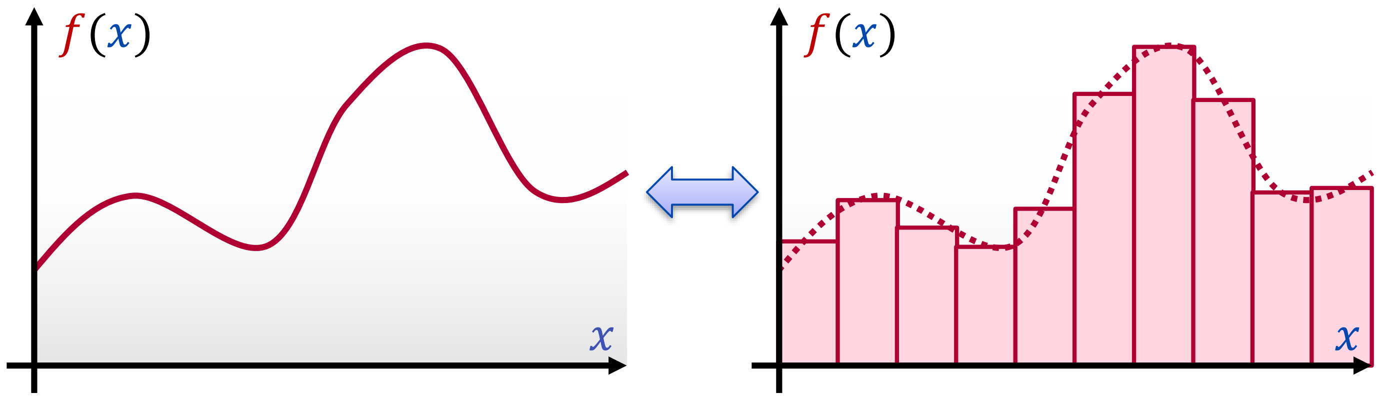
Modelling 1

SUMMER TERM 2020



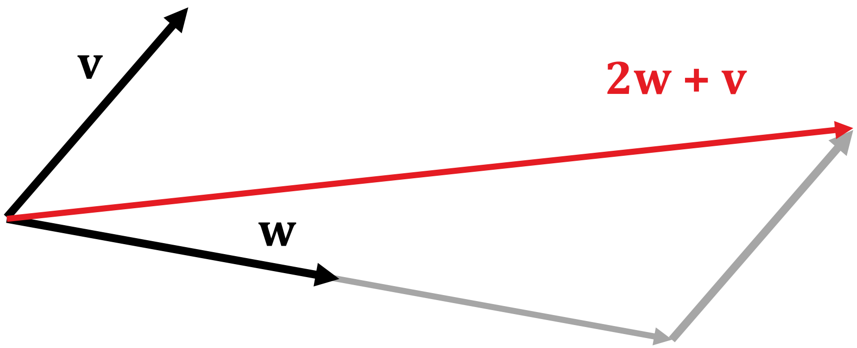
LECTURE 6

Representing Functions

Michael Wand · Institut für Informatik · Michael.Wand@uni-mainz.de

Representing Functions

Recap: Vector Spaces



Geometry

$$2\mathbf{w} + \mathbf{v} = \begin{pmatrix} 2w_1 + v_1 \\ 2w_2 + v_2 \\ 2w_3 + v_3 \end{pmatrix}$$

