KU LEUVEN

Introducing redundancy into numerical computations

Computing with frames

Goal

- **What?** Function approximation
- **Why?** Building block of larger problems
- **How?** Efficient and accurate algorithms

Function Approximation

“Be approximately right rather than exactly wrong” - J. Tukey

What is a function?

- Gives you output for certain **inputs**
- Mathematical object:

$$f(\boldsymbol{x}) = \int \frac{\sin(x)}{x}$$

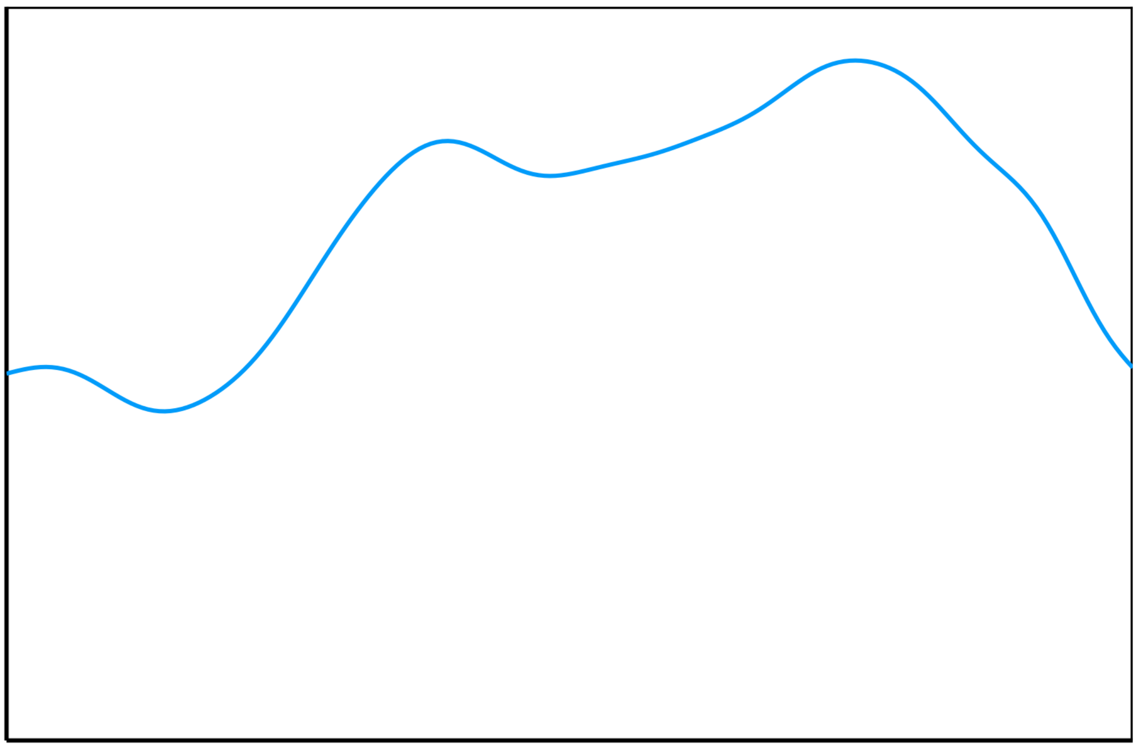
- Data:

