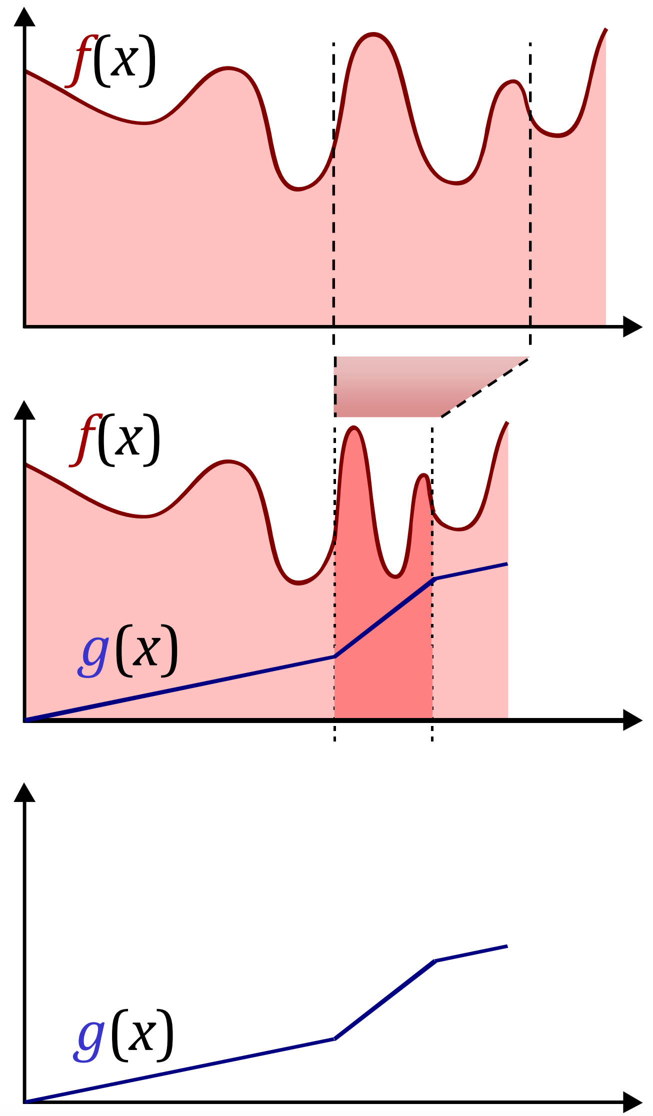


Integral Transformations

Integration by substitution:

$$\int_a^b f(x)dx = \int_{g^{-1}(a)}^{g^{-1}(b)} f(g(t))g'(t)dt$$

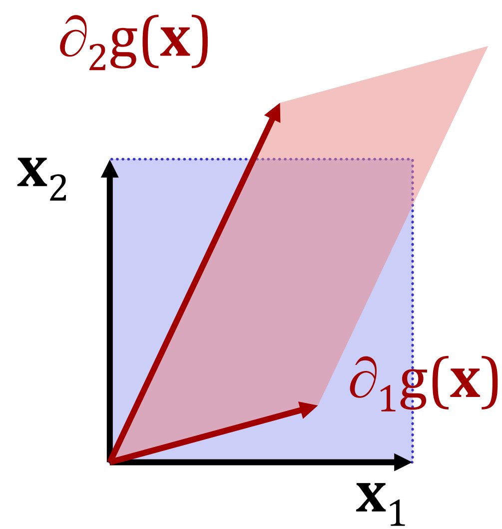
Need to compensate for speed of movement that shrinks the measured area.



Multi-Dimensional Substitution

Transformation of Integrals:

$$\int_{\Omega} f(\mathbf{x})d\mathbf{x} = \int_{g^{-1}(\Omega)} f\big(g(\mathbf{y})\big)\left|\det\big(\nabla g(\mathbf{y})\big)\right| d\mathbf{y}$$

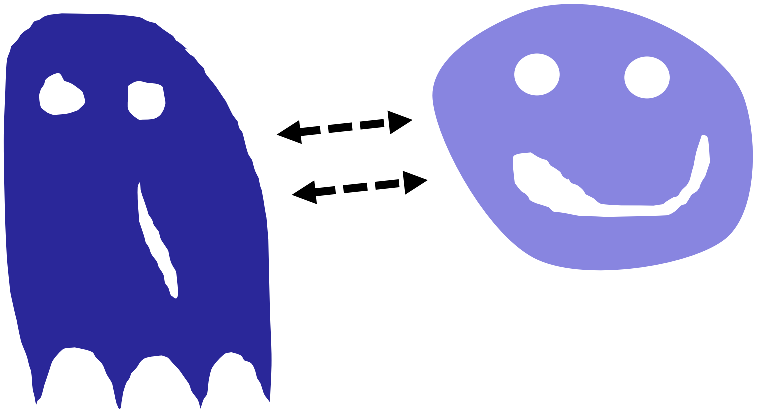


- $g \in C^1$, invertible
- Jacobian approximates local behavior of $g()$
- Determinant: local area/volume change
- In particular: $\left|\det\big(\nabla g(\mathbf{y})\big)\right| = 1$ means $g()$ is *area/volume conserving*.

Topology

- a very short primer -

A Few Concepts from Topology



Homeomorphism:

- $f : X \rightarrow Y$
- f is bijective
- f is continuous
- f^{-1} exists and is continuous
- Basically, a continuous deformation