Algebra

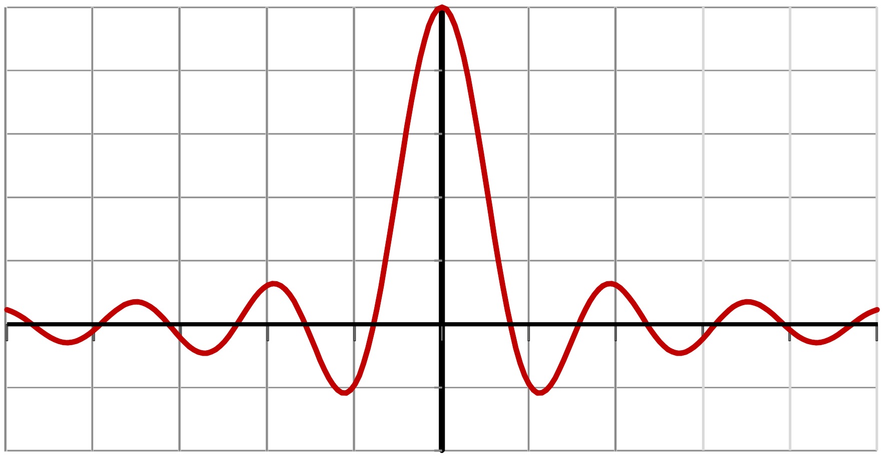
Abstract vector spaces

≈ “arrows” in flat space

≈ arrays of numbers

Same principle for set of functions

Representing Functions



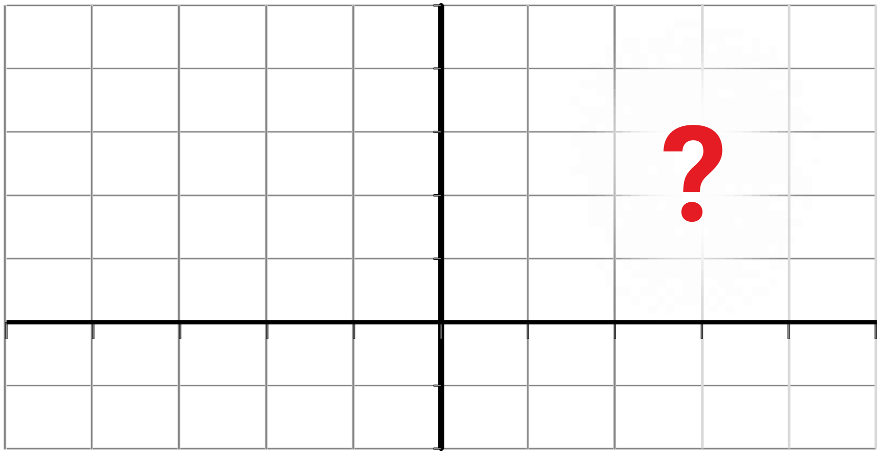
Option #1:

- Write down algorithm / formula,

  e.g.: $f(x) := \frac{\sin(x)}{x}$

- Problem: closed form formulation often not known

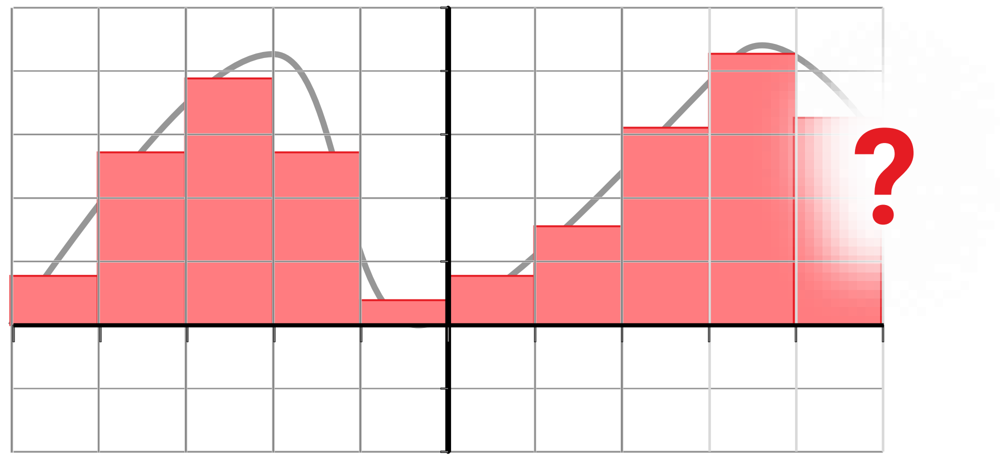
Representing Functions



Option #2:

- Parametrize function space
  - Then search for the right one
- Two approaches
  - Array of numbers
  - Combination of basis functions

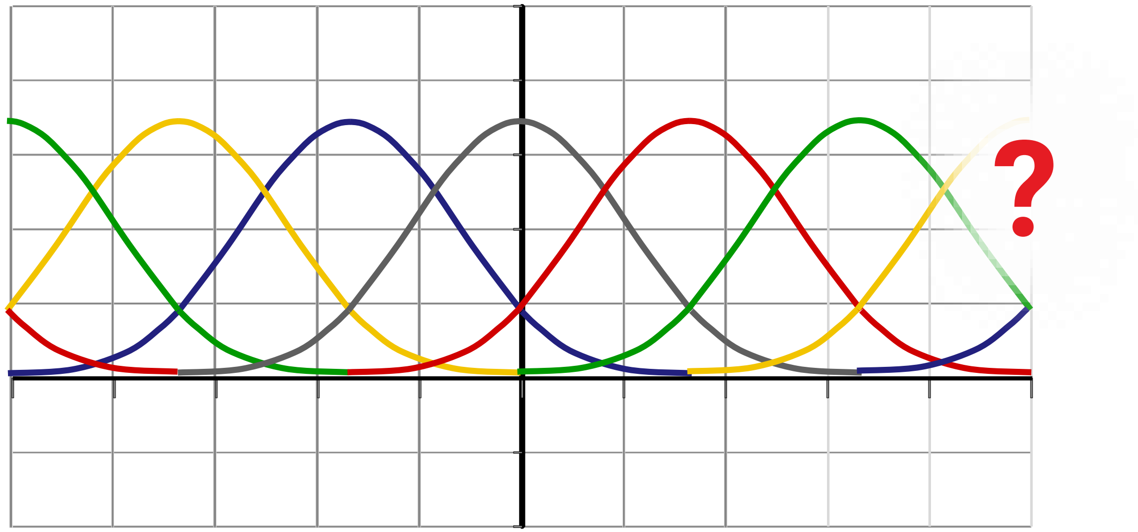
Approximation of Function Spaces



Parametrization #1: Array of Numbers

- Sample function f on discrete grid
- Store sample values
- (Use this as intuition)