*Average baby weight as a function of **age***

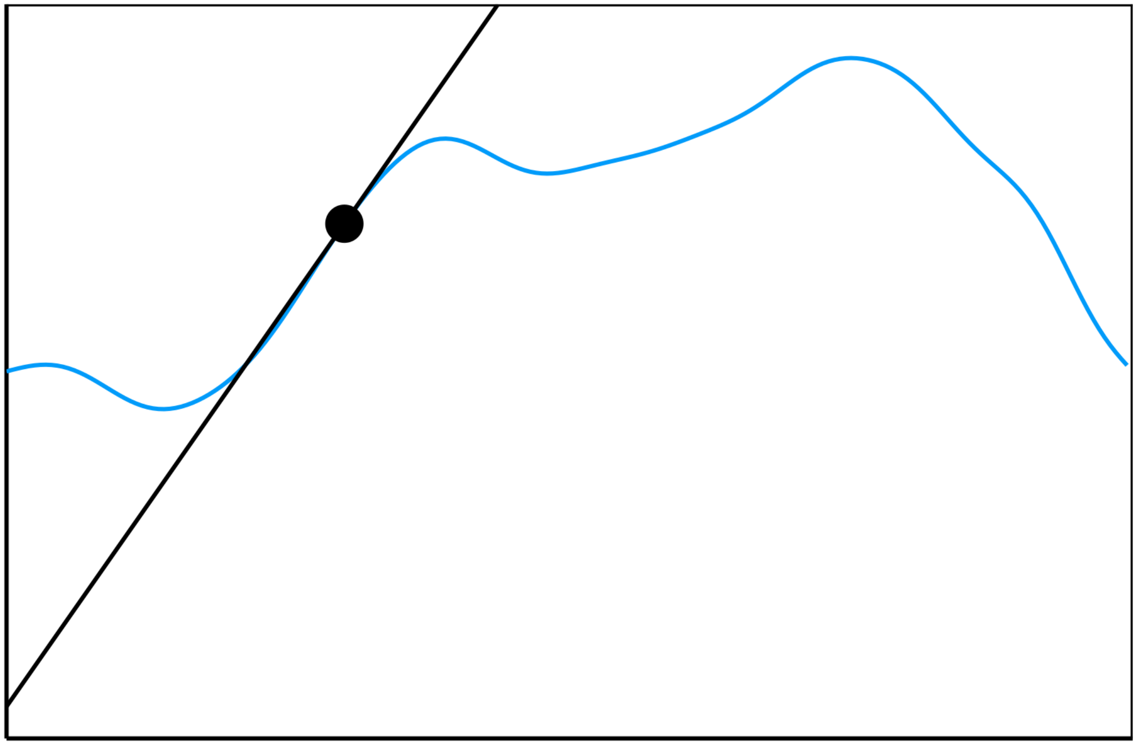
- Physical signal:

*Outside temperature as a function of **time***

Why approximate?

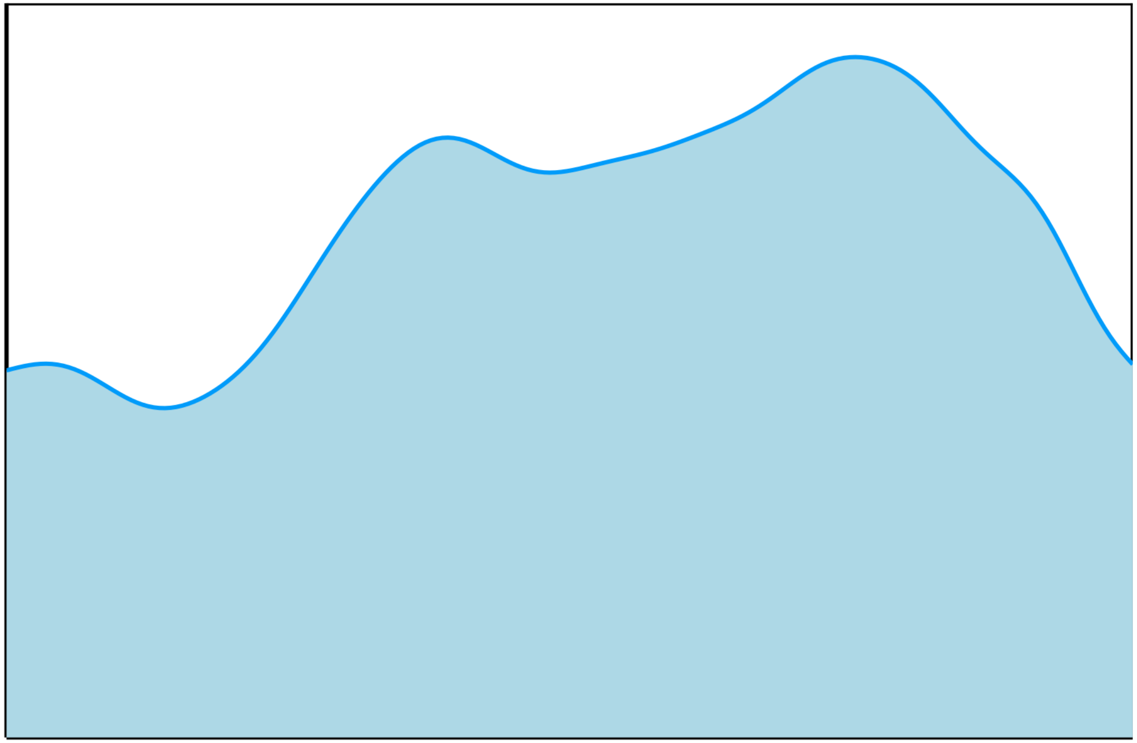


Why approximate?



