Geometric Modeling

Summer Semester 2012

Linear Algebra & Function Spaces

(Recap)

UNIVERSITÄT DES SAARLANDES

CLUSTER OF EXCELLENCE

max planck institut informatik

Announcement

Room change:

- On Thursday, April 26th, room 024 is occupied.
- The lecture will be moved to room 021, E1 4 (the Tuesday's lecture room).
- Only on this date.

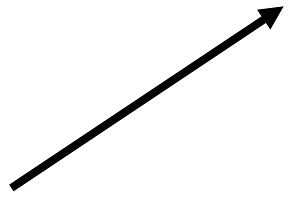
Today...

Topics:

- Introduction: Geometric Modeling
  - Motivation
  - Overview: Topics
  - Basic modeling techniques
- Mathematical Background
  - Function Spaces
  - Differential Geometry
- Interpolation and approximation
- Spline curves

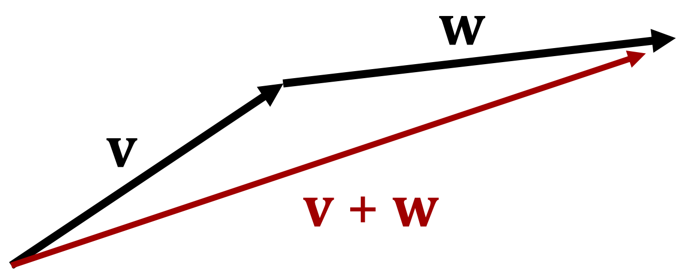
Vector Spaces

Vectors



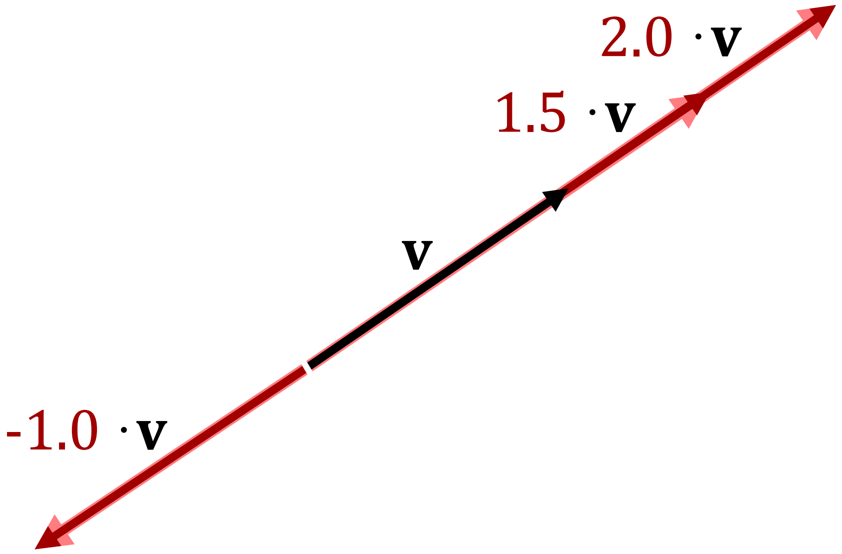
vectors are arrows in space
classically: 2 or 3 dim. Euclidian space

Vector Operations



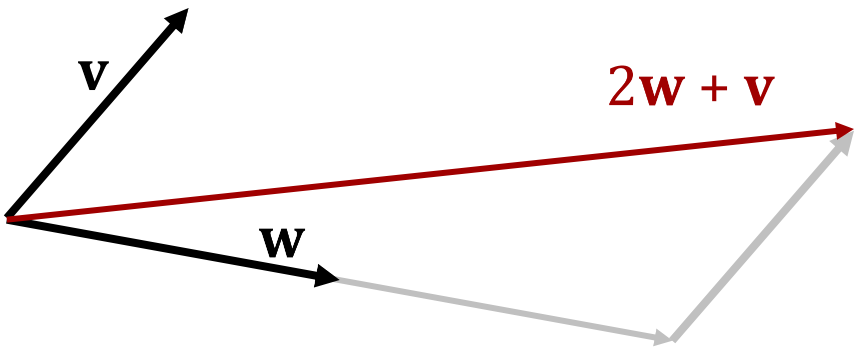
“Adding” Vectors:
Concatenation

Vector Operations



Scalar Multiplication:
Scaling vectors (incl. mirroring)

You can combine it...



Linear Combinations:

This is basically all you can do.

$$\mathbf{r} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i$$

Vector Spaces

Vector space:

- Set of vectors V
- Based on field F (we use only F = $\mathbb{R}$)
- Two operations:
  - Adding vectors **u** = **v** + **w** (**u**, **v**, **w** $\in$ V)
  - Scaling vectors w = λv (**u** $\in$ V, λ $\in$ F)
- Vector space *axioms*:

(a1) $\forall \mathbf{u},\mathbf{v},\mathbf{w} \in \mathrm{V}: \ (\mathbf{u}+\mathbf{v})+\mathbf{w} = \mathbf{u}+(\mathbf{v}+\mathbf{w})$

(a2) $\forall \mathbf{u},\mathbf{v} \in \mathrm{V}: \ \mathbf{u}+\mathbf{v} = \mathbf{v}+\mathbf{u}$

(a3) $\exists \mathbf{0}_{\mathrm{V}} \in \mathrm{V} : \forall \mathbf{v} \in \mathrm{V}: \ \mathbf{v}+\mathbf{0}_{\mathrm{V}} = \mathbf{v}$

(a4) $\forall \mathbf{v} \in \mathrm{V} : \exists \mathbf{w} \in \mathrm{V}: \ \mathbf{v}+\mathbf{w} = \mathbf{0}_{\mathrm{V}}$

(s1) $\forall \mathbf{v} \in \mathrm{V}, \lambda, \mu \in \mathrm{F}: \ \lambda(\mu\mathbf{v}) = (\lambda\mu)\mathbf{v}$

(s2) $\text{for } 1_{\mathrm{F}} \in \mathrm{F} : \forall \mathbf{v} \in \mathrm{V}: \ 1_{\mathrm{F}}\mathbf{v} = \mathbf{v}$

(s3) $\forall \lambda \in \mathrm{F} : \forall \mathbf{v},\mathbf{w} \in \mathrm{V}: \ \lambda(\mathbf{v}+\mathbf{w}) = \lambda\mathbf{v}+\lambda\mathbf{w}$

(s4) $\forall \lambda, \mu \in \mathrm{F}, \mathbf{v} \in \mathrm{V}: \ (\lambda+\mu)\mathbf{v} = \lambda\mathbf{v}+\mu\mathbf{v}$

Additional Tools

More concepts:

- Subspaces, linear spans, bases
- Scalar product
  - Angle, length, orthogonality
  - Gram-Schmidt orthogonalization
- Cross product ($\mathbb{R}^3$)
- Linear maps
  - Matrices
- Eigenvalues & eigenvectors
- Quadratic forms

(Check your old math books)

Structure

Vector spaces

- Any finite-dim., real vector space is isomorphic to $\mathbb{R}^n$
  - Arrays of numbers
  - Behave like arrows in a flat (Euclidean) geometry
- Proof:
  - Construct basis
  - Represent as span of basis vectors

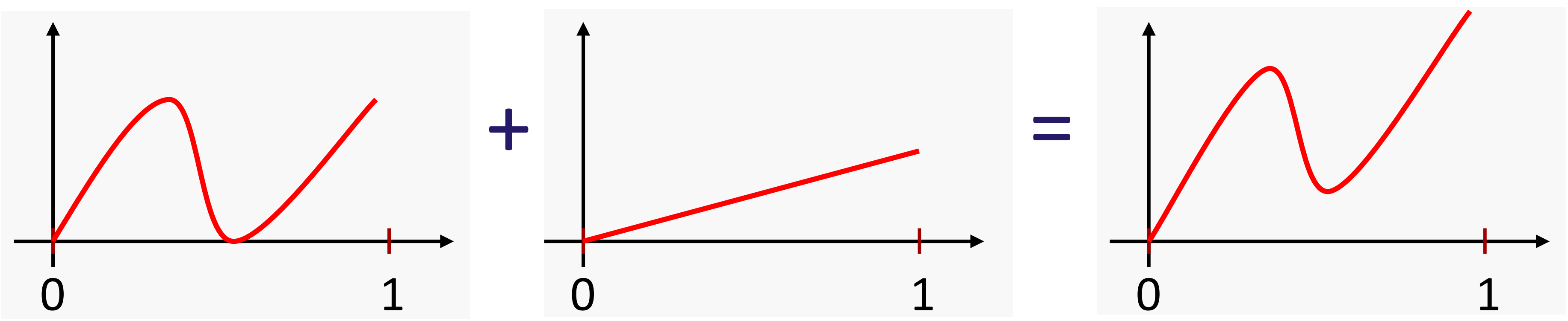
Infinite-dimensional spaces

- Require more numbers
  - Same principle
  - Approximate with finite basis

Example Spaces

Function spaces:

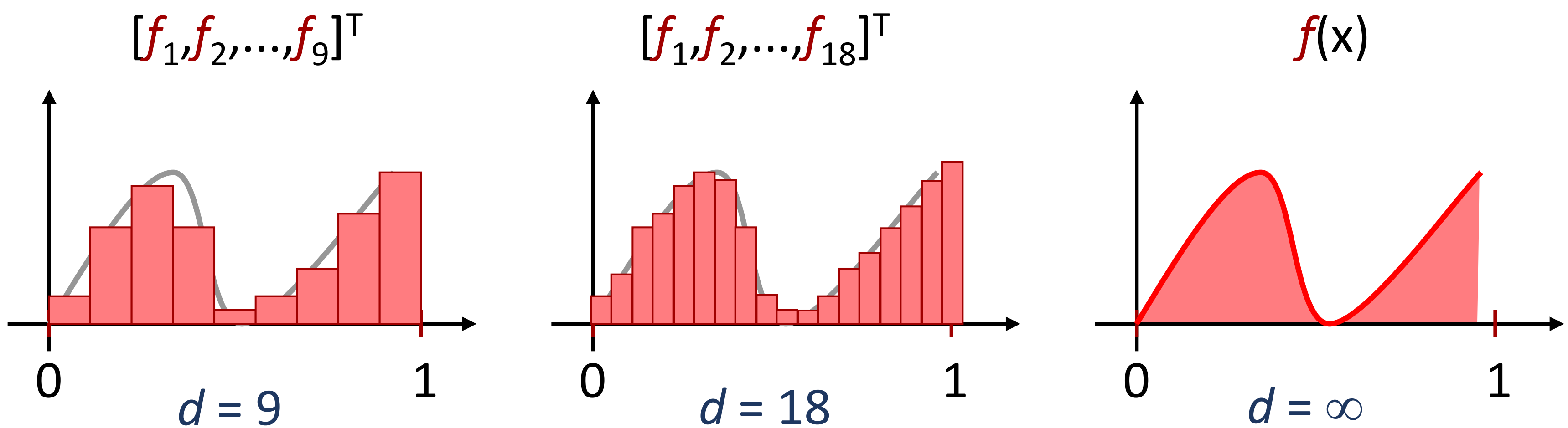
- Space of all functions f: $\mathbb{R} \to \mathbb{R}$
- Space of all smooth C^k functions f: $\mathbb{R} \to \mathbb{R}$
- Space of all functions f: $[0..1]^5 \to \mathbb{R}^8$
- etc...



Function Spaces

Intuition:

- Start with a finite dimensional vector
- Increase sampling density towards infinity
- Real numbers: uncountable amount of dimensions



Dot Product on Function Spaces

Scalar products

- For square-integrable functions *f*, *g*: $\Omega \subseteq \mathbb{R}^n \to \mathbb{R}$, *standard scalar product* defined as:

$$f \cdot g := \int_{\Omega} f(x) g(x) dx$$

- Measures abstract length and “angle” (not in a geometric sense)

Orthogonal functions:

- No mutual influence in linear combinations
- Adding one to the other does not change the value in the other ones direction.

Approximation of Function Spaces

Finite dimensional subspaces:

- Function spaces with infinite dimension are hard to represented on a computer
- For numerical purpose, finite-dimensional subspaces are used to approximate the larger space
- Two basic approaches

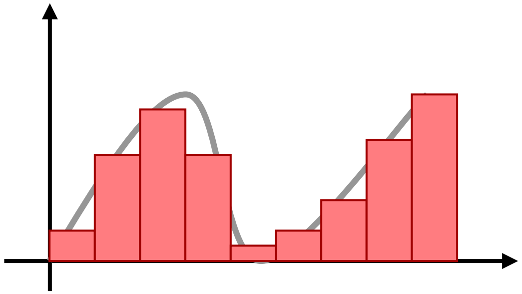
Approximation of Function Spaces

Task:

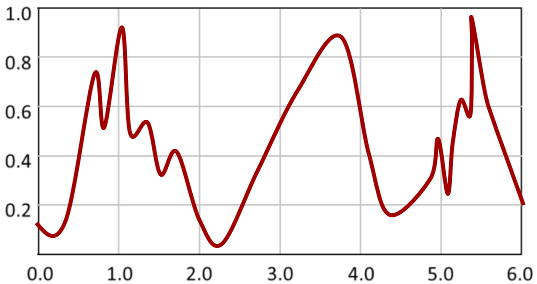
- **Given:** Infinite-dimensional function space V.
- **Task:** Find $f \in V$ with a certain property.