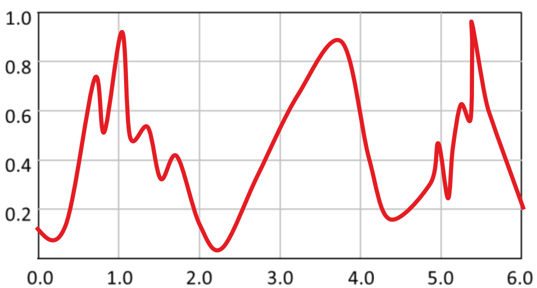
Approximation of Function Spaces



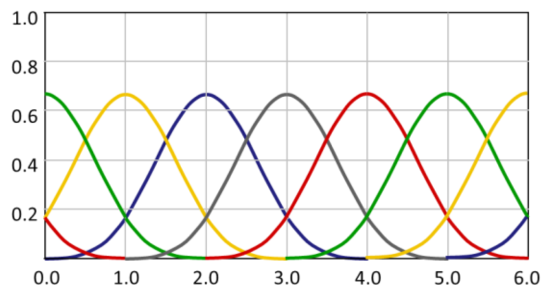
Parametrization #2: Linear Ansatz

- Choose basis functions
- Find linear combination
- (More flexible)

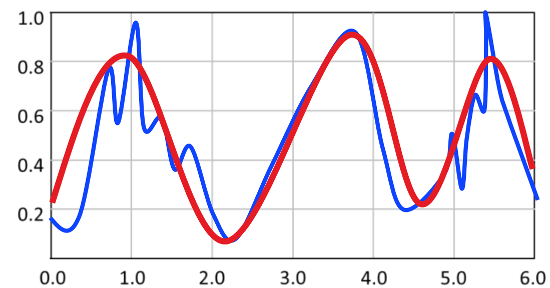
Approximation of Function Spaces



actual solution



function space basis

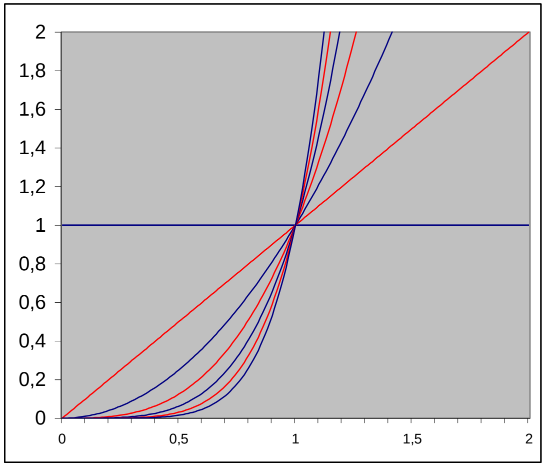


approximate solution

Approach #2:

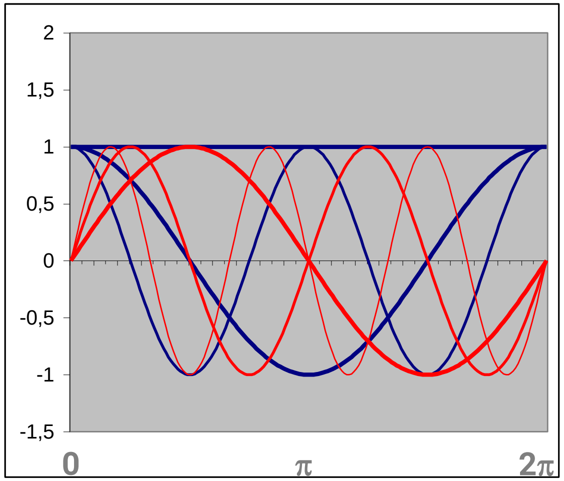
- Choose basis functions $b_1, \ldots, b_d \in \mathrm{V}$
- Find approximation $\tilde{f} = \sum_{i=1}^{d} \lambda_i b_i$
- Coordinates: $\tilde{f}$ is described by $(\lambda_1, \ldots, \lambda_d)$
- Euclidean geometry