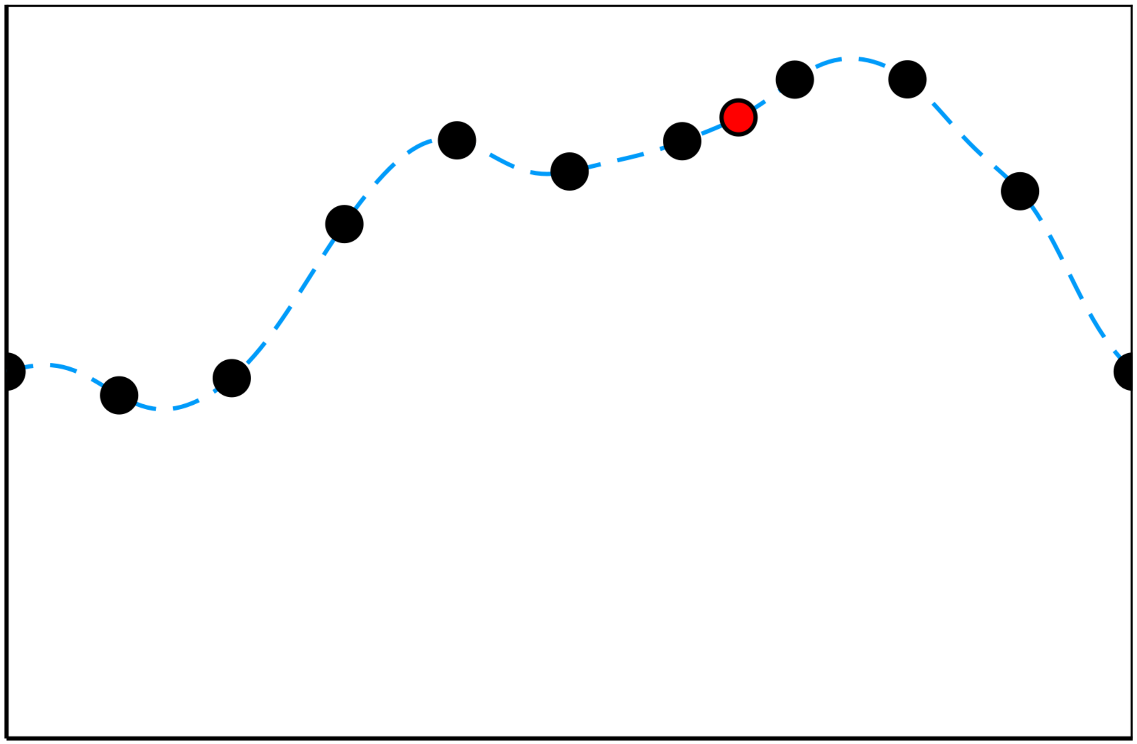
Derivatives

Why approximate?