Recipe: “Finite Differences”

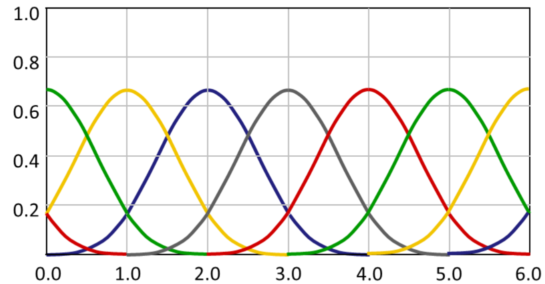
- Sample function f on discrete grid
- Approximate property discretely
  - Derivatives => finite differences
  - Integrals => Finite sums
- Optimization: Find best discrete function



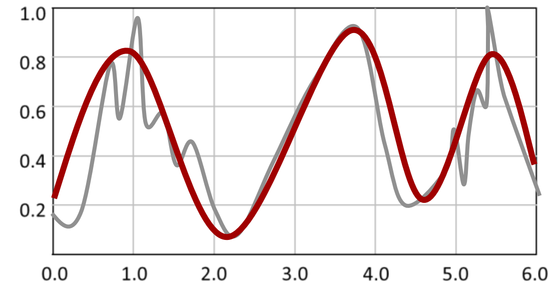
Approximation of Function Spaces



actual solution



function space basis

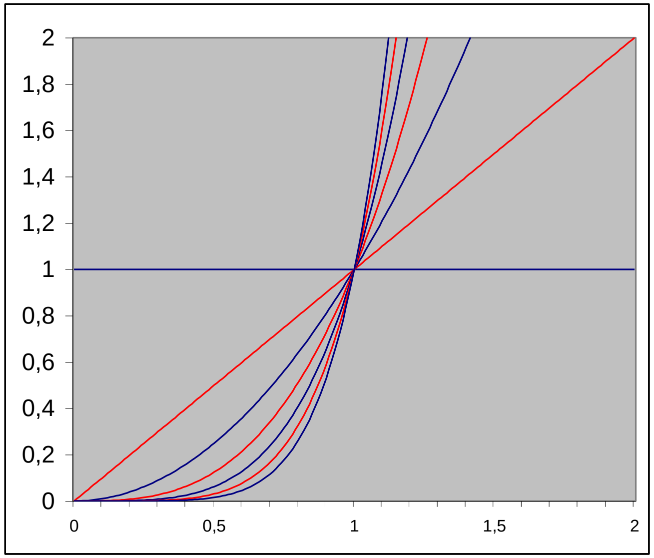


approximate solution

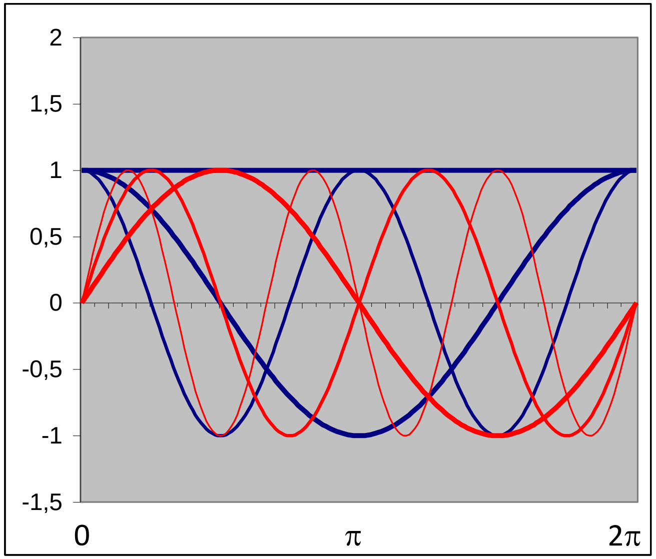
Recipe: "Finite Elements"

- Choose basis functions $b_1, \ldots, b_d \in \mathrm{V}$
- Find $\tilde{f} = \sum_{i=1}^{d} \lambda_i b_i$ that matches the property best
- $\tilde{f}$ is described by $(\lambda_1, \ldots, \lambda_d)$

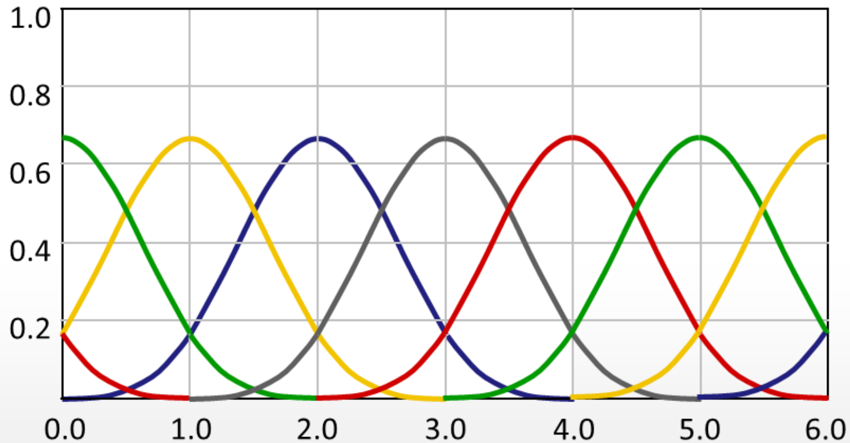
Examples



Monomial basis



Fourier basis



B-spline basis

"Best Match"

Linear combination matches best

- **Solution 1:** Least squares minimization

$$\int_{\mathbb{R}} \left(f(x) - \sum_{i=1}^{n} \lambda_i b_i(x) \right)^2 dx \;\rightarrow min$$

- **Solution 2:** Galerkin method

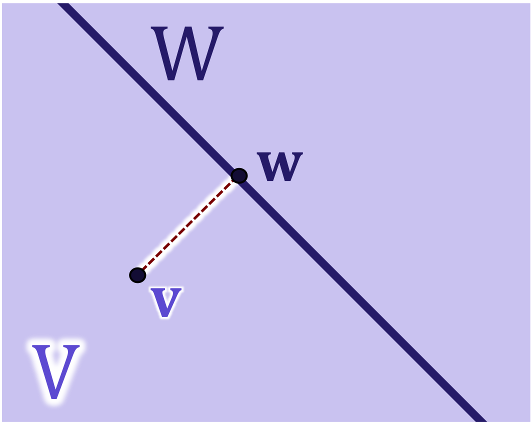
$$\forall i = 1..n: \left\langle f - \sum_{i=1}^{n} \lambda_i b_i \,,\; b_i \right\rangle = 0$$

- Both are equivalent

Optimality Criterion

Given:

- Subspace $\mathrm{W} \subseteq \mathrm{V}$
- An element $\mathbf{v} \in \mathrm{V}$



Then we get:

- $\mathbf{w} \in \mathrm{W}$ minimizes the quadratic error $(\mathrm{w} - \mathbf{v})^2$ (i.e. the Euclidean distance) if and only if:
- the residual $(\mathrm{w} - \mathbf{v})$ is orthogonal to W

Least squares = minimal Euclidean distance

Formal Derivation

Least-squares

$$
\begin{aligned}
\mathrm{E}(f) &= \int_{\mathbb{R}} \left(f(x) - \sum_{i=1}^{n} \lambda_i b_i(x) \right)^2 dx \\
&= \int_{\mathbb{R}} \left(f^2(x) - 2\sum_{i=1}^{n} \lambda_i f(x) b_i(x) + \sum_{i=1}^{n}\sum_{i=1}^{n} \lambda_i \lambda_j b_i(x) b_j(x) \right) \; dx
\end{aligned}
$$

Setting derivatives to zero:

$$
\nabla \mathrm{E}(f) = -2 \begin{pmatrix} \lambda_1 \langle f, b_1 \rangle \\ \vdots \\ \lambda_n \langle f, b_n \rangle \end{pmatrix} + [\lambda_1, \ldots, \lambda_n] \begin{pmatrix} \ddots & \vdots & ⋰ \\ \cdots & \langle b_i(x), b_j(x) \rangle & \cdots \\ ⋰ & \vdots & \ddots \end{pmatrix}
$$

Result:

$$
\forall j = 1..n\colon \; \left\langle \left(f - \sum_{i=1}^{n} \lambda_i b_i \right), b_j \right\rangle = 0
$$

Linear Maps

Linear Maps

A Function

- f: V $\rightarrow$ W between vector spaces V, W

is linear if and only if:

- $\forall v_1, v_2 \in V$: $f(v_1+v_2) = f(v_1) + f(v_2)$
- $\forall v \in V$, $\lambda \in F$: $f(\lambda v) = \lambda f(v)$

Constructing linear mappings:

A linear map is uniquely determined if we specify a mapping value for each basis vector of V.

Matrix Representation

Finite dimensional spaces

- Linear maps can be represented as matrices
- For each basis vector $\mathbf{v}_i$ of V, we specify the mapped vector $\mathbf{w}_i$.
- Then, the map *f* is given by:

$$f(\mathbf{v}) = f\left(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}\right) = v_1\mathbf{w}_1 + \ldots + v_n\mathbf{w}_n$$

Matrix Representation

This can be written as matrix-vector product:

$$f(v) = \begin{pmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_n \\ | & & | \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

The columns are the images of the basis vectors (for which the coordinates of **v** are given)

Affine Maps

Intuition

- Linear maps do not permit translations
- Affine map = linear map + translation

Representation

- $f: \mathbb{R}^n \to \mathbb{R}^m$
- $f(\mathbf{x}) = \mathbf{M}\mathbf{x} + \mathbf{t}$
- Matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, vector $\mathbf{t} \in \mathbb{R}^m$

Affine Maps

Formal characterization

f is affine if and only if:

Given weights α_i with

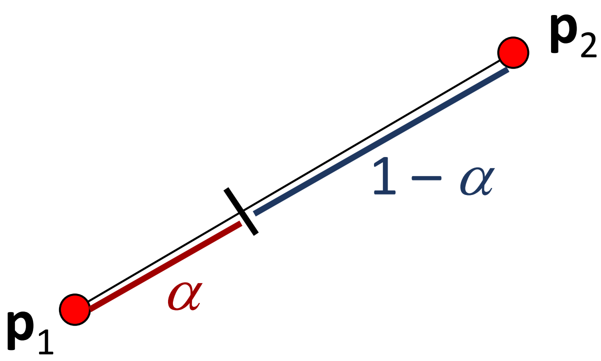
$$\sum_{k=1}^{n} \alpha_i = 1$$

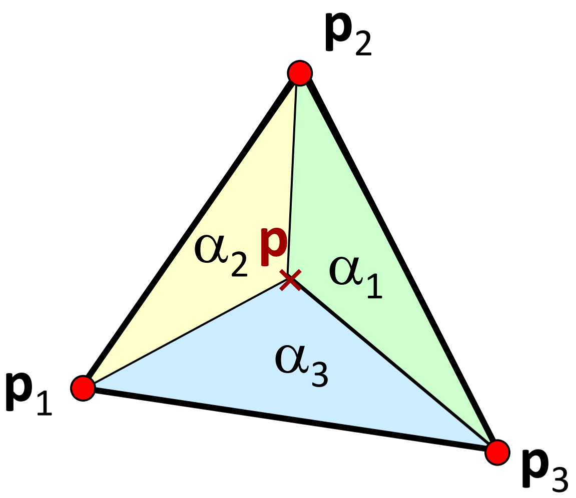
we always have:

$$f\left(x_1, \ldots, \sum_{k=1}^{n} \alpha_i x_i^{(k)}, \ldots, x_m\right) = \sum_{k=1}^{n} \alpha_i f\left(x_1, \ldots, x_i^{(k)}, \ldots, x_m\right)$$