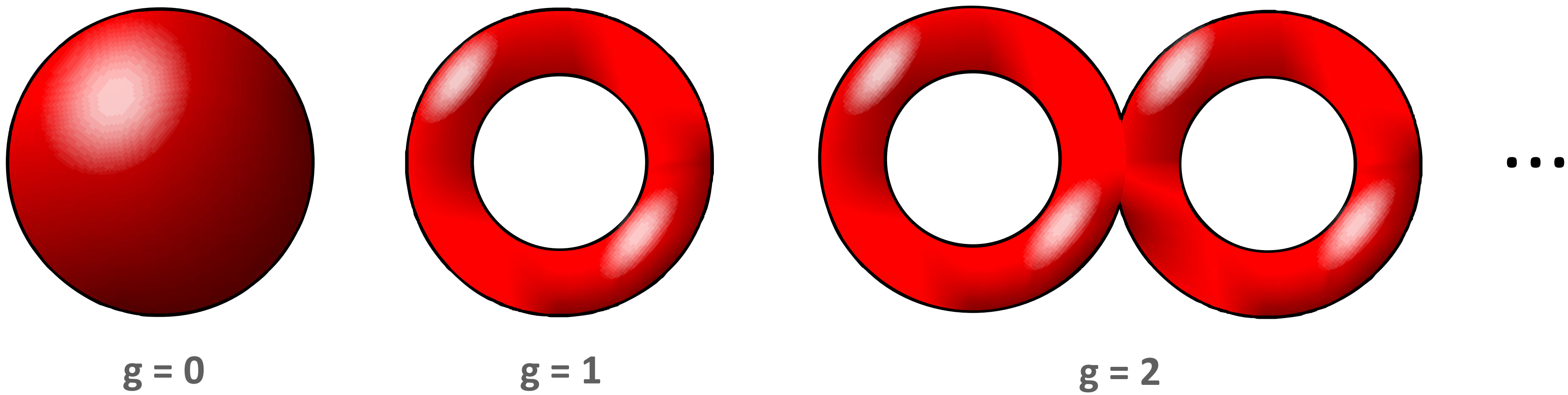
Topological equivalence

- Objects are topologically equivalence if there exists a homeomorphism that maps between them
- “Can be deformed into each other”

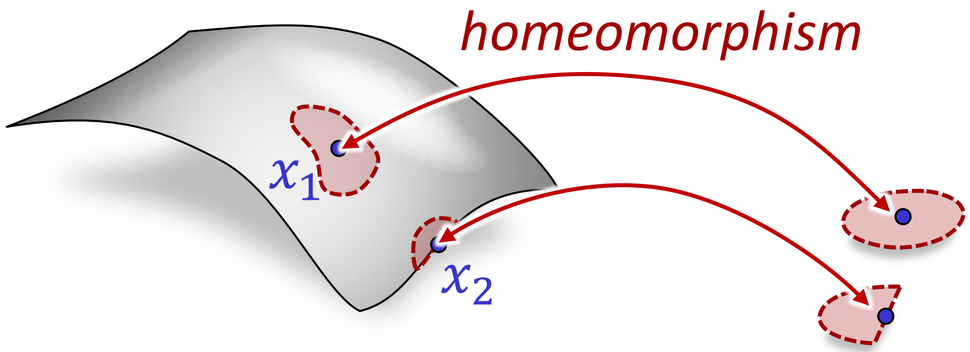
Surfaces

Boundaries of volumes in 3D

- Topological Equivalence classes
  - Sphere
  - Torus
  - n-fold Torus
- Genus = number of tunnels



Manifold



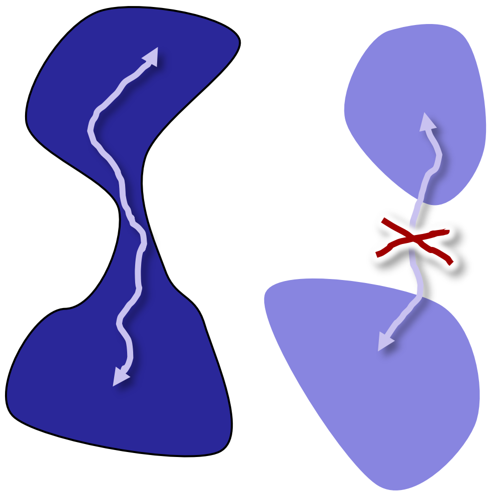
Definition: Manifold

- A d-manifold M:
  At every $x \in M$ there exists an ϵ-environment homeomorphic to a d-dimensional disc
- With boundary: *disc* or *half-disc*

Further concepts

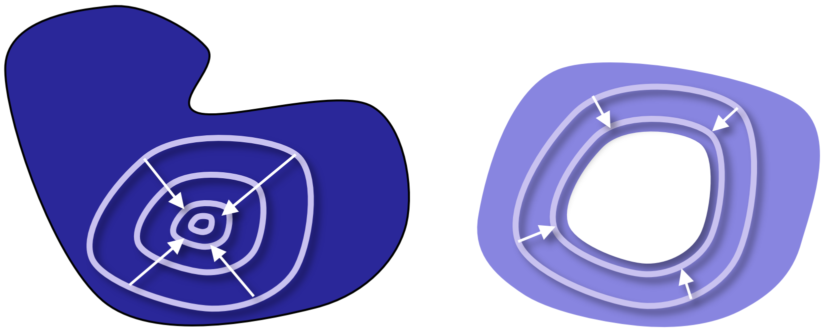
Connected Set

- There exists a continuous curve within the set between all pairs of points



Simply Connected

- Every closed loop can be continuously shrunken until it disappears



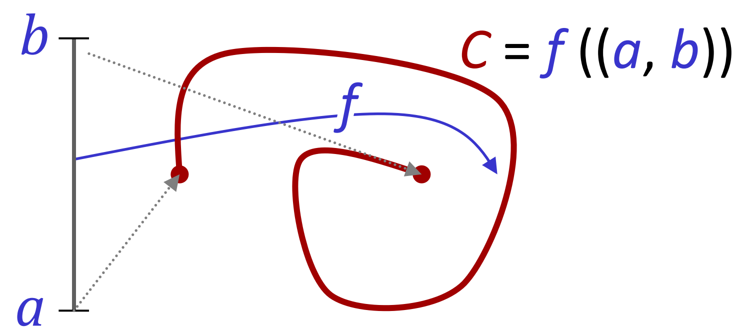
Differential Geometry

of Curves & Surfaces

Part I: Curves

Parametric Curves

Parametric Curves:



- A differentiable function

  $f: (a, b) \to \mathbb{R}^n$

  describes a *parametric curve*

  $C = f((a, b)), C \subseteq \mathbb{R}^n$.

- The parametrization is called *regular* if $f'(t) \neq 0$ for all t.