Example Basis Functions



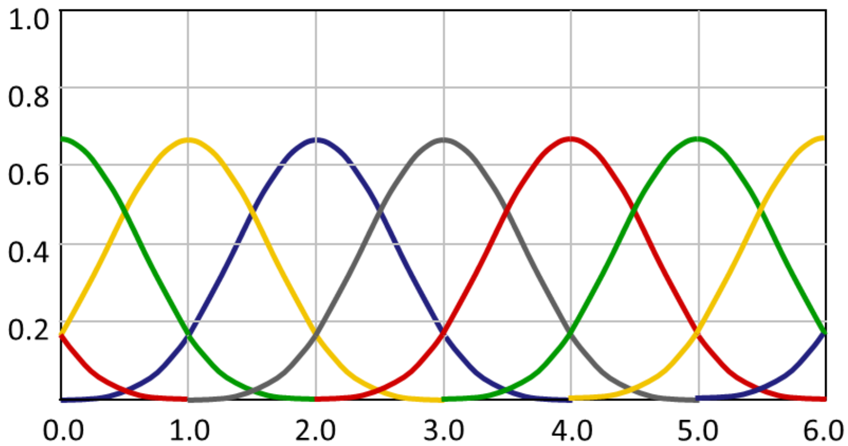
Monomial basis

$\{1, x, x^2, x^3, \dots\}$



Fourier basis

$\{\sin \omega x, cos\omega x | \omega = 0,1,2, \dots\}$



Gaussian basis (**B-spline basis**)

$\{e^{-(x-i)^2} | i = \cdots - 2, -1,0,1,2, \dots\}$

Constructing Bases

How to construct a basis?

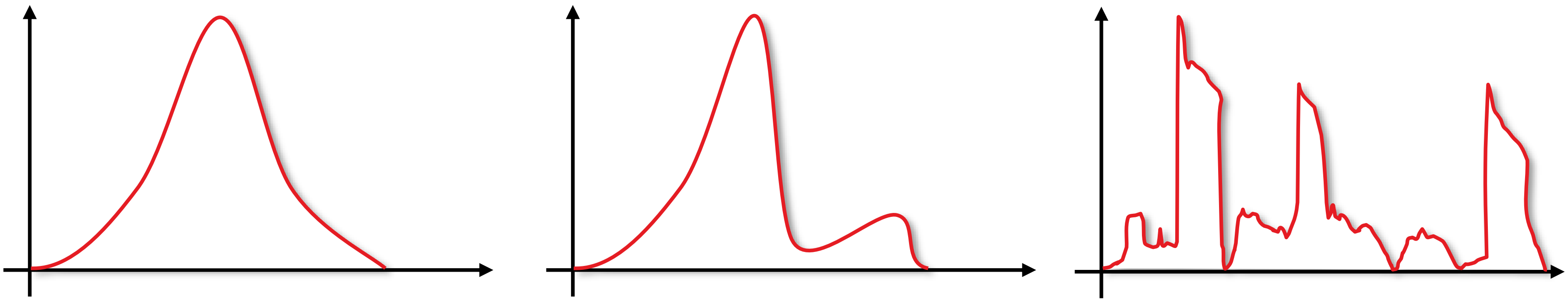
Important tool

- Build a good basis for a problem

Ingredients:

- Basis functions
- Placement in space
- Semantics

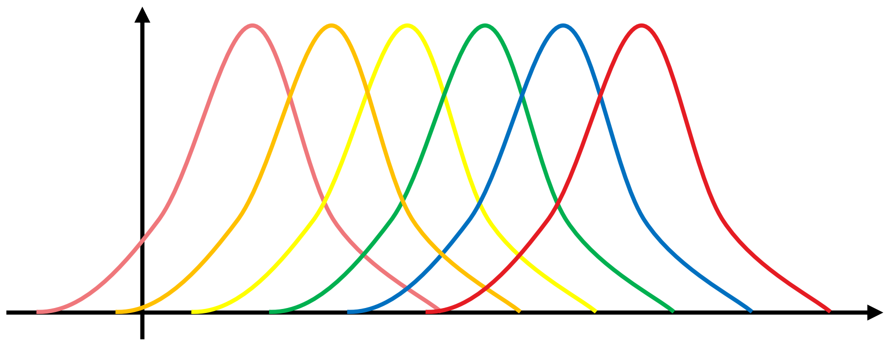
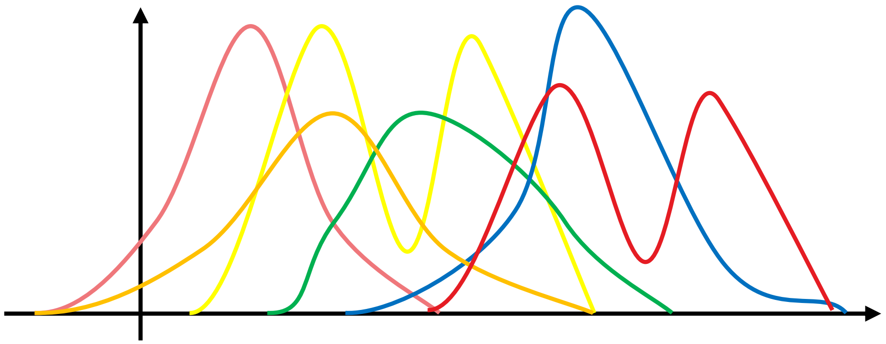
Basis Function



Shape of individual functions:

- Smoothness
- Symmetry
- Support

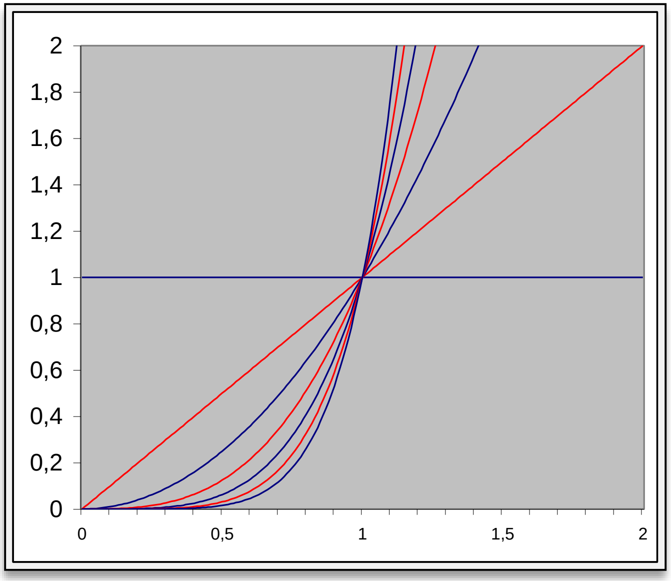
Ensembles of Functions

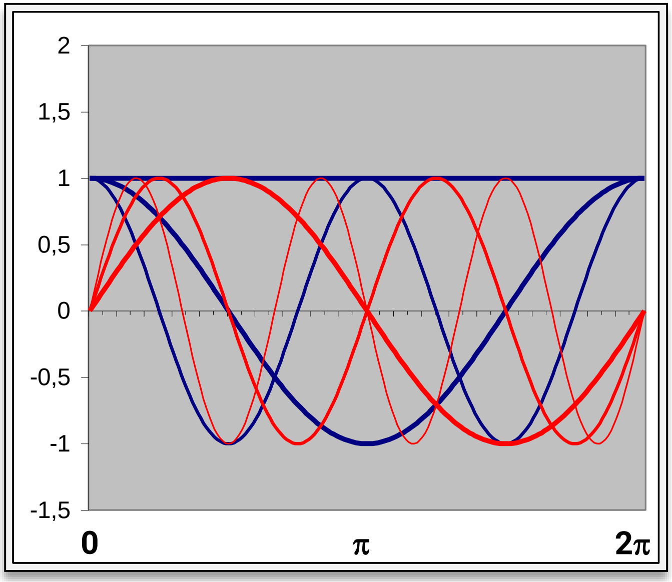
Basis function sets:

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

Ensembles of Functions



Monomial basis



Fourier basis (orthogonal)

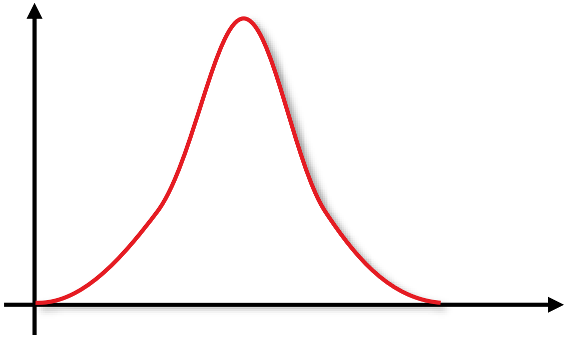
Basis function sets:

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

Example: Radial Basis functions

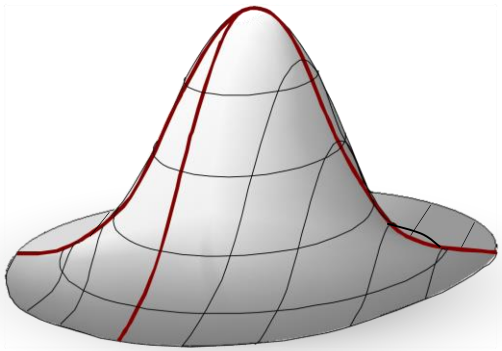
Radial basis function:

- Pick one template function
- Symmetric around "center" point



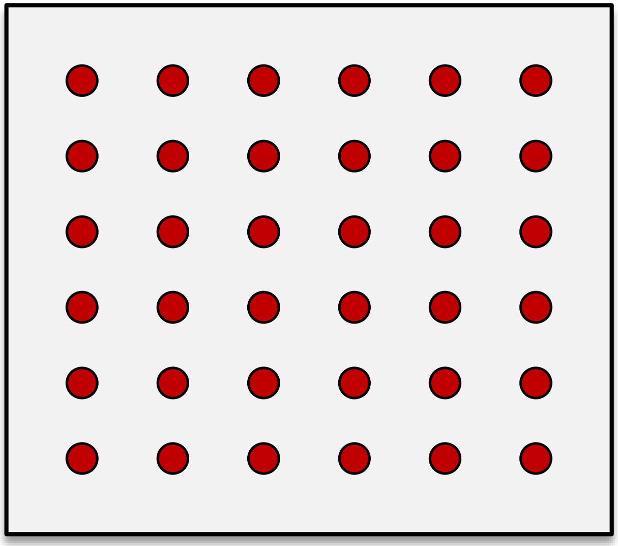
1D

Instantiate by placing in domain

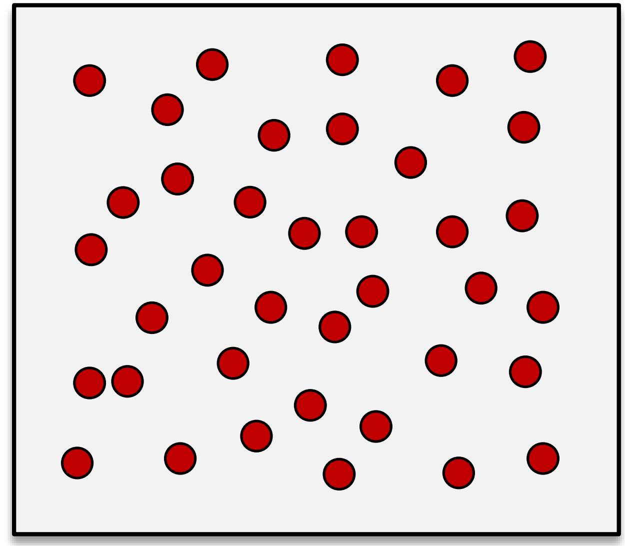


2D

Placement



Regular grids

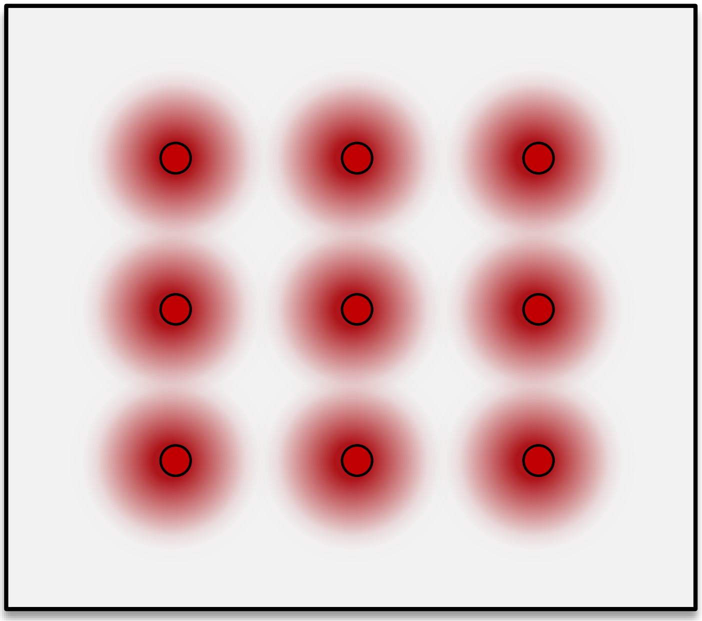


Irregular

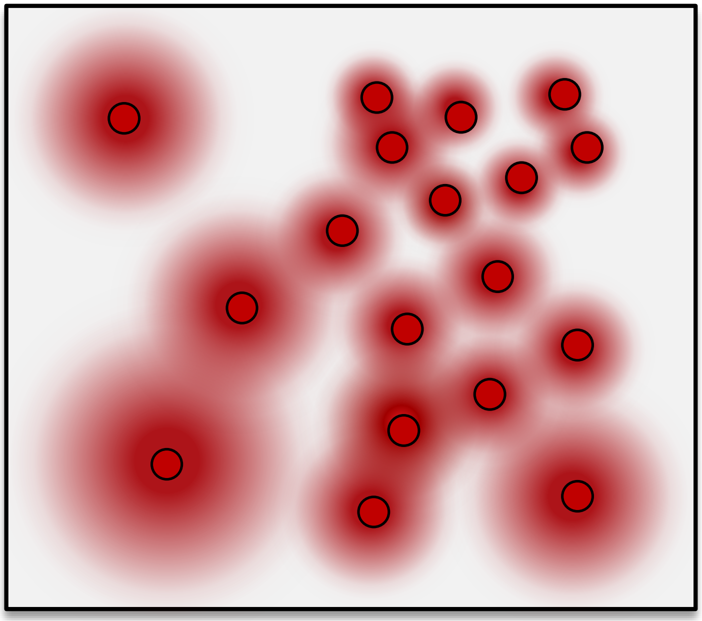