Geometric Intuition

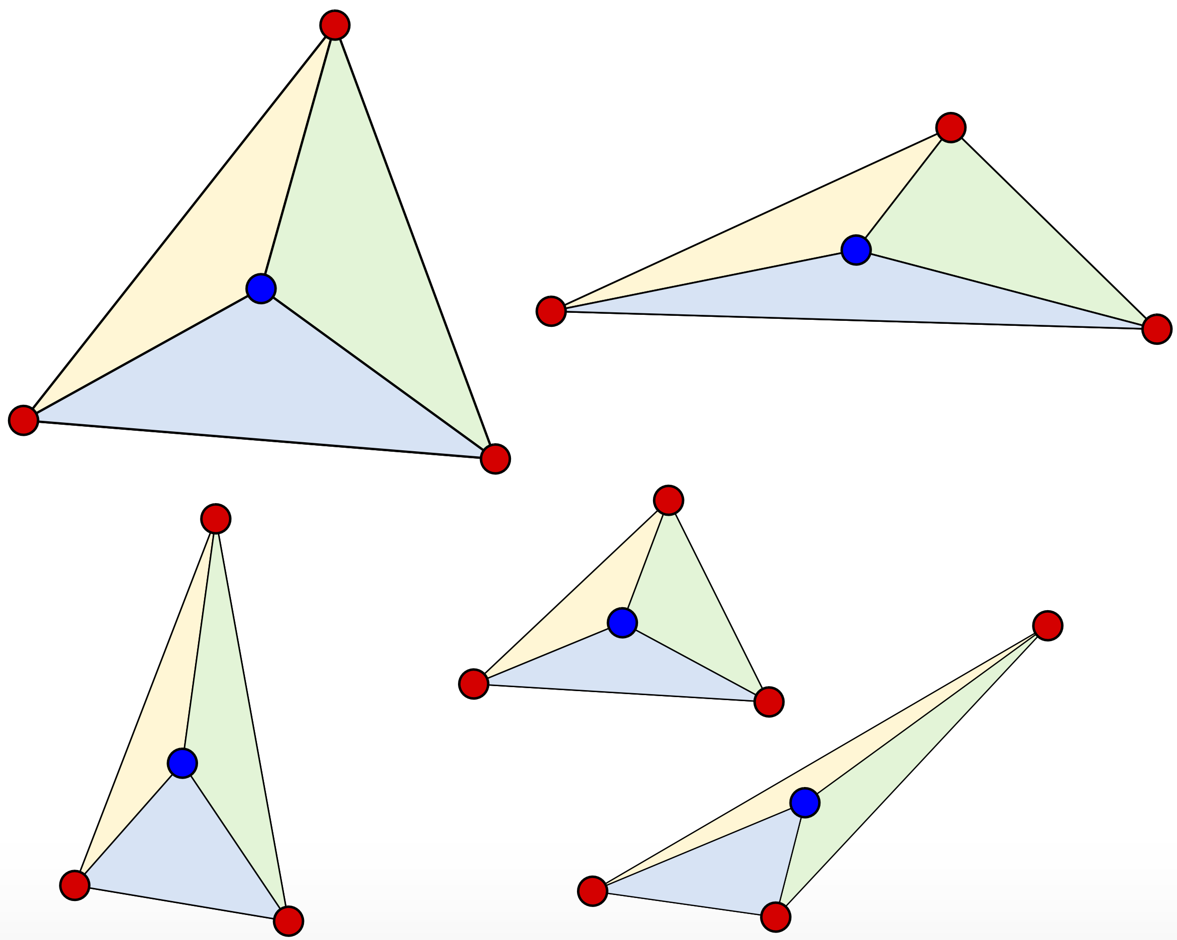
Weighted averages of points are preserved:





Geometric Intuition

Weighted averages of points are preserved:



Linear Systems of Equations

Problem: Invert an affine map

- Given: $\mathbf{Mx} = \mathbf{b}$
- We know $\mathbf{M}$, $\mathbf{b}$
- Looking for $\mathbf{x}$

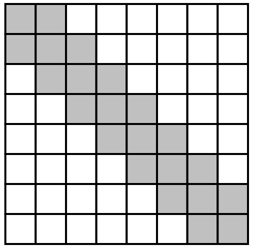
Solution

- Set of solutions: always an *affine subspace* of $\mathbb{R}^n$, or the empty set.
  - Point, line, plane, hyperplane...
- Innumerous algorithms for solving linear systems

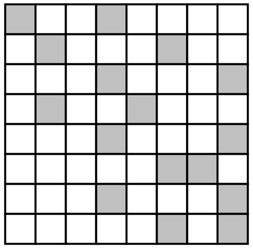
Solvers for Linear Systems

Algorithms for solving linear systems of equations:

- Gaussian elimination: $O(n^3)$ operations for $n \times n$ matrices
- We can do better, in particular for special cases:
  - **Band matrices:**
    constant bandwidth



  - **Sparse matrices:**
    constant number of non-zero entries per row
    - Store only non-zero entries
    - Instead of (3.5, 0, 0, 0, 7, 0, 0), store [(1:3.5), (5:7)]



Solvers for Linear Systems

Algorithms for solving linear systems of *n* equations:

- Band matrices, O(1) bandwidth:
  - Modified O(n) elimination algorithm.
- Iterative Gauss-Seidel solver
  - Converges for diagonally dominant matrices
  - Typically: O(n) iterations, each costs O(n) for a sparse matrix.
- Conjugate Gradient solver
  - Only symmetric, positive definite matrices
  - Guaranteed: O(n) iterations
  - Typically good solution after O($\sqrt{n}$) iterations.

More details on iterative solvers: *J. R. Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, 1994.*

Eigenvectors & Eigenvalues

Definition:

- Linear map $\mathbf{M}$, non-zero vector $\mathbf{x}$ with

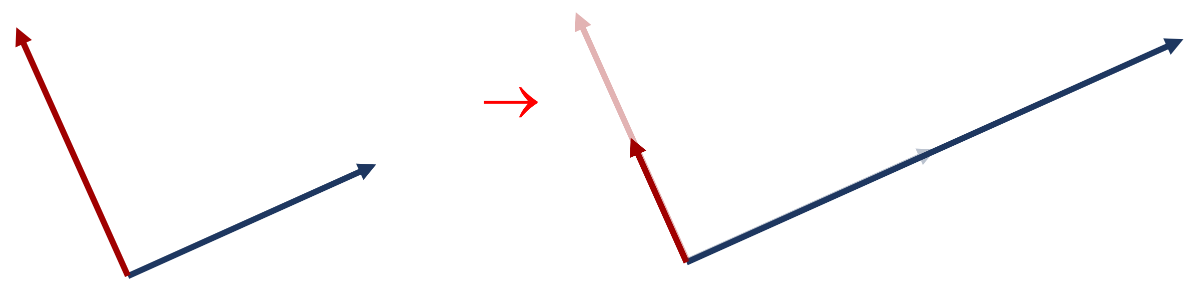
  $\mathbf{M}\mathbf{x} = \lambda\mathbf{x}$

- λ an is *eigenvalue* of $\mathbf{M}$
- $\mathbf{x}$ is the corresponding *eigenvector*.

Example

Intuition:

- In the direction of an eigenvector, the linear map acts like a scaling



- Example: two eigenvalues (0.5 and 2)
- Two eigenvectors
- Standard basis contains no eigenvectors

Eigenvectors & Eigenvalues

Diagonalization:

In case an $n \times n$ matrix **M** has n linear independent eigenvectors, we can *diagonalize* **M** by transforming to this coordinate system: $\mathbf{M} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}$.

Spectral Theorem

Spectral Theorem:

Given: symmetric $n \times n$ matrix $\mathbf{M}$ of real numbers ($\mathbf{M} = \mathbf{M}^T$)

It follows: There exists an *orthogonal* set of n eigenvectors.

This implies:

Every (real) symmetric matrix can be *diagonalized*:

$\mathbf{M} = \mathbf{T}\mathbf{D}\mathbf{T}^T$ with an orthogonal matrix $\mathbf{T}$, diagonal matrix $\mathbf{D}$.

Computation

Simple algorithm

- "Power iteration" for symmetric matrices
- Computes largest eigenvalue even for large matrices
- Algorithm:
  - Start with a random vector (maybe multiple tries)
  - Repeatedly multiply with matrix
  - Normalize vector after each step
  - Repeat until ration before / after normalization converges (this is the eigenvalue)
- Intuition:
  - Largest eigenvalue = "dominant" component/direction

Powers of Matrices

What happens:

- A symmetric matrix can be written as:

$$\mathbf{M} = \mathbf{T}\mathbf{D}\mathbf{T}^{\mathrm{T}} = \mathbf{T}\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}\mathbf{T}^{\mathrm{T}}$$

- Taking it to the *k*-th power yields:

$$\mathbf{M}^k = \mathbf{T}\mathbf{D}\mathbf{T}^{\mathrm{T}}\mathbf{T}\mathbf{D}\mathbf{T}^{\mathrm{T}}\ldots\mathbf{T}\mathbf{D}\mathbf{T}^{\mathrm{T}} = \mathbf{T}\mathbf{D}^k\mathbf{T}^{\mathrm{T}} = \mathbf{T}\begin{pmatrix} \lambda_1{}^k & & \\ & \ddots & \\ & & \lambda_n{}^k \end{pmatrix}\mathbf{T}^{\mathrm{T}}$$

- Bottom line: Eigenvalue analysis key to understanding powers of matrices.

Improvements

Improvements to the power method:

- Find smallest? – use inverse matrix.
- Find all (for a symmetric matrix)? – run repeatedly, orthogonalize current estimate to already known eigenvectors in each iteration (Gram Schmidt)
- How long does it take? – ratio to next smaller eigenvalue, gap increases exponentially.

There are more sophisticated algorithms based on this idea.

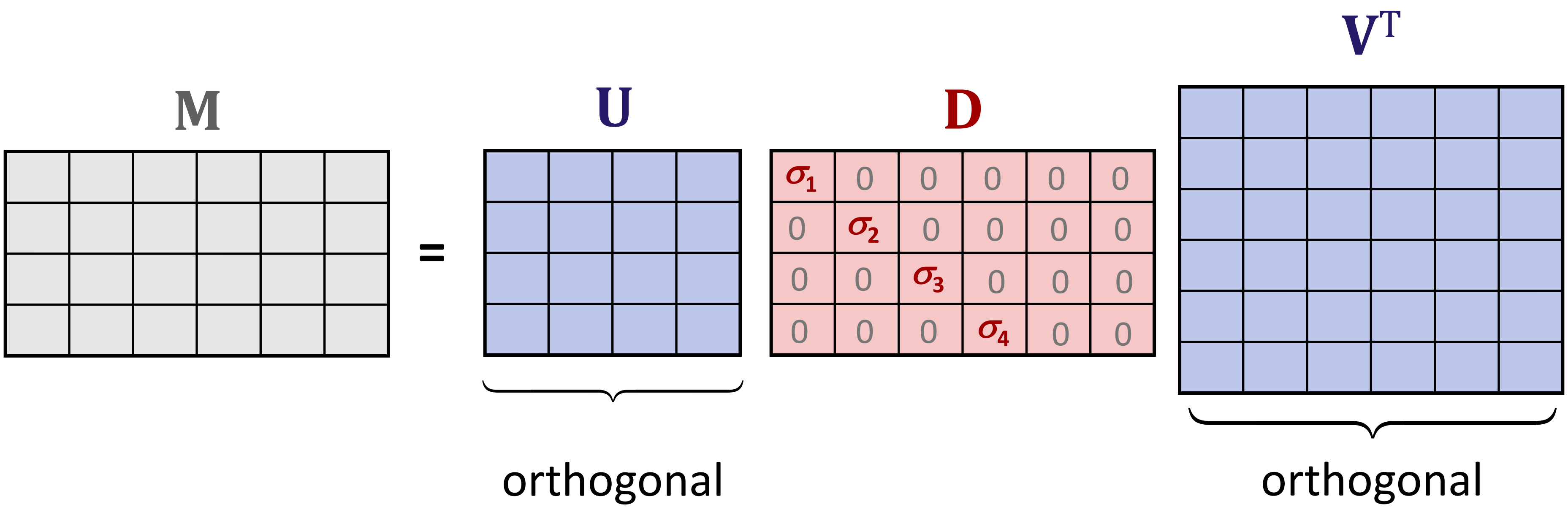
Generalization: SVD

Singular value decomposition:

- Let **M** be an arbitrary real matrix (may be rectangular)
- Then **M** can be written as:
  - $\mathbf{M} = \mathbf{U}\ \mathbf{D}\ \mathbf{V}^T$
  - The matrices **U**, **V** are orthogonal
  - **D** is a diagonal matrix (might contain zeros)
  - The diagonal entries are called *singular values*.
- **U** and **V** are usually different
- Diagonalizable matrices:
  - **U** = **V**
  - Singular values = eigenvalues

Singular Value Decomposition

Singular value decomposition



Singular Value Decomposition

Singular value decomposition

- Can be used to solve linear systems of equations
- For full rank, square $\mathbf{M}$:

$$\mathbf{M} = \mathbf{U}\ \mathbf{D}\ \mathbf{V}^{T}$$

$$\Rightarrow \mathbf{M}^{-1} = (\mathbf{U}\ \mathbf{D}\ \mathbf{V}^{T})^{-1} = (\mathbf{V}^{T})^{-1}\ \mathbf{D}^{-1}\ (\mathbf{U}^{-1}) = \mathbf{V}\ \mathbf{D}^{-1}\ \mathbf{U}^{T}$$

- Good numerical properties (numerically stable)
- More expensive than iterative solvers
- The OpenCV library provides a very good implementation of the SVD

Example:

Linear Inverse Problems

Inverse Problems

Settings

- A (physical) process f takes place
- It transforms the original input **x** into an output **b**
- Task: recover **x** from **b**

Examples:

- 3D structure from photographs
- Tomography: values from line integrals
- 3D geometry from a noisy 3D scan

Linear Inverse Problems

Assumption: f is linear and finite dimensional

$$f(\mathbf{x}) = \mathbf{b} \;\Rightarrow\; \mathbf{M}_f\mathbf{x} = \mathbf{b}$$

Inversion of f is said to be an ill-posed problem, if one of the following three conditions hold:

- There is no solution
- There is more than one solution
- There is exactly one solution, but the SVD contains very small singular values.