- If $\| f'(t) \| \equiv 1$ for all t, f is called a *unit-speed parametrization* of the curve C.

Length of a Curve

The length of a curve:

- The length of a regular curve C is defined as:

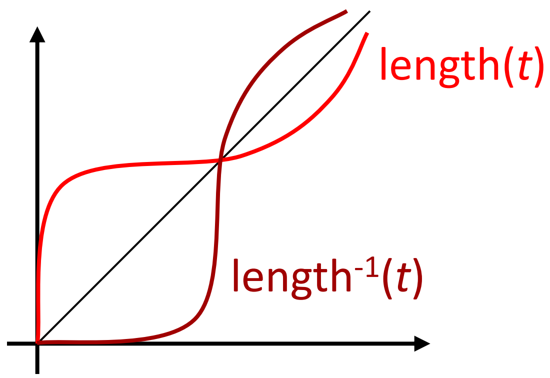
$$\text{length}(C) = \int_a^b \|f'(t)\| dt$$

- Independent of the parametrization (integral transformation theorem).
- Alternative: length(C) = $|b - a|$ for a unit-speed parametrization

Reparametrization

Enforcing unit-speed parametrization:

- Assume: $\| f'(t) \| \neq 0$ for all t.
- We have:
  $$\text{length}(C) = \int_a^b \| f'(t) \| dt$$ (invertible, because $f'(t) > 0$)
- Concatenating $f \circ \text{length}^{-1}(C)$ yields a unit-speed parametrization of the curve



Tangents

Unit Tangents:

- The unit tangent vector at $x \in (a, b)$ is given by:

$$\text{tangent}(t) = \frac{f'(t)}{\|f'(t)\|}$$

- For curves $C \subseteq \mathbb{R}^2$, the unit normal vector of the curve is defined as:

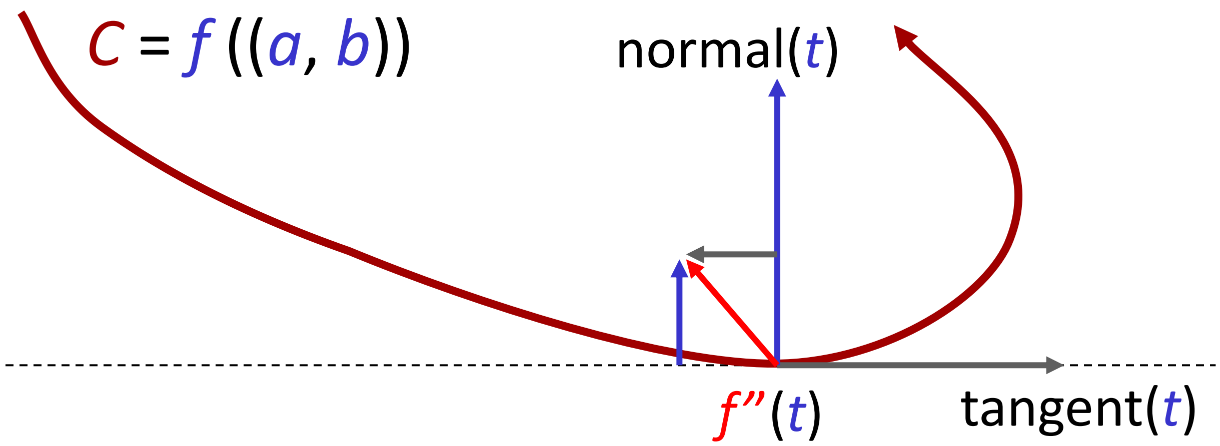
$$\text{normal}(t) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{f'(t)}{\|f'(t)\|}$$

Curvature

Curvature:

- First derivatives show curve direction / speed of movement.
- Curvature is encoded in 2nd order information.
- Why not just use f''?
- Problem: Depends on parametrization
  - Different velocity yields different results
  - Need to distinguish between acceleration in tangential and non-tangential directions.

Curvature & 2nd Derivatives



Definition of curvature

- We want only the non-tangential component of f''.
- Accelerating/slowing down does not matter for curvature of the traced out curve C.
- Need to normalize speed.

Curvature

Curvature of a Curve C ∈ $\mathbb{R}^2$:

$$\boldsymbol{\kappa 2}(t) = \frac{\left\langle f''(t), \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} f'(t) \right\rangle}{\left\| f'(t) \right\|^3}$$

- Normalization factor:
  - Divide by $\|f'\|$ to obtain unit tangent vector
  - Divide again twice to normalize f''
    - Taylor expansion / chain rule:

$$f(\lambda t) = f(t_0) + \lambda f'(t_0)(t - t_0) + \frac{1}{2}\lambda^2 f''(t)(t - t_0)^2 + O(t^3)$$

    - Second derivative scales quadratically with speed

Unit-speed parametrization

Unit-speed parametrization:

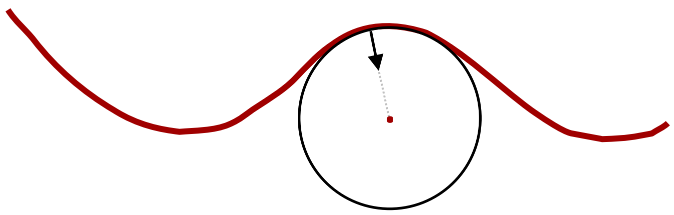
- Assume a unit-speed parametrization, i.e. $\|f'\| \equiv 1$.
- Then, $\boldsymbol{\kappa}2$ simplifies to:

$$\boldsymbol{\kappa 2}(t) = \|f''(t)\|$$

Radius of Curvature

Easy to see:

- Curvature of a circle is constant, $\kappa 2 \equiv \pm 1/r$ (r = radius). (see problem sets)
- Accordingly: Define radius of curvature as $1/\kappa 2$.
- Osculating circle:
  - Radius: $1/\kappa 2$
  - Center: $f(t) + \frac{1}{\kappa 2} normal(t)$



Theorems

Definition:

- Rigid motion: $\mathbf{x} \to \mathbf{A}\mathbf{x}+\mathbf{b}$ with orthogonal $\mathbf{A}$
  - Orientation preserving (no mirroring) if $\det(\mathbf{A}) = +1$
  - Mirroring leads to $\det(\mathbf{A}) = -1$

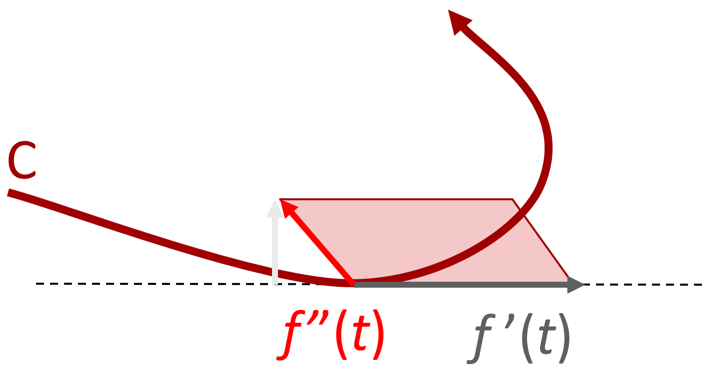
Theorems for plane curves:

- Curvature is invariant under rigid motion
  - Absolute value is invariant
  - Signed value is invariant for orientation preserving rigid motion
- Two unit speed parameterized curves with identical signed curvature function differ only in a orientation preserving rigid motion.

Space Curves

General case: Curvature of a Curve $C \subseteq \mathbb{R}^n$

- *W.l.o.g.*: Assume we are given a unit-speed parametrization f of C
- The *curvature* of C at parameter value t is defined as:
  $$\boldsymbol{\kappa}(t) = \|f''(t)\|$$
- For a general, regular curve $C \subseteq \mathbb{R}^3$ (any regular parametrization):
  $$\boldsymbol{\kappa}(t) = \frac{\|f'(t) \times f''(t)\|}{\|f'(t)\|^3}$$
- General curvature is unsigned



Torsion

Characteristics of Space Curves in $\mathbb{R}^3$:

- Curvature not sufficient
- Curve may “bend” in space
- Curvature is a 2nd order property
- 2nd order curves are always flat
  - Quadratic curves are specified by 3 points in space, which always lie in a plane
  - Cannot capture out-of-plane bends
- Missing property: Torsion

Torsion

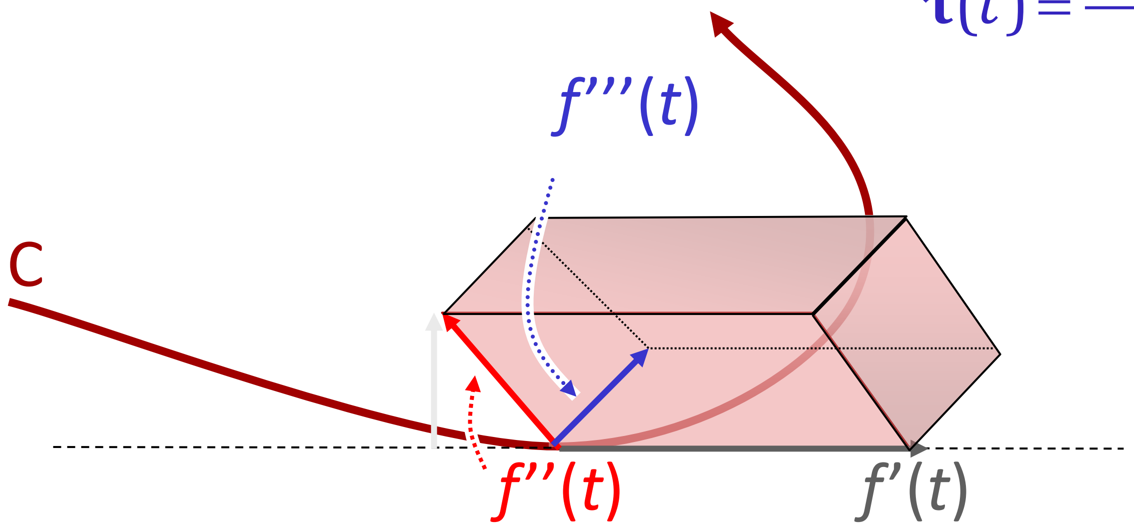
Definition:

- Let f be a regular parametrization of a curve $C \subseteq \mathbb{R}^3$ with non-zero curvature
- The torsion of f at t is defined as

$$\boldsymbol{\tau}(t) = \frac{f'(t) \times f''(t) \cdot f'''(t)}{\left\| f'(t) \times f''(t) \right\|^2} = \frac{\det\left(f'(t), f''(t), f'''(t)\right)}{\left\| f'(t) \times f''(t) \right\|^2}$$

Illustration

$$\tau(t) = \frac{\det\left(f'(t), f''(t), f'''(t)\right)}{\left\| f'(t) \times f''(t) \right\|^2}$$



Theorem

Fundamental Theorem of Space Curves

- Two unit speed parameterized curves C $\subseteq \mathbb{R}^3$ with identical, positive curvature and identical torsion are identical up to a rigid motion.

Part II: Surfaces

Parametric Patches

Parametric Surface Patches:

A smoothly differentiable function

$$f\colon \mathbb{R}^2 \supseteq \Omega \to \mathbb{R}^n$$

describes a *parametric surface patch*

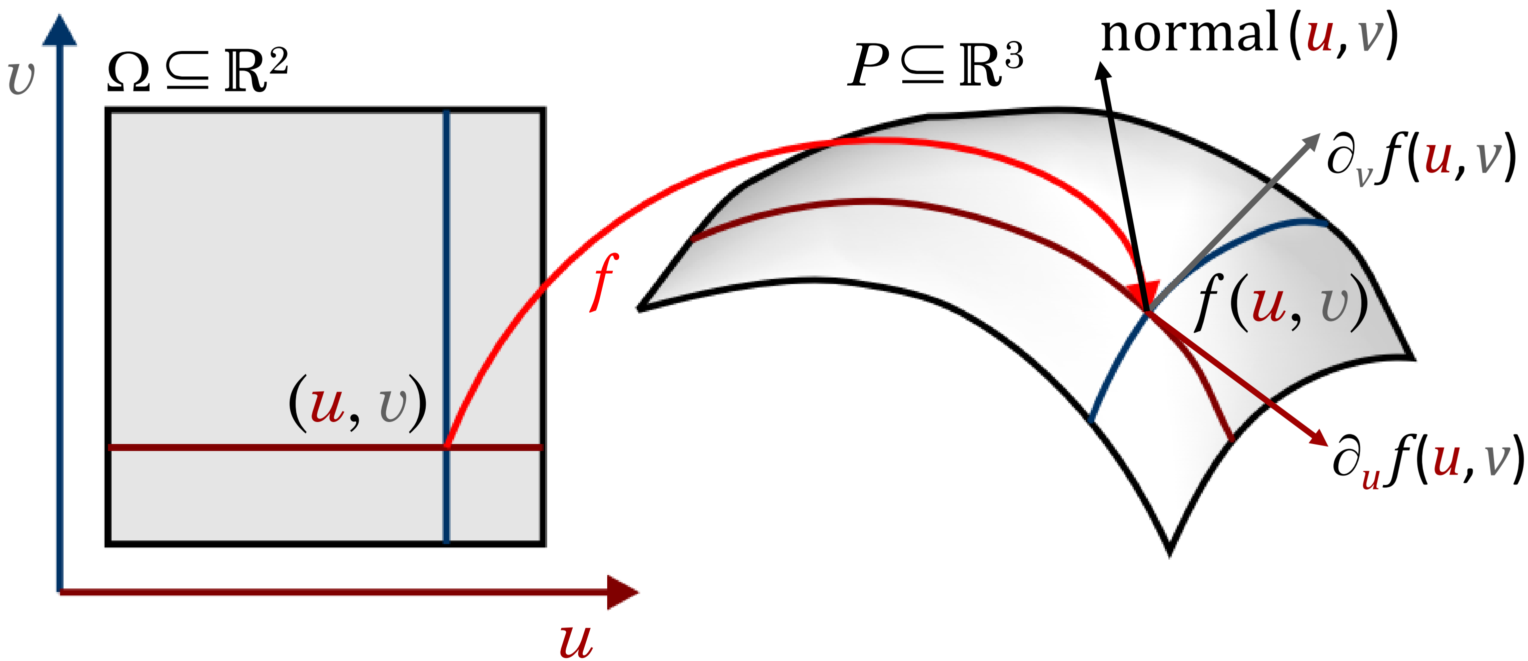
$$P = f(\Omega),\ P \subseteq \mathbb{R}^n.$$

Parametric Patches

Function $\mathrm{f}(\mathbf{x}) = f(u, v) \rightarrow \mathbb{R}^3$

- Tangents: $\frac{d}{dt} f(\mathbf{x}_0 + t\mathbf{r}) = \nabla_{\mathbf{r}} f(\mathbf{x}_0)$
- Canonical tangents: $\partial_u f(u,v),\ \partial_v f(u,v)$
- Normal: $\mathbf{n}(\mathbf{x}_0) = \frac{\partial_u f(u,v) \times \partial_v f(u,v)}{\|\partial_u f(u,v) \times \partial_v f(u,v)\|}$

Illustration



Surface Area

Surface Area:

- Patch P: $f : \Omega \to \mathbb{R}^3$
- Computation is simple
- Integrate over constant function f ≡ 1 over surface
- Then apply integral transformation theorem:

$$\text{area}(P) = \int_{\Omega} \left\| \partial_u f(\mathbf{x}) \times \partial_u f(\mathbf{x}) \right\| d\mathbf{x}$$

Fundamental Forms

Fundamental Forms:

- Describe the local parametrized surface
- Measure...
  - ...distortion of length (first fundamental form)
  - ...surface curvature (second fundamental form)
- Parametrization independent surface curvature measures will be derived from this

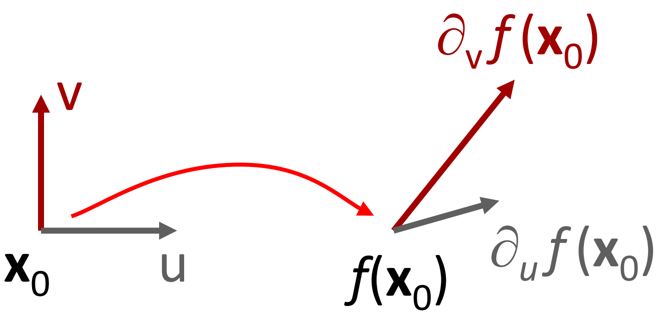
First Fundamental Form

First Fundamental Form

- Also known as *metric tensor*.
- Given a regular parametric patch f: $\mathbb{R}^2 \supseteq \Omega \to \mathbb{R}^3$.
- f will distort angles and distances
- We will look at a local first order Taylor approximation to measure the effect:

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

- Length changes become visible in the scalar product...

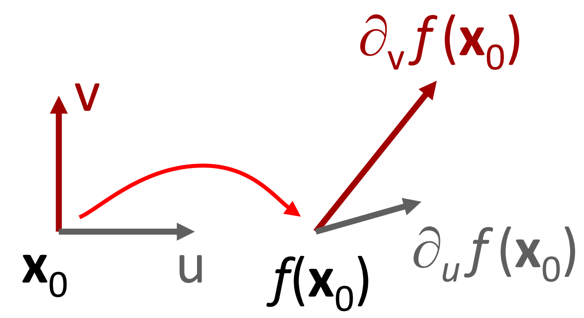


First Fundamental Form

First Fundamental Form

- First order Taylor approximation:

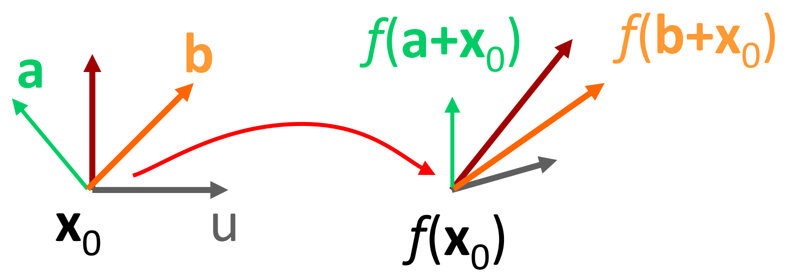
$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$



- Scalar product of vectors a, b $\in \mathbb{R}^2$:

$$\langle f(\mathbf{x}_0 + \mathbf{a}) - f(\mathbf{x}_0), f(\mathbf{x}_0 + \mathbf{b}) - f(\mathbf{x}_0) \rangle \approx \langle \nabla f(\mathbf{x}_0)\mathbf{a}, \nabla f(\mathbf{x}_0)\mathbf{b} \rangle$$

$$= \mathbf{a}^{\mathrm{T}} \underbrace{\left(\nabla f(\mathbf{x}_0)^{\mathrm{T}} \nabla f(\mathbf{x}_0) \right)}_{\text{first fundamental form}} \mathbf{b}$$



First Fundamental Form

First Fundamental Form

- The first fundamental form can be written as a 2×2 matrix:

$$\left(\nabla f^{\mathrm{T}} \nabla f\right) = \begin{pmatrix} \partial_{\mathbf{u}} f \, \partial_{\mathbf{u}} f & \partial_{\mathbf{u}} f \, \partial_{\mathbf{v}} f \\ \partial_{\mathbf{u}} f \, \partial_{\mathbf{v}} f & \partial_{\mathbf{v}} f \, \partial_{\mathbf{v}} f \end{pmatrix} =: \begin{pmatrix} E & F \\ F & G \end{pmatrix} \qquad \mathbf{I}(\mathbf{x},\mathbf{y}) := \mathbf{x}^{\mathrm{T}} \left(\nabla f^{\mathrm{T}} \nabla f\right) \mathbf{y}$$

- The matrix is symmetric and positive definite (regular parametrization, semi-definte otherwise)
- Defines a *generalized scalar product* that measures lengths and angles *on the surface*.

Second Fundamental Form

Problems:

- The first fundamental form measures length changes only.
- A cylinder looks like a flat sheet in this view.
- We need a tool to measure curvature of a surface as well.
- This requires second order information.
  - Any first order approximation is inherently “flat”.

Second Fundamental Form

Definition:

- Given: regular parametric patch f: $\mathbb{R}^2 \supseteq \Omega \to \mathbb{R}^3$.
- *Second fundamental* form:
  (a.k.a. *shape operator*, *curvature tensor*)

$$S(\mathbf{x}_0) = \begin{pmatrix} \partial_{\mathbf{uu}} f(\mathbf{x}_0) \cdot \mathbf{n} & \partial_{\mathbf{uv}} f(\mathbf{x}_0) \cdot \mathbf{n} \\ \partial_{\mathbf{uv}} f(\mathbf{x}_0) \cdot \mathbf{n} & \partial_{\mathbf{vv}} f(\mathbf{x}_0) \cdot \mathbf{n} \end{pmatrix}$$

- Notation:

$$\mathbf{II}(\mathbf{x},\mathbf{y}) = \mathbf{x}^{\mathrm{T}} \begin{pmatrix} \partial_{\mathbf{uu}} f(\mathbf{x}_0) \cdot \mathbf{n} & \partial_{\mathbf{uv}} f(\mathbf{x}_0) \cdot \mathbf{n} \\ \partial_{\mathbf{uv}} f(\mathbf{x}_0) \cdot \mathbf{n} & \partial_{\mathbf{vv}} f(\mathbf{x}_0) \cdot \mathbf{n} \end{pmatrix} \mathbf{y}$$

Second Fundamental Form

Basic Idea:

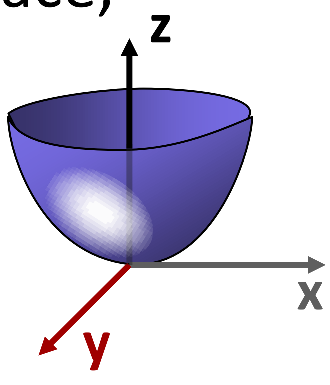
- Compute second derivative vectors
- Project in normal direction (remove tangential acceleration)

Alternative Computation

Alternative Formulation (Gauss):

- Local height field parameterization f(**x**,**y**) = **z**
- Orthonormal **x**,**y** coordinates *tangential* to surface, **z** in normal direction, origin at zero
- 2nd order Taylor representation:

$$f(\mathbf{x}) \approx \frac{1}{2}\underbrace{\mathbf{x}^{\mathrm{T}} f''(\mathbf{x})\mathbf{x}} + \underbrace{f'(\mathbf{x})\mathbf{x} + f(0)}_{0}$$
$$= ex^2 + 2fxy + gy^2$$



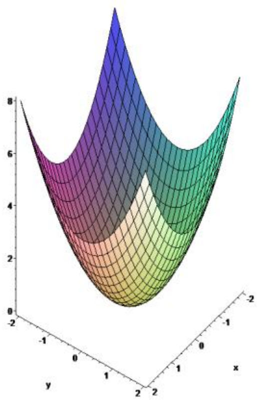
- Second fundamental form: Matrix of second derivatives

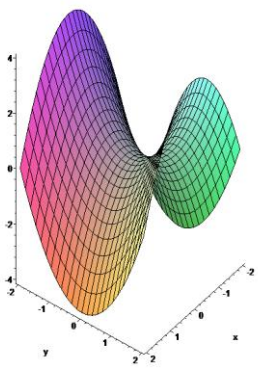
$$\begin{pmatrix} \partial_{xx} f & \partial_{xy} f \\ \partial_{xy} f & \partial_{yy} f \end{pmatrix} =: \begin{pmatrix} e & f \\ f & g \end{pmatrix}$$

Basic Idea

In other words:

- *First fundamental form:* **I**
  Linear part (squared) of local Taylor approximation.
- *Second fundamental form:* **II**
  Quadratic part of heightfield approximation
- Both matrices are symmetric.
  - Next: eigenanalysis, of course...