Context

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

Placement



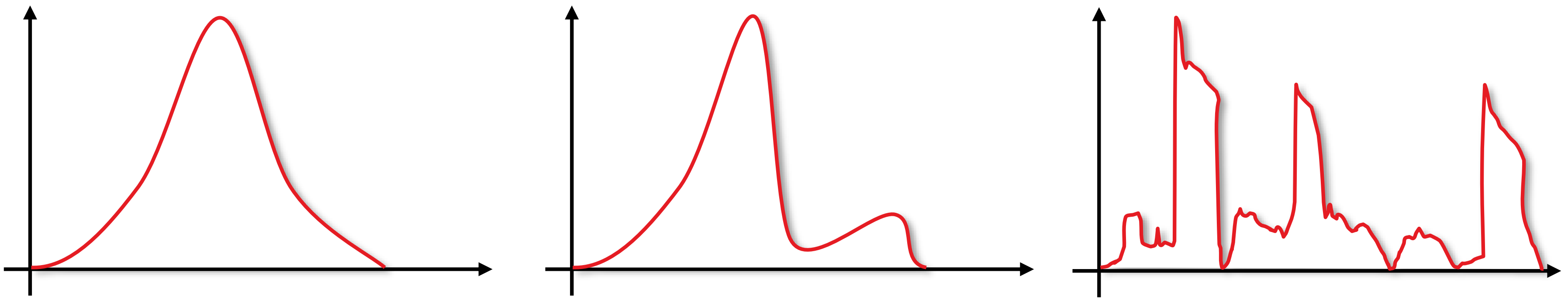
Regular grids



Irregular
(w/scaling)

How to shape basis functions?

Back to this problem:

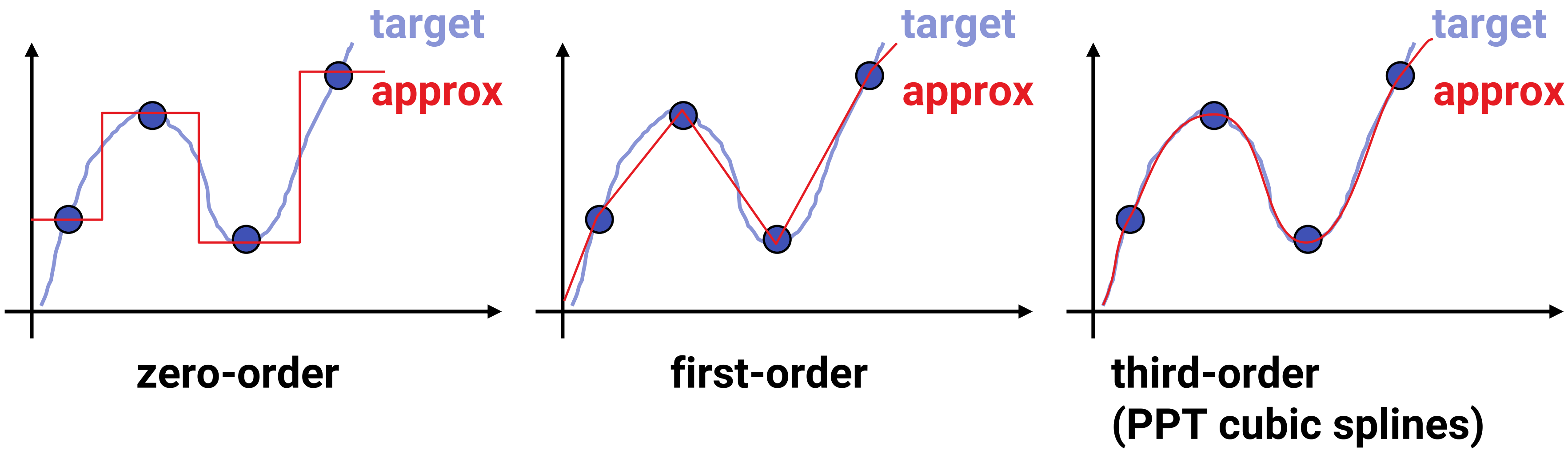


- Shape the functions of an *ensemble* (a whole basis)

Tools:

- Consistency order
- Frequency space analysis

Consistency Order

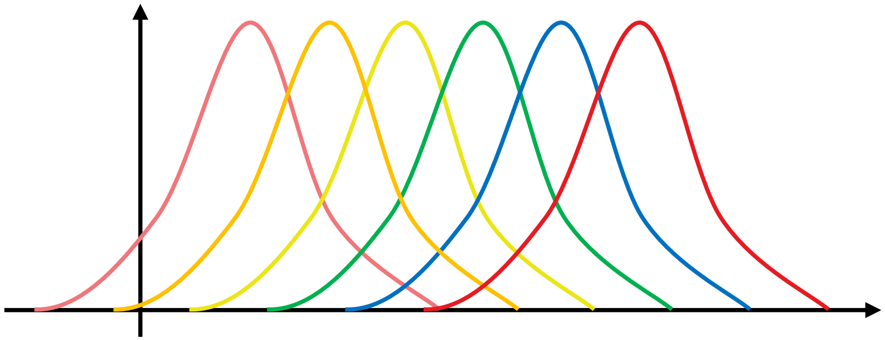


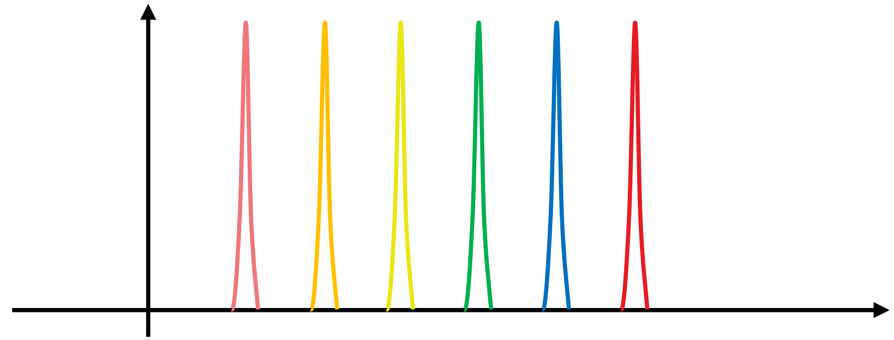
Consistency order:

- Basis of order k
  $\Leftrightarrow$ represents polynomials of (total) degree k exactly
  - Better fit to smooth targets
  - High consistency order: risk of oscillations

Frequency Space Analysis

Which of the following two is better?



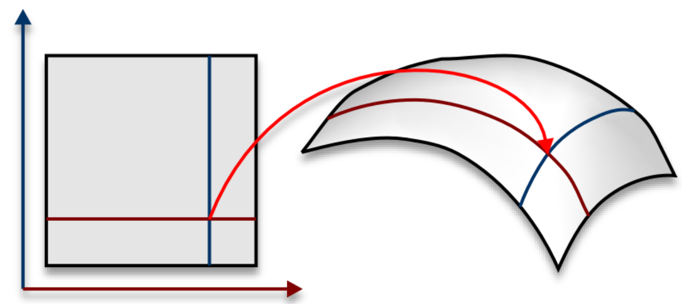


Obvious, but why?

- Long story...
  - Sampling theory
  - Fourier transforms involved
- We'll look at this later.

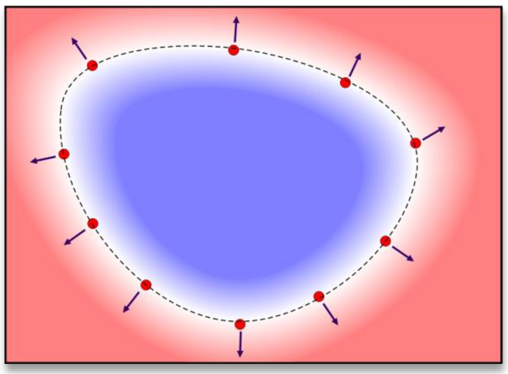
Semantics

Explicit representations



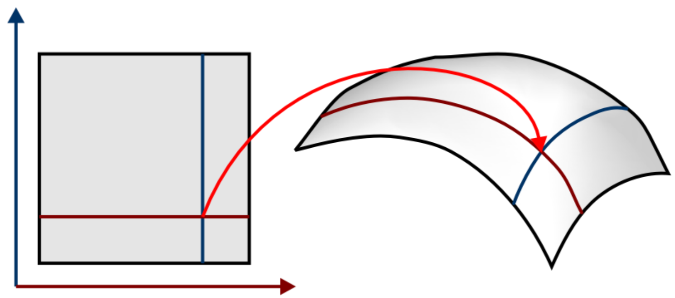
- Height field
- Parametric surface
- Function value corresponds to actual geometry

Implicit representation

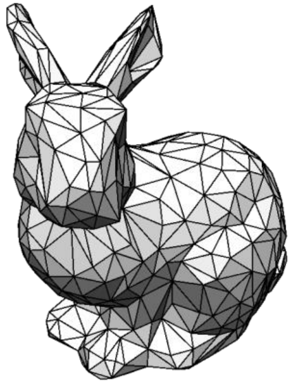


- Scalar fields
- Zero crossings correspond to actual geometry

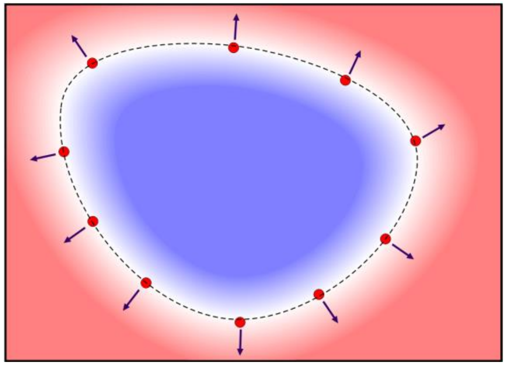
Modeling Zoo



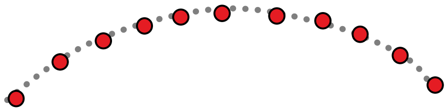
Parametric Models



Primitive Meshes



Implicit Models



Point-Based Models