Ill posed Problems

Ratio: Small singular values amplify errors

- Assume inexact input
  - Measurement noise
  - Numerical noise
- Reminder: $\mathbf{M}^{-1} = \mathbf{V}\ \mathbf{D}^{-1}\ \mathbf{U}^{T}$

does not hurt (orthogonal)

does not hurt (orthogonal)

this one is decisive

- Orthogonal transforms preserve norm of $\mathbf{x}$, so $\mathbf{V}$ and $\mathbf{U}$ do not cause problems

Ill posed Problems

Ratio: Small singular values amplify errors

- Reminder: $\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} = (\mathbf{V}\,\mathbf{D}^{-1}\,\mathbf{U}^{T})\mathbf{b}$
- Say **D** looks like that:

$$\mathbf{D} := \begin{pmatrix} 2.5 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 \\ 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.000000001 \end{pmatrix}$$

- Any input noise in **b** in the direction of the fourth right singular vector will be amplified by 10^9.
- If our measurement precision is less than that, the result will be unusable.
- Does *not* depend on *how* we invert the matrix.
- Condition number: $\sigma_{max} / \sigma_{min}$

Ill Posed Problems

Two problems:

(1) Mapping destroys information
- goes below noise level
- cannot be recovered by any means

$$\mathbf{D} := \begin{pmatrix} 2.5 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 \\ 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.000000001 \end{pmatrix}$$

(2) Inverse mapping amplifies noise
- yields garbage solution
- even remaining information not recovered
- extremely large random solutions are obtained

We can do something about problem #2

Regularization

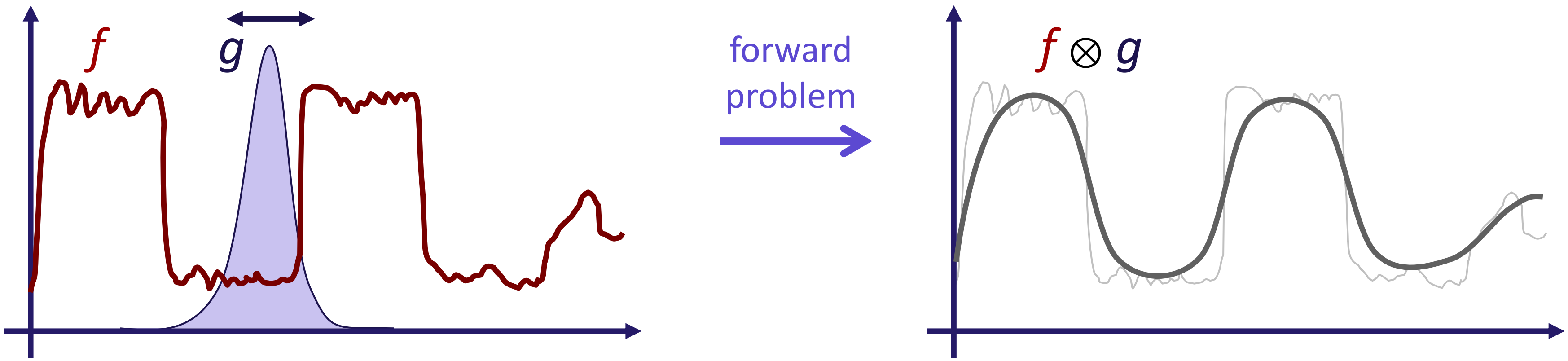
Regularization

- Avoiding destructive noise caused by inversion
  - Various techniques
  - Goal: ignore the misleading information

Approaches

- Subspace inversion: Ignore subspace with small singular values
  - Needs an SVD, risk of “ringing”
- Additional assumptions:
  - smoothness (or something similar)
  - make compound problem (f^{-1} + assumptions) well posed

Illustration of the Problem



original function

smoothed function

Illustration of the Problem

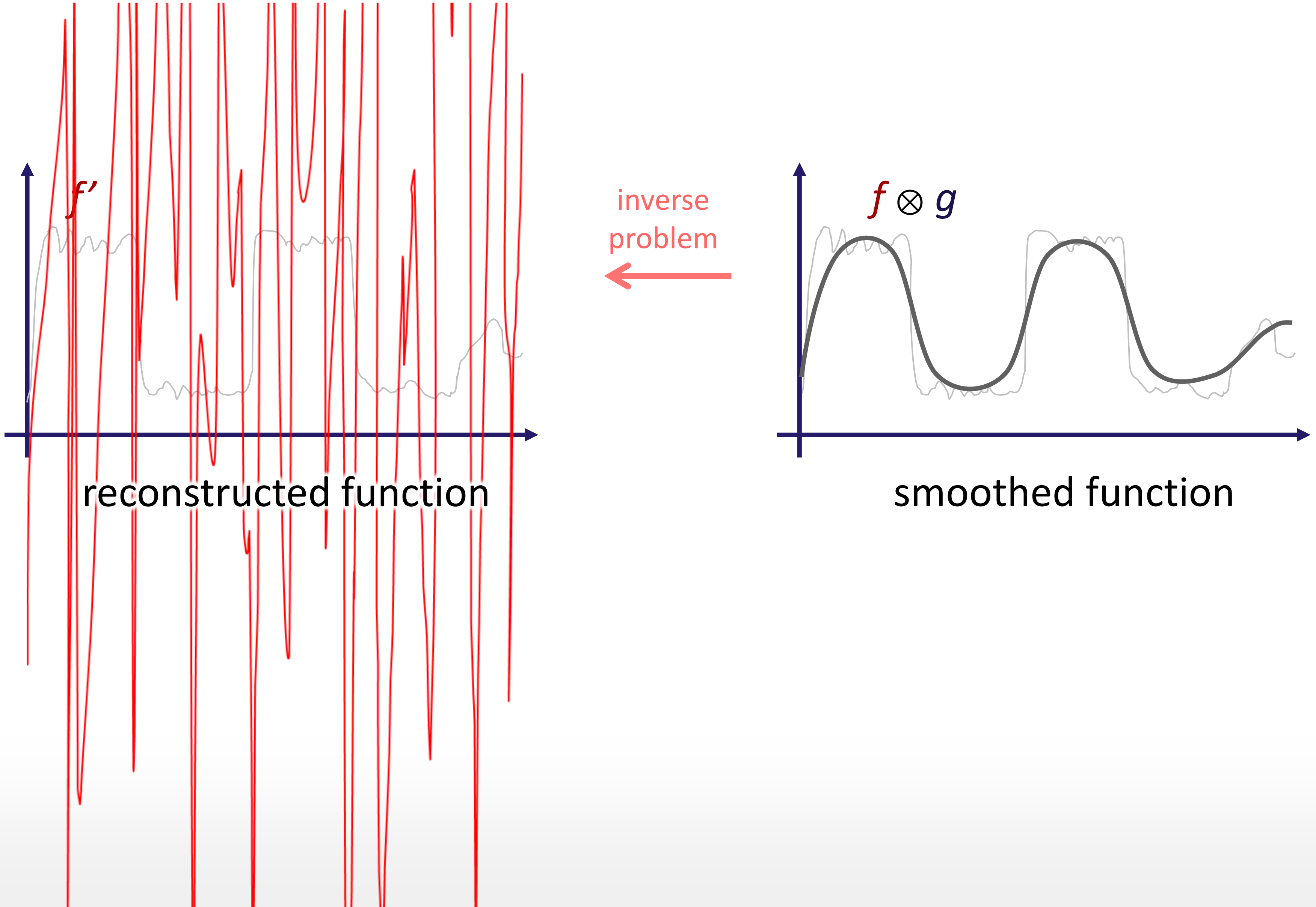
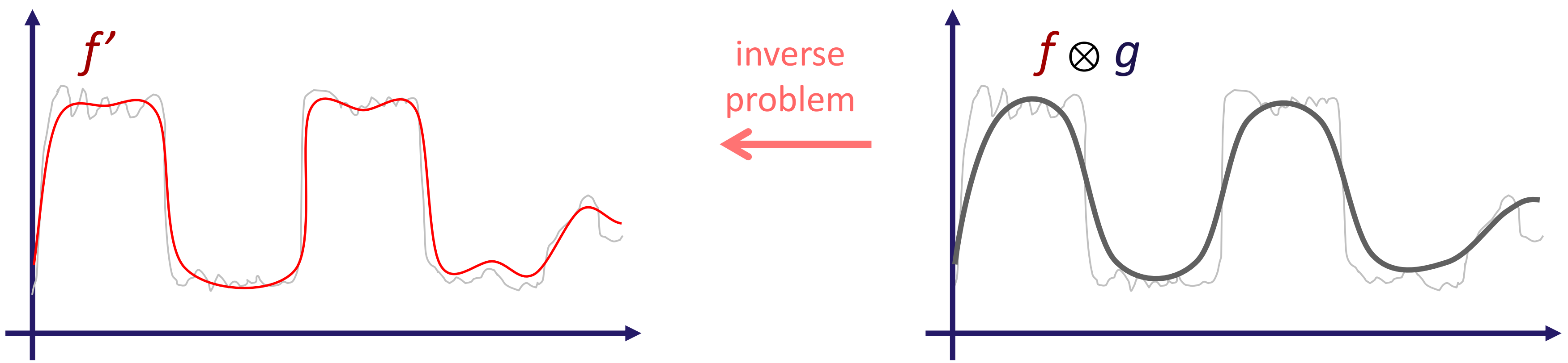


Illustration of the Problem



regularized
reconstructed function

smoothed function

Quadratic Forms

Multivariate Polynomials

A *multi-variate* polynomial of total degree *d*:

- A function $f\colon \mathbb{R}^n \to \mathbb{R}$, $\mathbf{x} \to f(\mathbf{x})$
- f is a polynomial in the components of $\mathbf{x}$
- Any 1D direction $f(\mathbf{s} + t\mathbf{r})$ is a polynomial of maximum degree d in t.

Examples:

- $f(\mathrm{x}, \mathrm{y}) := \mathrm{x} + \mathrm{xy} + \mathrm{y}$ is of total degree 2. In diagonal direction, we obtain $f(t[1/\sqrt{2}, 1/\sqrt{2}]^{\mathrm{T}}) = t^2$.
- $f(\mathrm{x}, \mathrm{y}) := c_{20}\mathrm{x}^2 + c_{02}\mathrm{y}^2 + c_{11}\mathrm{xy} + c_{10}\mathrm{x} + c_{01}\mathrm{y} + c_{00}$ is a quadratic polynomial in two variables

Quadratic Polynomials

In general, any quadratic polynomial in *n* variables can be written as:

- $\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$
- $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{b}$ is an n-dim. vector, c is a number
- Matrix $\mathbf{A}$ can always be chosen to be symmetric
- If it isn't, we can substitute by $0.5 \cdot (\mathbf{A} + \mathbf{A}^T)$, not changing the polynomial

Example

Example:

$$f\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \mathbf{x}$$

$$= [x\ y] \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = [x\ y] \begin{pmatrix} 1x & 2y \\ 3x & 4y \end{pmatrix}$$

$$= x1x + x2y + y3x + y4y$$

$$= 1x^2 + (2+3)xy + 4y^2$$

$$= 1x^2 + (2.5+2.5)xy + 4y^2$$

$$= \mathbf{x}^{\mathrm{T}} \frac{1}{2} \left[\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \right] \mathbf{x} = \mathbf{x}^{\mathrm{T}} \begin{pmatrix} 1 & 2.5 \\ 2.5 & 4 \end{pmatrix} \mathbf{x}$$

Quadratic Polynomials

Specifying quadratic polynomials:

- $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x} + c$
- **b** shifts the function in space (if **A** has full rank):

$$(x-\mu)^{\mathrm{T}}\mathbf{A}(x-\mu)+c$$

$$= x^{\mathrm{T}}\mathbf{A}x - \mu^{\mathrm{T}}\mathbf{A}x - x^{\mathrm{T}}\mathbf{A}\mu + \mu\cdot\mu + c$$

$$\overset{\text{(A sym.)}}{=} x^{\mathrm{T}}\mathbf{A}x - \underbrace{(2\mathbf{A}\mu)}_{=\,\mathbf{b}}\mathbf{x} + \mu\cdot\mu + c$$

- c is an additive constant

Some Properties

Important properties

- Multivariate polynomials form a vector space
- We can add them component-wise:

$$
\begin{array}{rl}
 & 2x^2 + 3y^2 + 4xy + 2x + 2y + 4 \\
+ & 3x^2 + 2y^2 + 1xy + 5x + 5y + 5 \\
\hline\hline
= & 5x^2 + 5y^2 + 5xy + 7x + 7y + 9
\end{array}
$$

- In vector notation:

$$
\begin{aligned}
& \mathbf{x}^{\mathrm{T}}\mathbf{A}_1\mathbf{x} + \mathbf{b}_1^{\mathrm{T}}\mathbf{x} + c_1 \\
+\ & \lambda(\mathbf{x}^{\mathrm{T}}\mathbf{A}_2\mathbf{x} + \mathbf{b}_2^{\mathrm{T}}\mathbf{x} + c_2) \\
=\ & \mathbf{x}^{\mathrm{T}}(\mathbf{A}_1 + \lambda\mathbf{A}_2)\mathbf{x} + (\mathbf{b}_1 + \lambda\mathbf{b}_2)^{\mathrm{T}}\mathbf{x} + (c_1 + \lambda c_2)
\end{aligned}
$$

Quadratic Polynomials

Quadrics

- Zero level set of a quadratic polynomial: “quadric”
- Shape depends on eigenvalues of $\mathbf{A}$
- $\mathbf{b}$ shifts the object in space
- c sets the level

Shapes of Quadrics

Shape analysis:

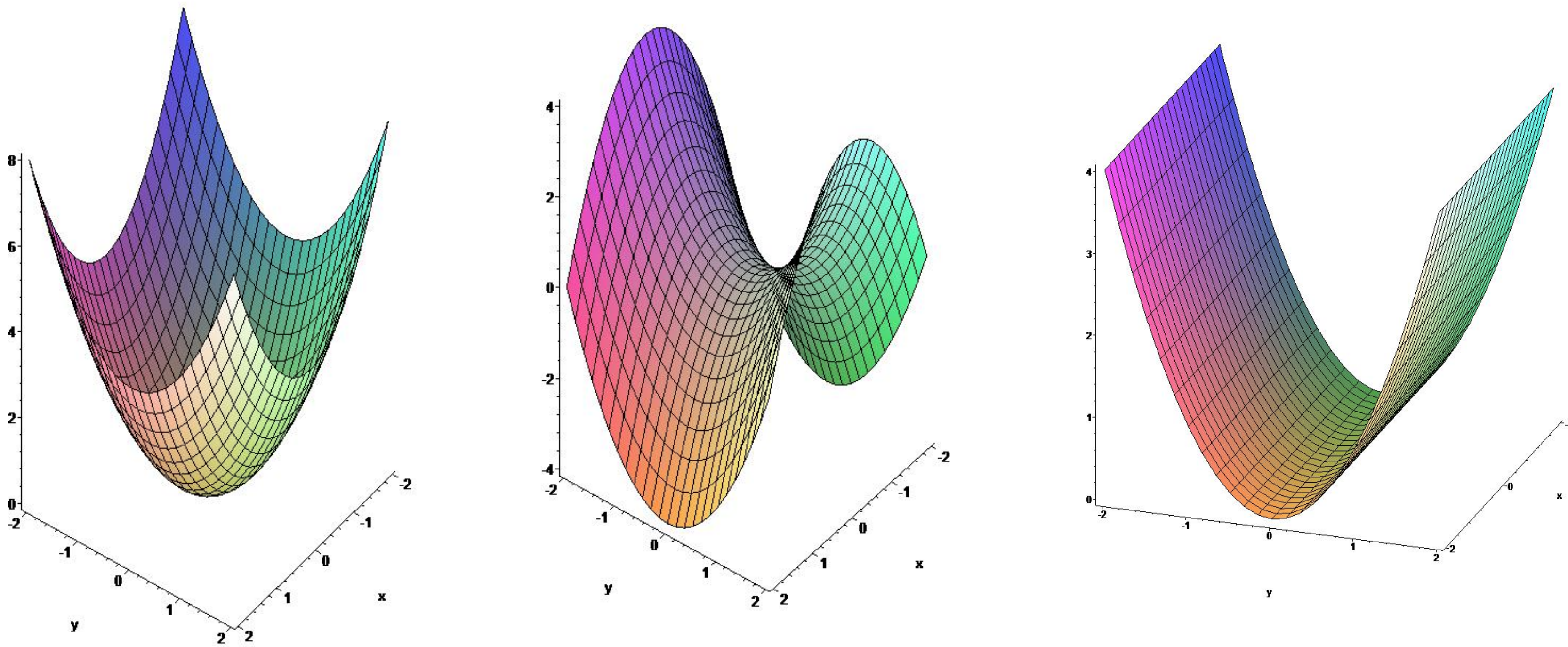
- $\mathbf{A}$ is symmetric
- $\mathbf{A}$ can be *diagonalized* with orthogonal *eigenvectors*

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} = x^{\mathrm{T}}\left[\mathbf{Q}^{\mathrm{T}}\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{pmatrix}\mathbf{Q}\right]x$$

$$= (\mathbf{Q}x)^{\mathrm{T}}\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{pmatrix}(\mathbf{Q}x)$$

- $\mathbf{Q}$ contains the principal axis of the quadric
- The eigenvalues determine the quadratic growth (up, down, speed of growth)

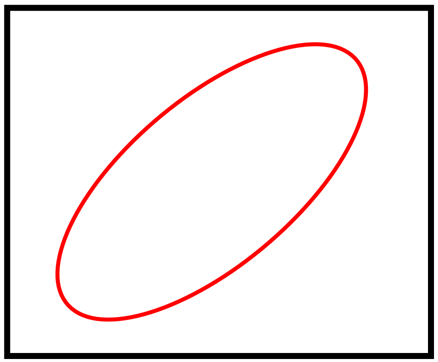
Shapes of Quadratic Polynomials



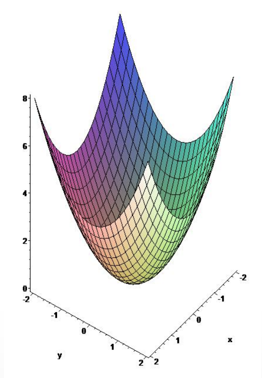
$\lambda_1 = 1, \lambda_2 = 1$ $\lambda_1 = 1, \lambda_2 = -1$ $\lambda_1 = 1, \lambda_2 = 0$

The Iso-Lines: Quadrics

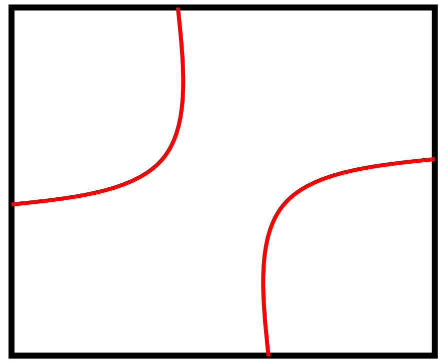
elliptic



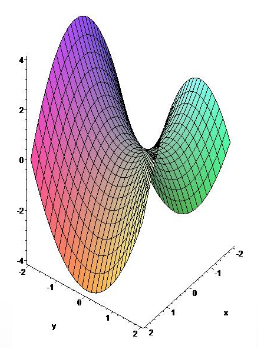
$\lambda_1 > 0, \lambda_2 > 0$



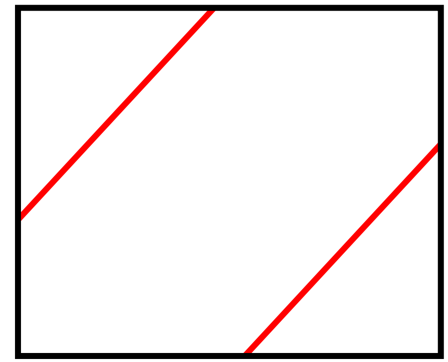
hyperbolic



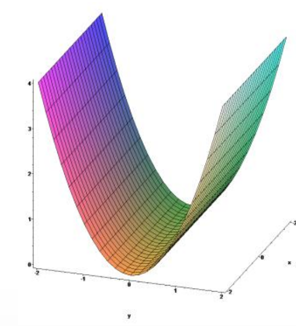
$\lambda_1 < 0, \lambda_2 > 0$



degenerate case



$\lambda_1 = 0, \lambda_2 \neq 0$



Quadratic Optimization

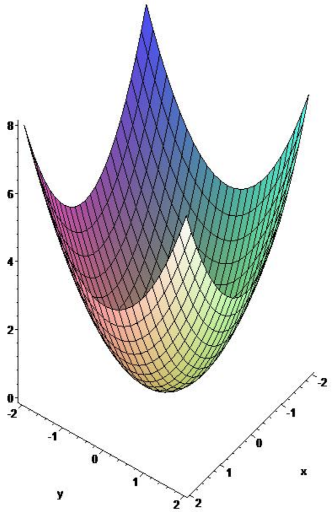
Quadratic Optimization

- Minimize quadratic objective function

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x} + c$$

- Required: $\mathbf{A} > 0$ (only positive eigenvalues)
  - It’s a paraboloid with a unique minimum
  - The vertex (critical point) can be determined by simply solving a linear system
- Necessary and sufficient condition

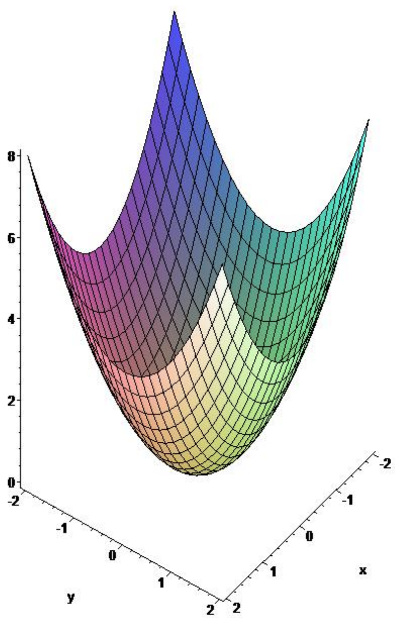
$$2\mathbf{A}\mathbf{x} = -\mathbf{b}$$



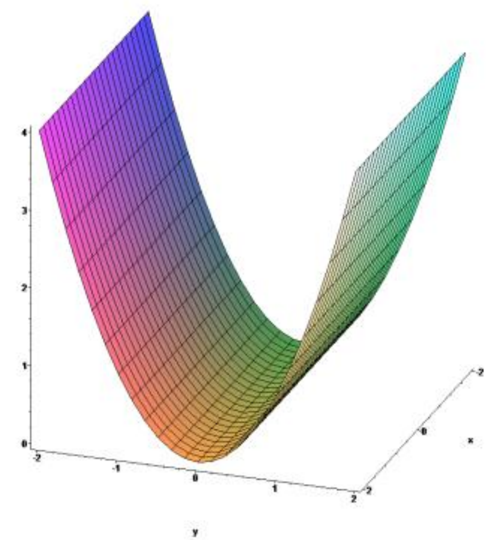
Condition Number

How stable is the solution?

- Depends on Matrix $\boldsymbol{A}$



good

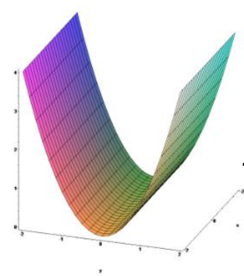


bad

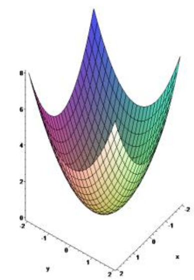
Regularization

Regularization

- Sums of positive semi-definite matrices are positive semi-definite
- Add regularizing quadric
  - "Fill in the valleys"
  - Bias in the solution



$$\mathbf{A} \quad + \quad \lambda\mathbf{I}$$



Example

- Original: $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x} + c$
- Regularized: $\mathbf{x}^{\mathrm{T}}(\mathbf{A} + \lambda\mathbf{I})\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x} + c$

Rayleigh Quotient

Relation to eigenvalues:

- Min/max eigenvalues of a symmetric **A** expressed as constraint quadratic optimization:

$$\lambda_{\min} = \min \frac{\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\mathbf{x}} = \min_{\|x\|=1} \left(\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}\right) \qquad \lambda_{\max} = \max \frac{\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\mathbf{x}} = \max_{\|x\|=1} \left(\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}\right)$$

- The other way round – eigenvalues solve a certain type of constrained, (non-convex) optimization problem.

Coordinate Transformations

One more interesting property:

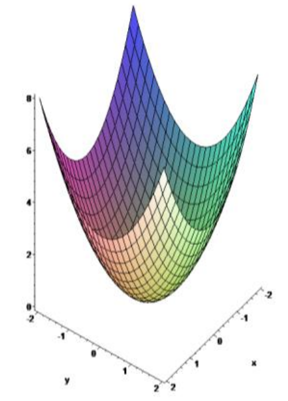
- Given a positive definite symmetric ("SPD") matrix **M** (all eigenvalues positive)
- Such a matrix can always be written as square of another matrix:

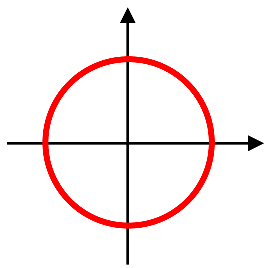
$$\mathbf{M} = \mathbf{TDT}^{\mathrm{T}} = \left(T\sqrt{D}\right)\left(\sqrt{D}^{T} T^{T}\right) = \left(T\sqrt{D}\right)\left(T\sqrt{D}\right)^{T} = \left(T\sqrt{D}\right)^{2}$$

$$\sqrt{D} = \begin{pmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \\ & & \sqrt{\lambda_n} \end{pmatrix}$$

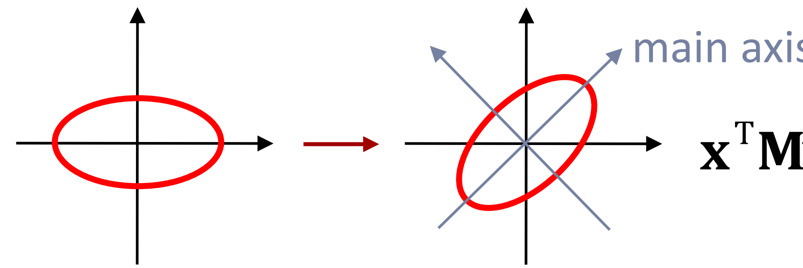
SPD Quadrics

$\mathbf{x}^T\mathbf{x}$





Identity $\mathbf{I}$



$\mathbf{x}^T\mathbf{M}\mathbf{x}$

$$\mathbf{M} = \mathbf{T}\mathbf{D}\mathbf{T}^T = \left(T\sqrt{D}\right)^2$$

Interpretation:

- Start with a unit positive quadric $\mathbf{x}^T\mathbf{x}$.
- Scale the main axis (diagonal of D)
- Rotate to a different coordinate system (columns of **T**)
- Recovering main axis from **M**: Compute eigensystem ("principal component analysis")

Why should I care?

What are quadrics good for?

- *log-probability* of Gaussian models
- Estimation in Gaussian probabilistic models...
  - ...is quadratic optimization.
  - ...is solving of linear systems of equations.
- Quadratic optimization
  - easy to use & solve
  - feasible :-)
- Approximate more complex models locally

Gaussian normal distribution

$$p_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Constructing Bases

How to construct a basis?