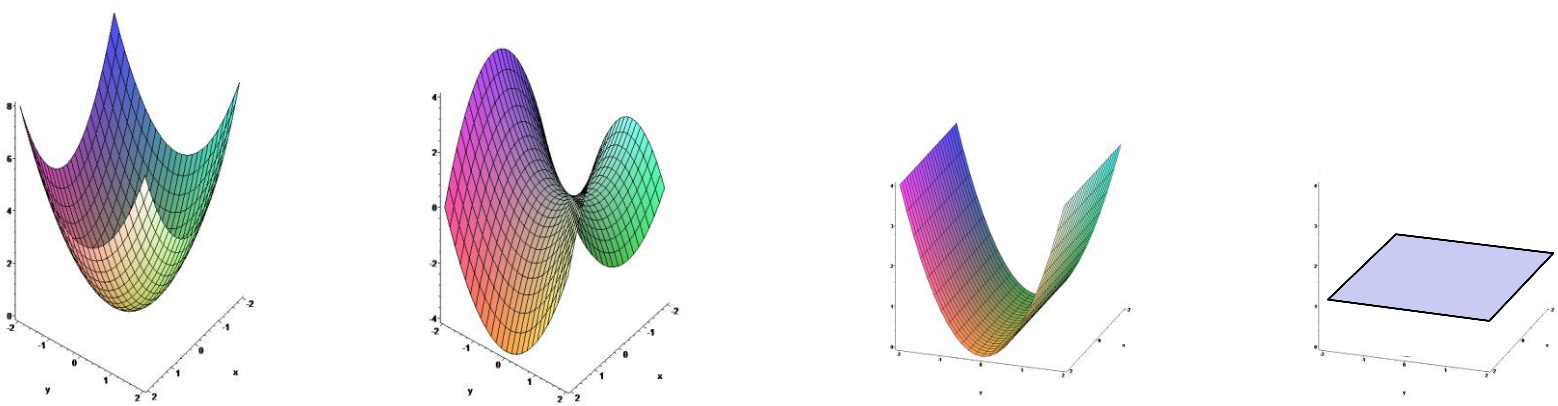




Principal Curvature

Eigenanalysis:

- Eigenvalues of *second fundamental form* for an *orthonormal tangent* basis are called *principal curvatures* κ_1, κ_2.
- Corresponding orthogonal eigenvectors are called *principal directions of curvature*.



$\kappa_i > 0$ | $\kappa_0 > 0, \kappa_1 < 0$ | $\kappa_0 = 0, \kappa_1 > 0$ | $\kappa_0 = 0, \kappa_1 = 0$ | …

Normal Curvature

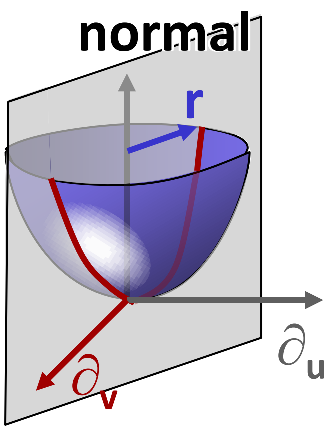
Definition:

- The *normal curvature* $k(\mathbf{r})$ in direction $\mathbf{r}$ for a unit length direction vector $\mathbf{r}$ at parameter position $\mathbf{x}_0$ is given by:

$$k_{\mathbf{x}_0}(\mathbf{r}) = \mathbf{II}_{\mathbf{x}_0}(\mathbf{r},\mathbf{r}) = \mathbf{r}^T\mathbf{S}(\mathbf{x}_0)\mathbf{r}$$

Relation to Curvature of Plane Curves:

- Intersect the surface locally with plane spanned by **normal** and $\mathbf{r}$ through point $\mathbf{x}_0$.
- Identical curvatures (up to sign).



Principal Curvatures

Relation to principal curvature:

- The maximum principal cuvature κ_1 is the maximum of the normal curvature
- The minimum principal cuvature κ_2 is the minimum of the normal curvature

Gaussian & Mean Curvature

More Definitions:

- The Gaussian curvature K is the product of the principal curvatures: $K = \kappa_1\kappa_2$
- The mean curvature H is the average: $H = 0.5 \cdot (\kappa_1 + \kappa_2)$

Theorems:

- $K(\mathbf{x}_0) = \det\left(S(x_0)\right) = \dfrac{eg - f^2}{EG - F^2}$
- $H(\mathbf{x}_0) = \dfrac{1}{2}\operatorname{tr}\left(S(x_0)\right) = \dfrac{eG - 2fF + gE}{2\left(EG - F^2\right)}$

Global Properties

Definition:

- An *isometry* is a mapping between surfaces that preserves distances on the surface (geodesics)
- A *developable surface* is a surface with Gaussian curvature zero everywhere (i.e. no curvature in at least one direction)
  - Examples: Cylinder, Cone, Plane
- A developable surface can be locally mapped to a plane isometrically (flattening out, unroll).

Theorema Egregium

Theorema egregium (Gauss):