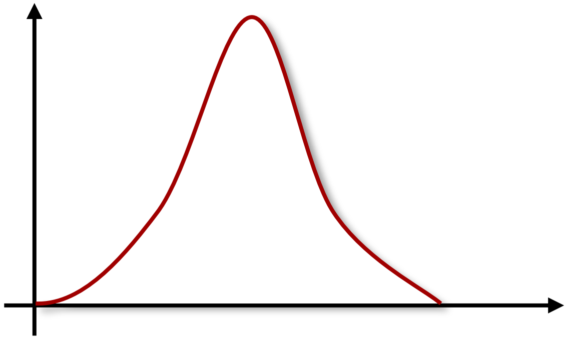
Goal (of much of this whole lecture):

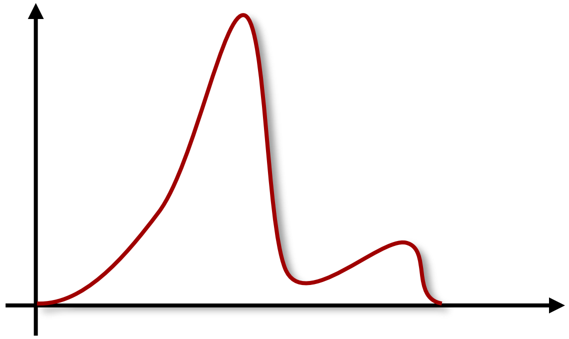
- Build a good basis for a problem

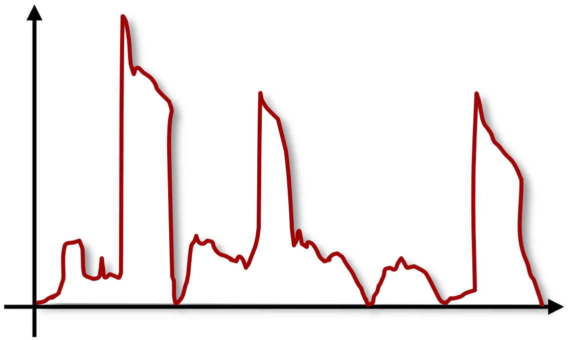
Ingredients:

- Basis functions
- Placement in space
- Semantics

Basis Function



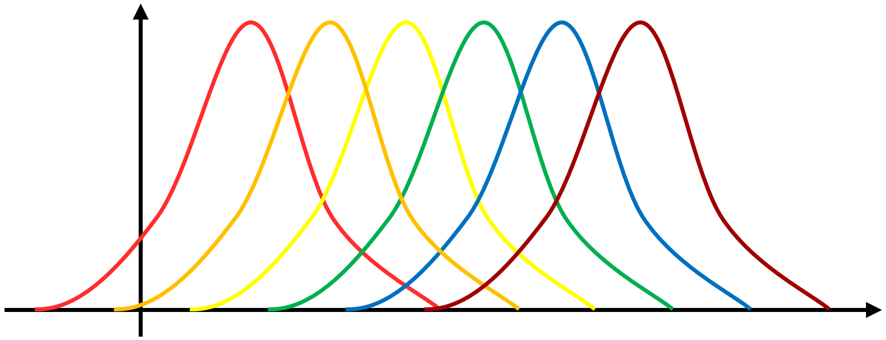


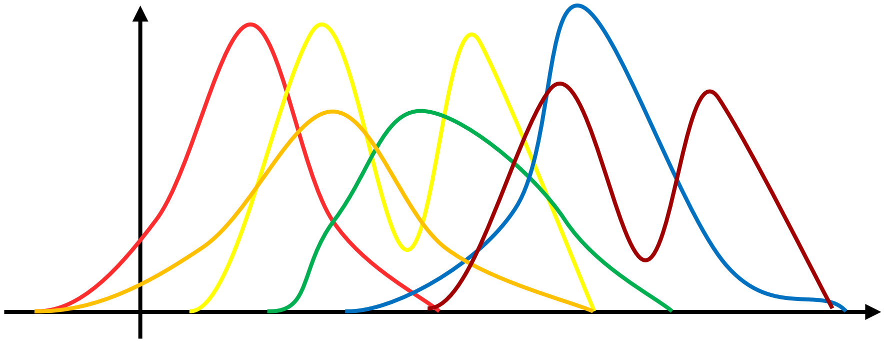


Shape of individual functions:

- Smoothness
- Symmetry
- Support

Ensembles of Functions

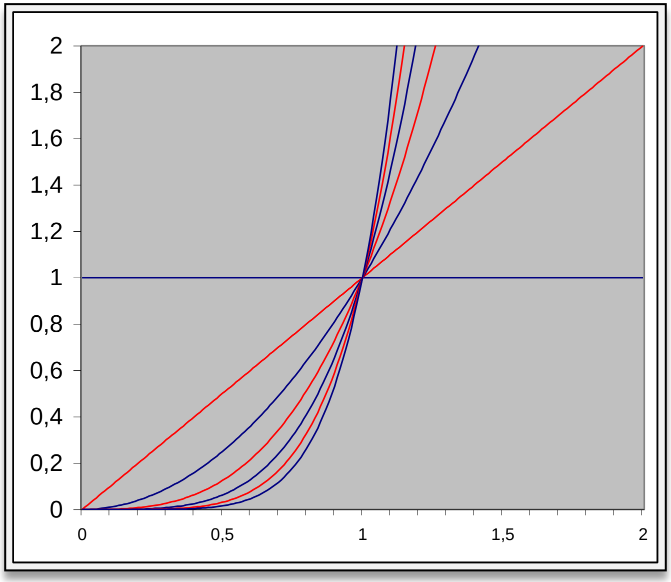




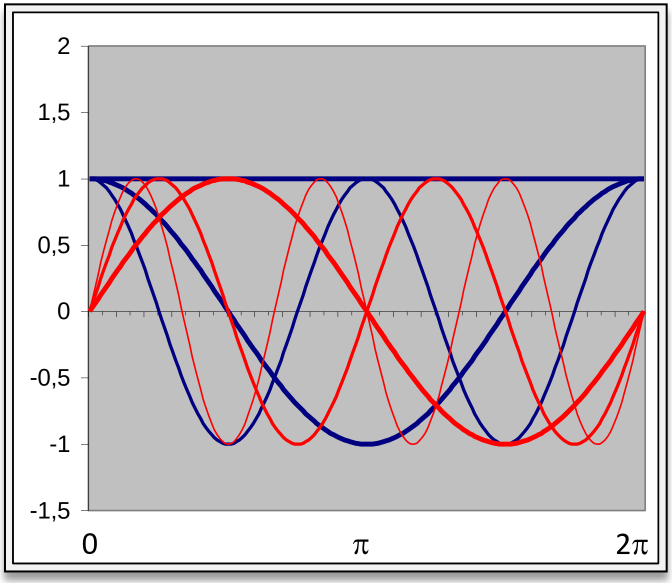
Basis function sets:

- Stationary
  - Same function repeating? (dilations)
  - Varying shapes

Ensembles of Functions



Monomial basis



Fourier basis (orthogonal)

Basis function sets:

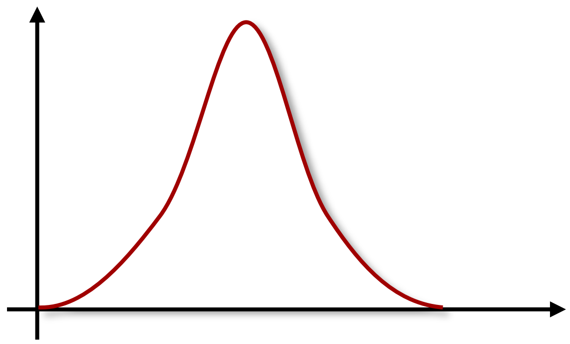
- Orthogonality?
  - Basis functions span independent directions?
  - Advantages: easier, faster, more stable computations
  - Disadvantages: strong constraint on function shape

Example: Radial Basis functions

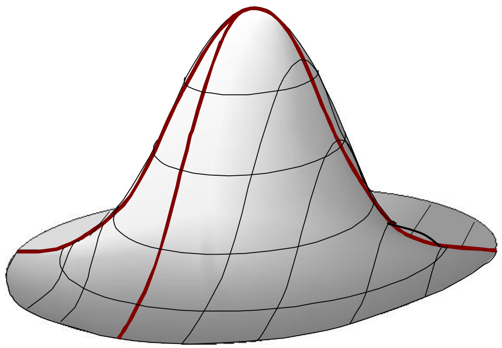
Radial basis function:

- Pick one template function
- Symmetric around “center” point

Instantiate by placing in domain

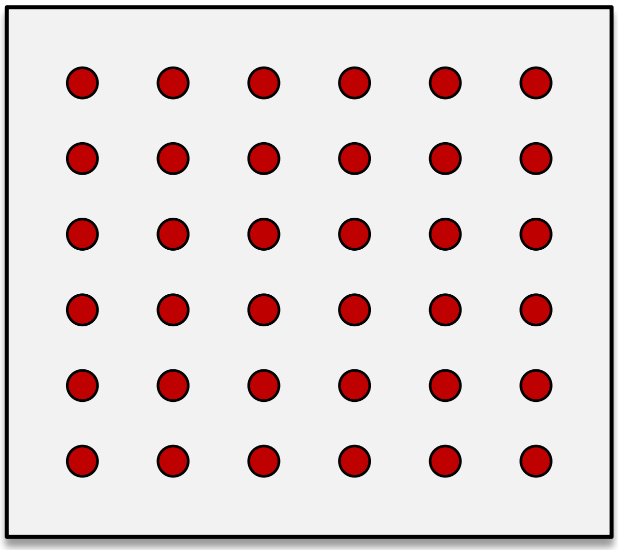


1D

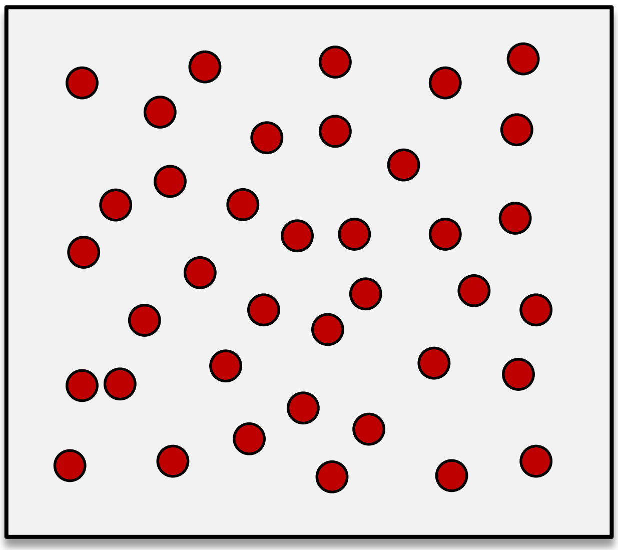


2D

Placement



Regular grids

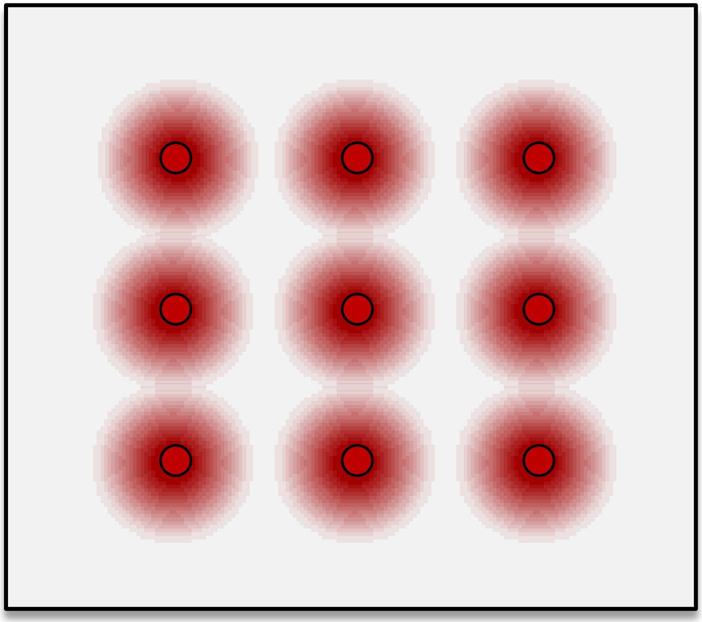


Irregular

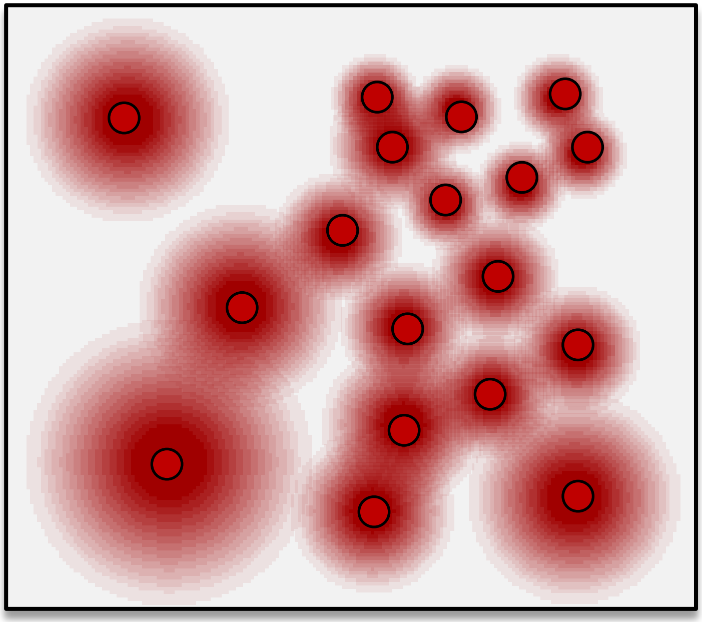
Context:

- Stationary functions, or very similar shape
- How to instantiate?

Placement



Regular grids

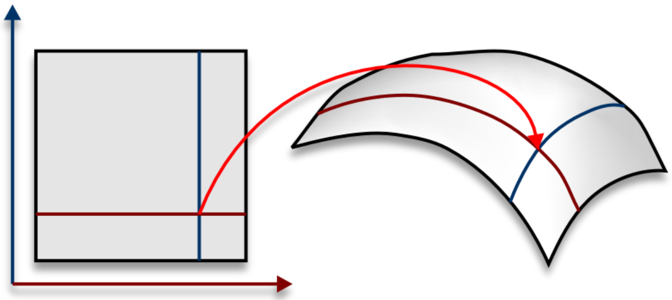


Irregular
(w/scaling)

Semantics

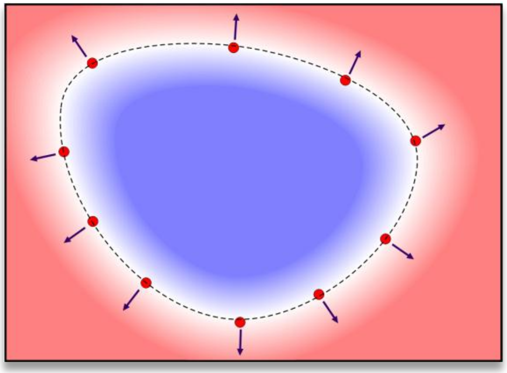
Explicit representations

- Height field
- Parametric surface
- Function value corresponds to actual geometry



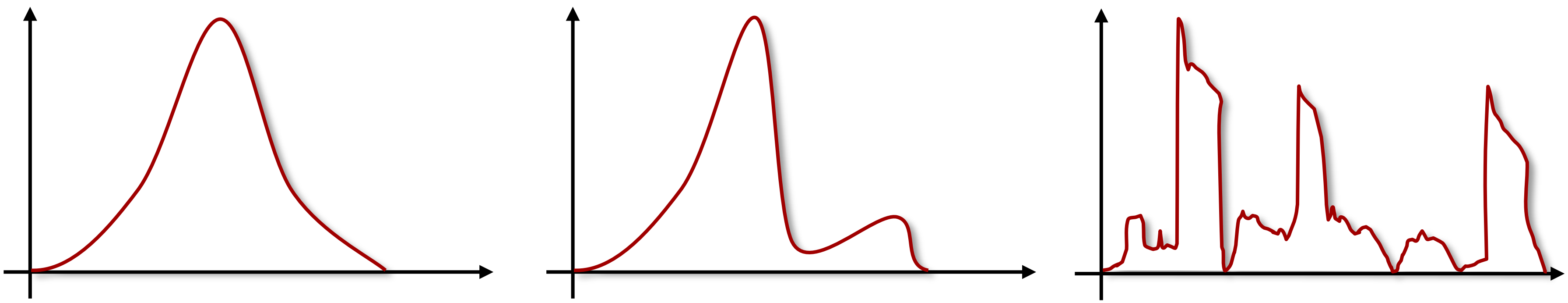
Implicit representation

- Scalar fields
- Zero crossings correspond to actual geometry



How to shape basis functions?

Back to this problem:

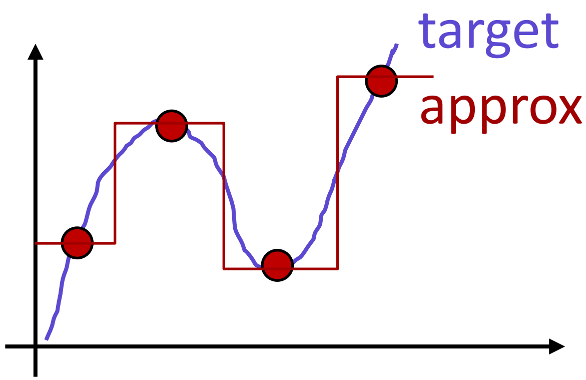


- Shape the functions of an *ensemble* (a whole basis)

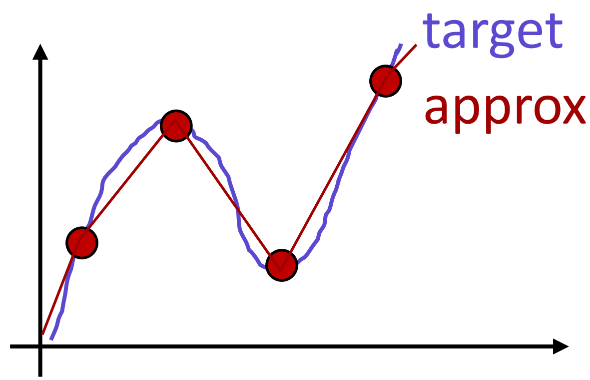
Tools:

- Consistency order
- Frequency space analysis

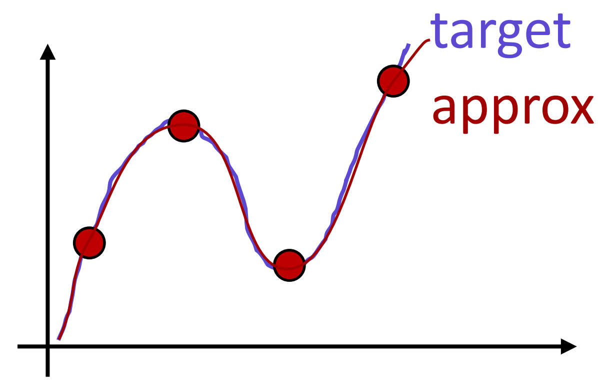
Consistency Order



zero-order



first-order



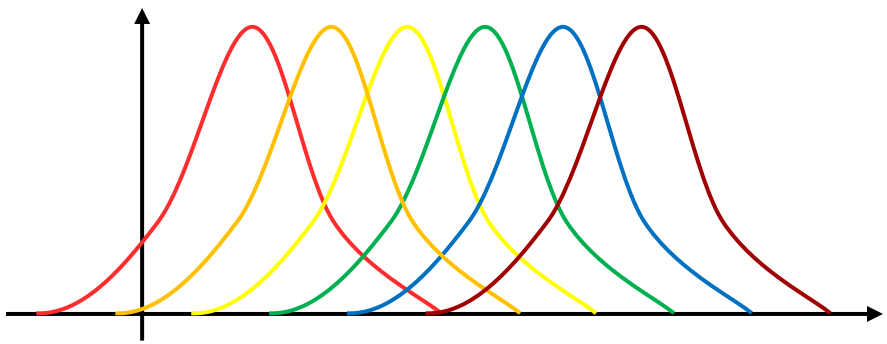
third-order
(PPT cubic splines)

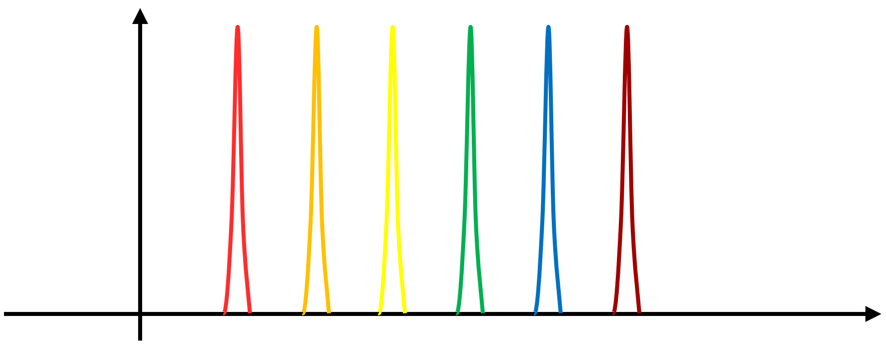
Consistency order:

- A basis of functions is of order *k* iff it can represent polynomials of total degree *k* exactly
- Better fit to smooth targets
- High consistency order: risk of oscillations (later)

Frequency Space Analysis

Which of the following two is better?





Why?

- Long story...
- We’ll look at this next.

A Very Brief Overview of Sampling Theory

Topics

Topics

- Fourier transform
- Theorems
- Analysis of regularly sampled signals
- Irregular sampling

Fourier Basis

Fourier Basis

- Function space: $\{f : \mathbb{R} \to \mathbb{R}, f \text{ sufficiently smooth}\}$
  - Fourier basis can represent
    - Functions of finite variation
    - Lipchitz-smooth functions
- Basis: sine waves of different *frequency* and *phase*:
  - Real basis:

$$\{\sin 2\pi\omega x \, , \cos 2\pi\omega x \ | \ \omega \in \mathbb{R}\}$$

  - Complex variant:

$$\{e^{-2\pi i \omega x} \ | \ \omega \in \mathbb{R}\}$$

(Euler's formula: $e^{ix} = \cos x + i \sin x$)

Fourier Transform

Fourier Basis properties:

- Fourier basis: $\{e^{-i2\pi\omega x} \mid \omega \in \mathbb{R}\}$
  - Orthogonal basis
  - Projection via scalar products $\Rightarrow$ Fourier transform
- Fourier transform: $(\mathrm{f}: \mathbb{R} \to \mathbb{C}) \quad \to \quad (\mathrm{F}: \mathbb{R} \to \mathbb{C})$

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

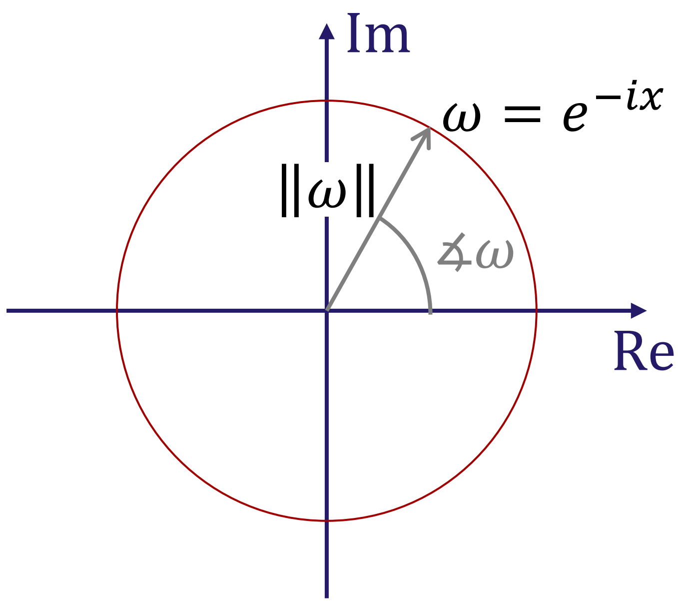
- Inverse Fourier transform: $(\mathrm{F}: \mathbb{R} \to \mathbb{C}) \to (\mathrm{f}: \mathbb{R} \to \mathbb{C})$

$$f(\omega) = \int_{-\infty}^{\infty} F(x) e^{2\pi i x \omega} dx$$

Fourier Transform

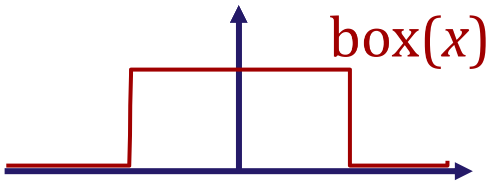
Interpreting the result:

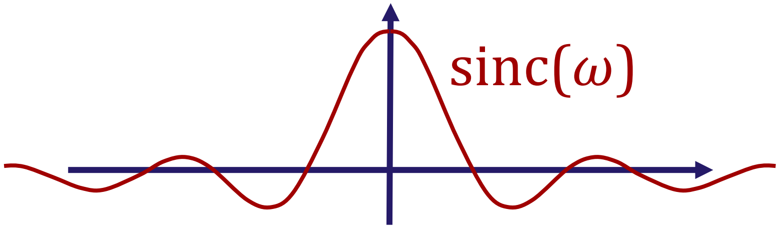
- Transforming a real function f: $\mathbb{R} \rightarrow \mathbb{R}$
- Result: $\mathrm{F}(\omega)$: $\mathbb{R} \rightarrow \mathbb{C}$
  - ω are frequencies (real)
  - Real input f:
    Symmetric $\mathrm{F}(-\omega) = \mathrm{F}(\omega)$
  - Output are complex numbers
    - Magnitude: "power spectrum" (frequency content)
    - Phase: phase spectrum (encodes shifts)



Important Functions

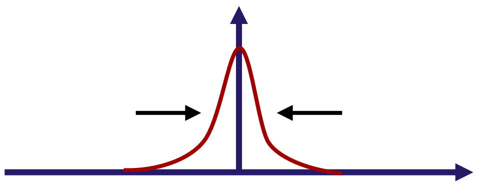
Some important Fourier-transform pairs

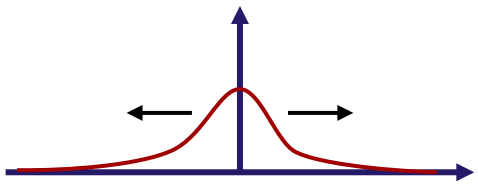




- Box function:

$$f(x) = \mathrm{box}(x) \quad \rightarrow \quad F(\omega) = \frac{\sin\omega}{\omega} := \mathrm{sinc}(\omega)$$