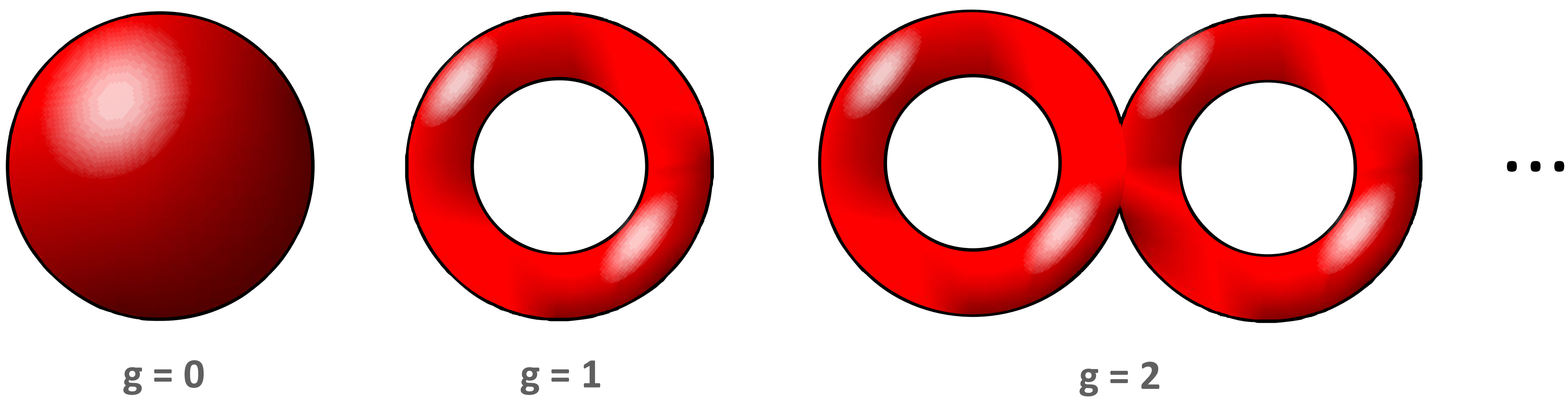
- Any isometric mapping preservers Gaussian curvature, i.e. Gaussian curvature is invariant under isometric maps ("intrinsic surface property")
- Consequence: The earth ($\approx$ sphere) cannot be mapped to a plane in an exactly length preserving way.

Gauss Bonnet Theorem

Gauss Bonnet Theorem:

For a compact, orientable surface without boundary in $\mathbb{R}^3$, the area integral of the Gauss curvature is related to the genus g of the surface:

$$\int_S K(x)dx = 4\pi(1-g)$$



Fundamental Theorem of Surfaces

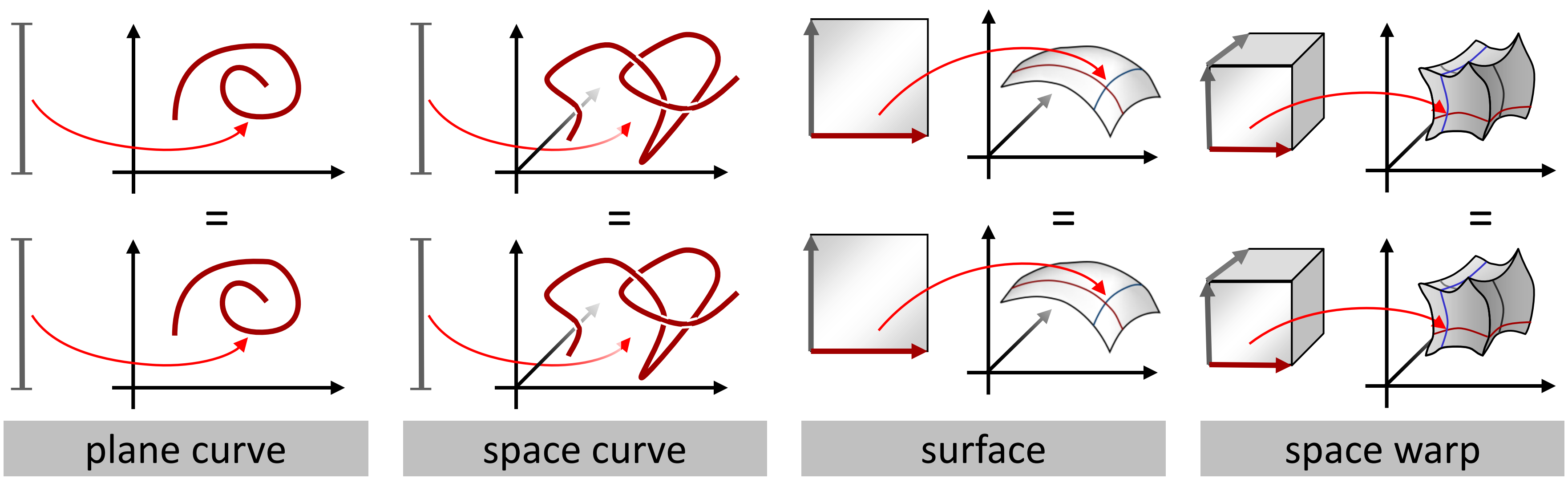
Theorem:

- Given two parametric patches in $\mathbb{R}^3$ defined on the same domain Ω.
- Assume that the first and second fundamental form are identical.
- Then there exists a rigid motion that maps on surface to the other.

Summary

Objects are the same up to a rigid motion, if...:

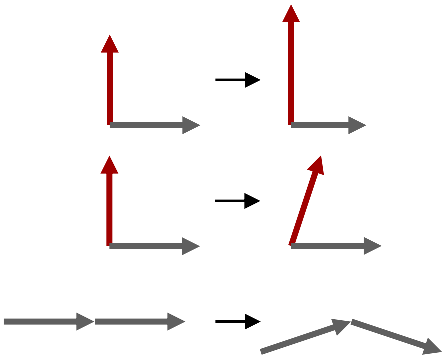
- Curves $\mathbb{R} \to \mathbb{R}^2$: Same *speed*, same *curvature*
- Curves $\mathbb{R} \to \mathbb{R}^3$: Same *speed*, same *curvature*, *torsion*
- Surfaces $\mathbb{R}^2 \to \mathbb{R}^3$: Same *first* & *second* fundamental form
- Volumetric Objects $\mathbb{R}^3 \to \mathbb{R}^3$: Same *first* fundamental form



Deformation Models

What if this does not hold?

- Deviation in fundamental forms is a measure of deformation
- Example: Surfaces
  - Diagonals of $\mathbf{I}_1$ - $\mathbf{I}_2$: scaling (stretching)
  - Off-diagonals of $\mathbf{I}_1$ - $\mathbf{I}_2$: sheering
  - Elements of $\mathbf{II}_1$ - $\mathbf{II}_2$: bending
- This is the basis of *deformation models*.



Reference: **D. Terzopoulos, J. Platt, A. Barr, K. Fleischer:** Elastically Deformable Models. In: *Siggraph '87 Conference Proceedings (Computer Graphics 21(4))*, 1987.