- Gaussian:

$$f(x) = e^{-ax^2} \quad \rightarrow \quad F(\omega) = \sqrt{\frac{\pi}{a}} \cdot e^{-\frac{(\pi\omega)^2}{a}}$$

Higher Dimensional FT

Multi-dimensional Fourier Basis:

- Functions f: $\mathbb{R}^d \to \mathbb{C}$
- 2D Fourier basis:

$f(x, y)$ represented
as combination of

$$\left\{e^{-i2\pi\omega_x x} \cdot e^{-i2\pi\omega_y y} \,\middle|\, \omega_x, \omega_y \in \mathbb{R}\right\}$$

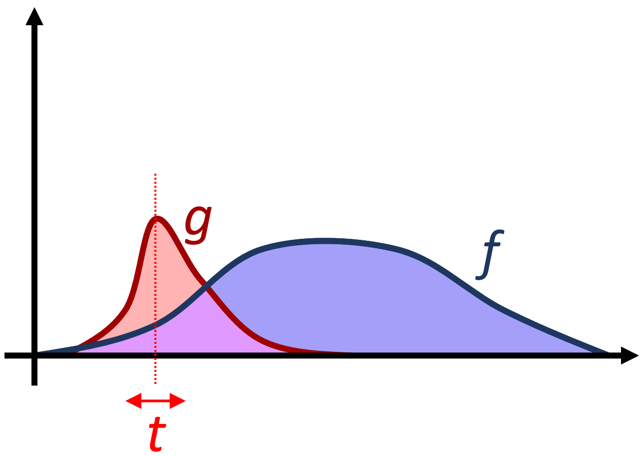
- In general: all combinations of 1D functions

Convolution

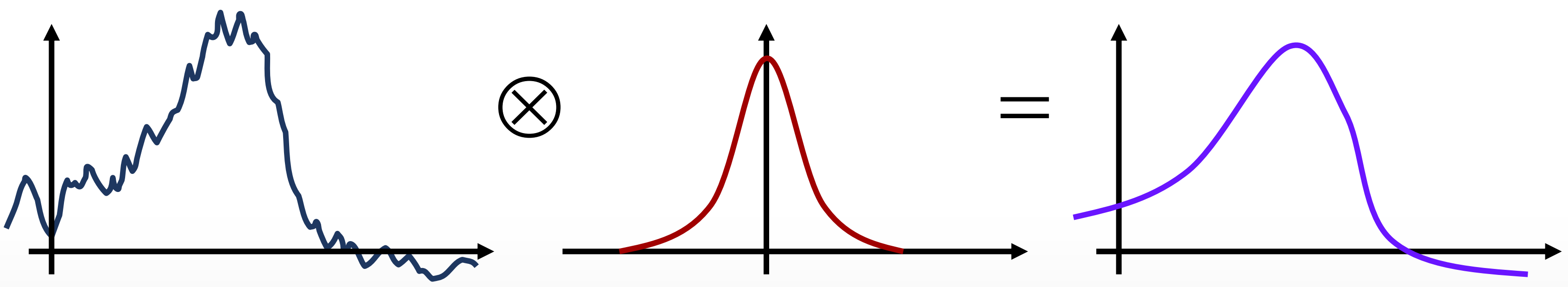
Convolution:

- Weighted average of functions
- Definition:

$$f(t) \otimes g(t) = \int_{-\infty}^{\infty} f(x) g(x-t) dx$$



Example:



Theorems

Fourier transform is an isometry:

- $\langle f, g \rangle = \langle F, G \rangle$
- In particular $\|f\| = \|F\|$

Convolution theorem:

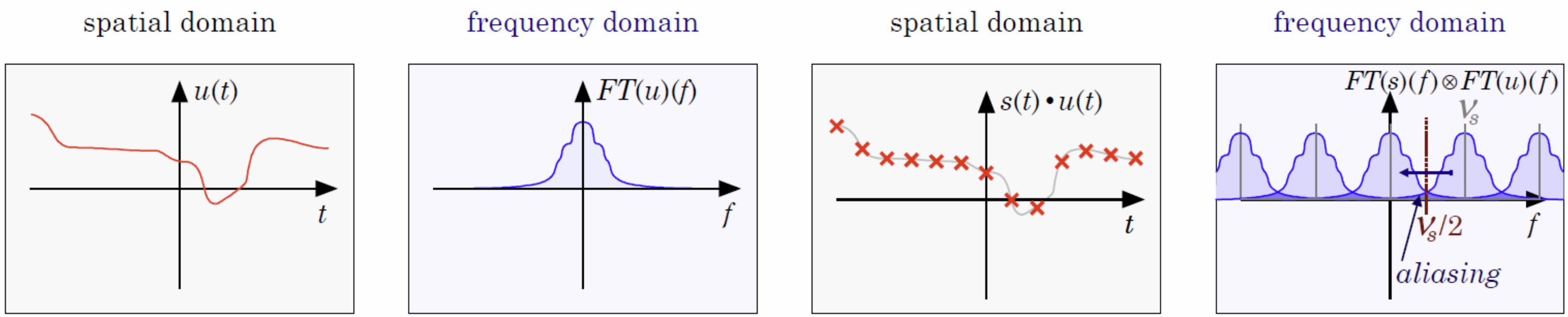
- $FT(f \otimes g) = F \cdot \mathrm{G}$
- Fourier Transform converts convolution into multiplication
  - All other cases as well: $FT^{-1}(f \cdot g) = F \otimes \mathrm{G}$, $FT(f \cdot g) = F \otimes \mathrm{G}$, $FT^{-1}(F \cdot G) = F \otimes \mathrm{G}$
  - Fourier basis diagonalizes shift-invariant linear operators

Sampling a Signal

Given:

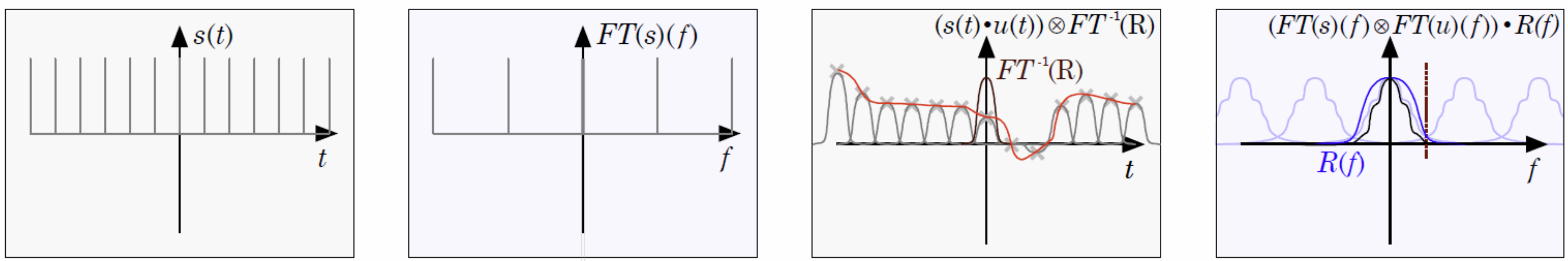
- Signal $f: \mathbb{R} \to \mathbb{R}$
- Store digitally:
  - Sample regularly … $f(0.3), f(0.4), f(0.5)$ …
- Question: what information is lost?

Sampling



(a) a continuous function and its frequency spectrum

(c) sampling: frequencies beyond the Nyquest limit $\nu_s/2$ appear as aliasing



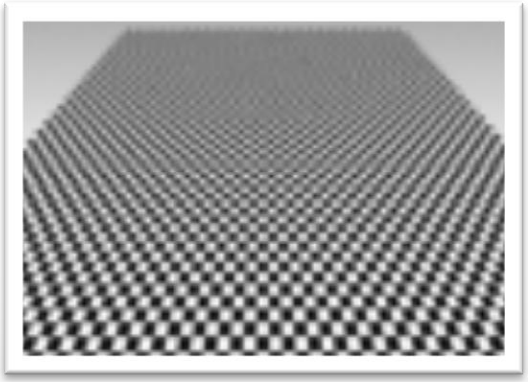
(b) a regular sampling pattern (impulse train) and its frequency spectrum

(d) reconstruction: filtering with a low-pass filter R to remove replicated spectra

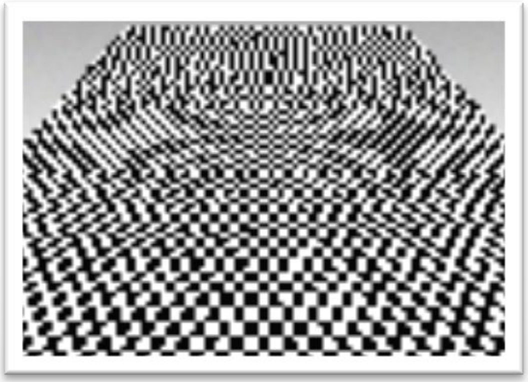
Regular Sampling

Case I: Sampling

- Band-limited signals can be represented exactly
  - Sampling with frequency ν_s:
    Highest frequency in Fourier spectrum $\leq \nu_s/2$
- Higher frequencies *alias*
  - Aliasing artifacts (low-frequency patterns)
  - Cannot be removed after sampling (loss of information)



band-limited

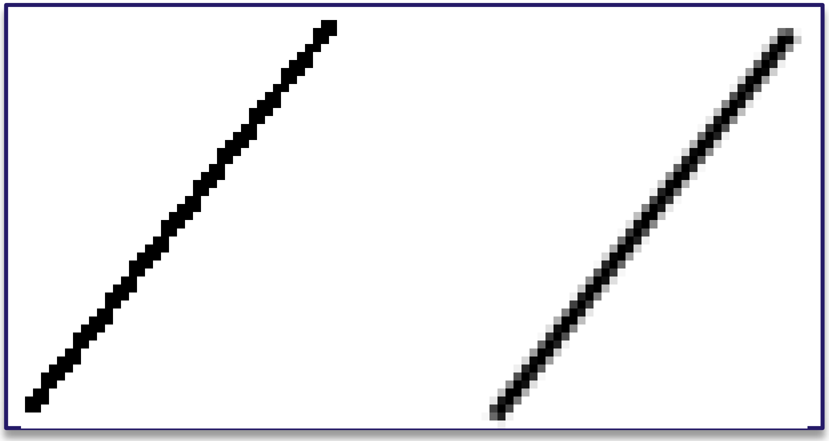


aliasing

Regular Sampling

Case II: Reconstruction

- When reconstructing from discrete samples
- Use band-limited basis functions
  - Highest frequency in Fourier spectrum $\leq \nu_s/2$
  - Otherwise: Reconstruction aliasing

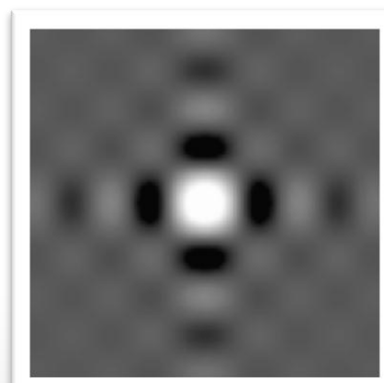


Regular Sampling

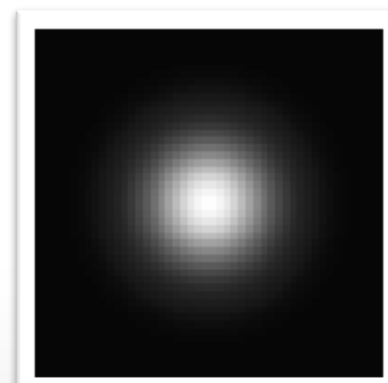
Reconstruction Filters

- Optimal filter: sinc
  (no frequencies discarded)
- However:
  - Ringing artifacts in spatial domain
  - Not useful for images (better for audio)
- Compromise
  - Gaussian filter
    (most frequently used)
  - There exist better ones,
    such as Mitchell-Netravalli,
    Lancos, etc...

Ringing by sinc reconstruction
from [Mitchell & Netravali, Siggraph 1988]



2D sinc



2D Gaussian

Irregular Sampling

Irregular Sampling

- No comparable formal theory
- However: similar idea
  - Band-limited by “sampling frequency”
  - Sampling frequency = mean sample spacing
    - Not as clearly defined as in regular grids
    - May vary locally (adaptive sampling)
- Aliasing
  - Random sampling creates noise as aliasing artifacts
  - Evenly distributed sample concentrate noise in higher frequency bands in comparison to purely random sampling

Consequences for our applications

When designing bases for function spaces

- Use band-limited functions
- Typical scenario:
  - Regular grid with spacing σ
  - Grid points $\mathbf{g}_i$
  - Use functions: $\exp\left(-\frac{(\mathbf{x}-\mathbf{g}_i)^2}{\sigma^2}\right)$
- Irregular sampling:
  - Same idea
  - Use estimated sample spacing instead of grid width
  - Set σ to average sample spacing to neighbors

Tutorials:

Linear Algebra Software

GeoX

GeoX comes with several linear algebra libraries:

- 2D, 3D, 4D vectors and matrices: *LinearAlgebra.h*
- Large (dense) vectors and matrices: *DynamicLinearAlgebra.h*
- Gaussian elimination: *invertMatrix()*
- Sparse matrices: *SparseLinearAlgebra.h*
- Iterative solvers (Gauss-Seidel, conjugate gradients, power iteration): *IterativeSolvers.h*