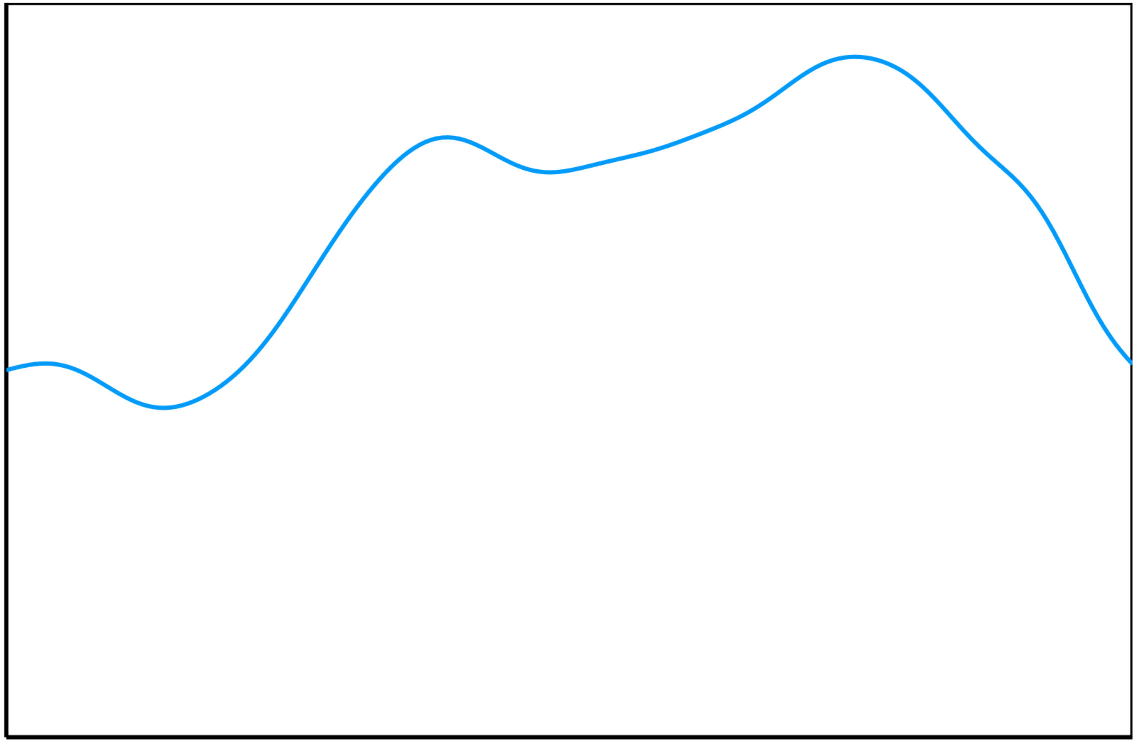
Derivatives, Integrals

Why approximate?



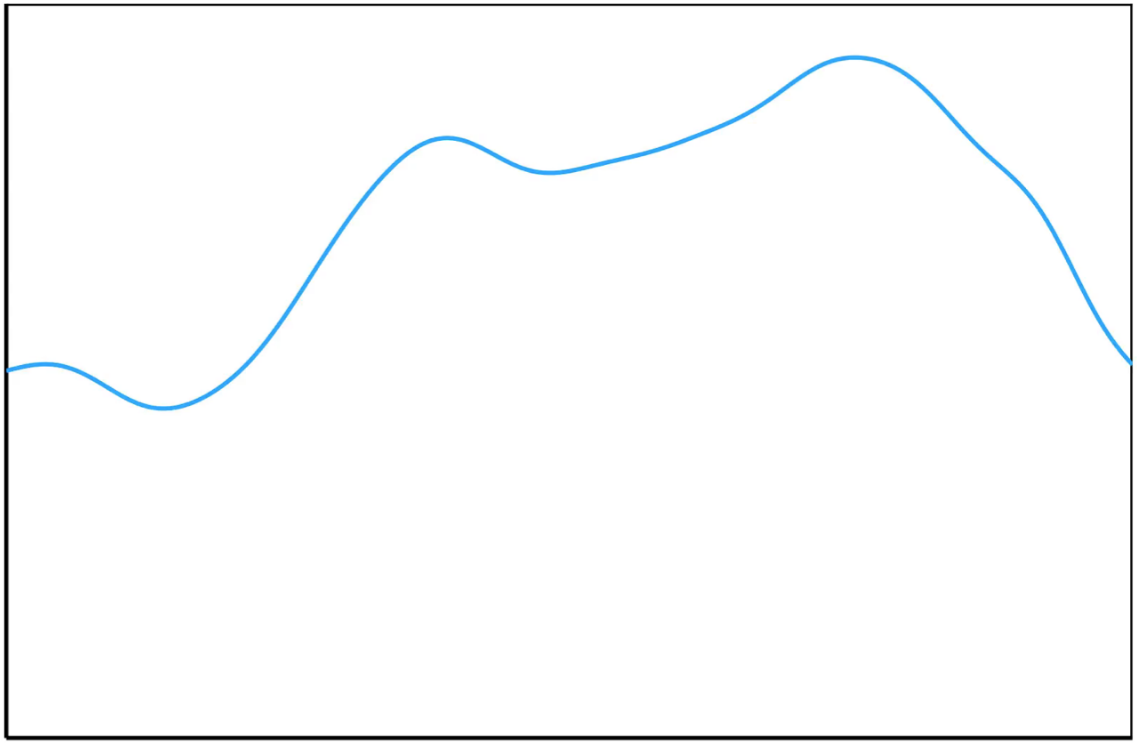
Derivatives, Integrals, Interpolation

Why approximate?



Derivatives, Integrals, Interpolation, Differential Equations

Why approximate?



Derivatives, Integrals, Interpolation, Differential Equations, …

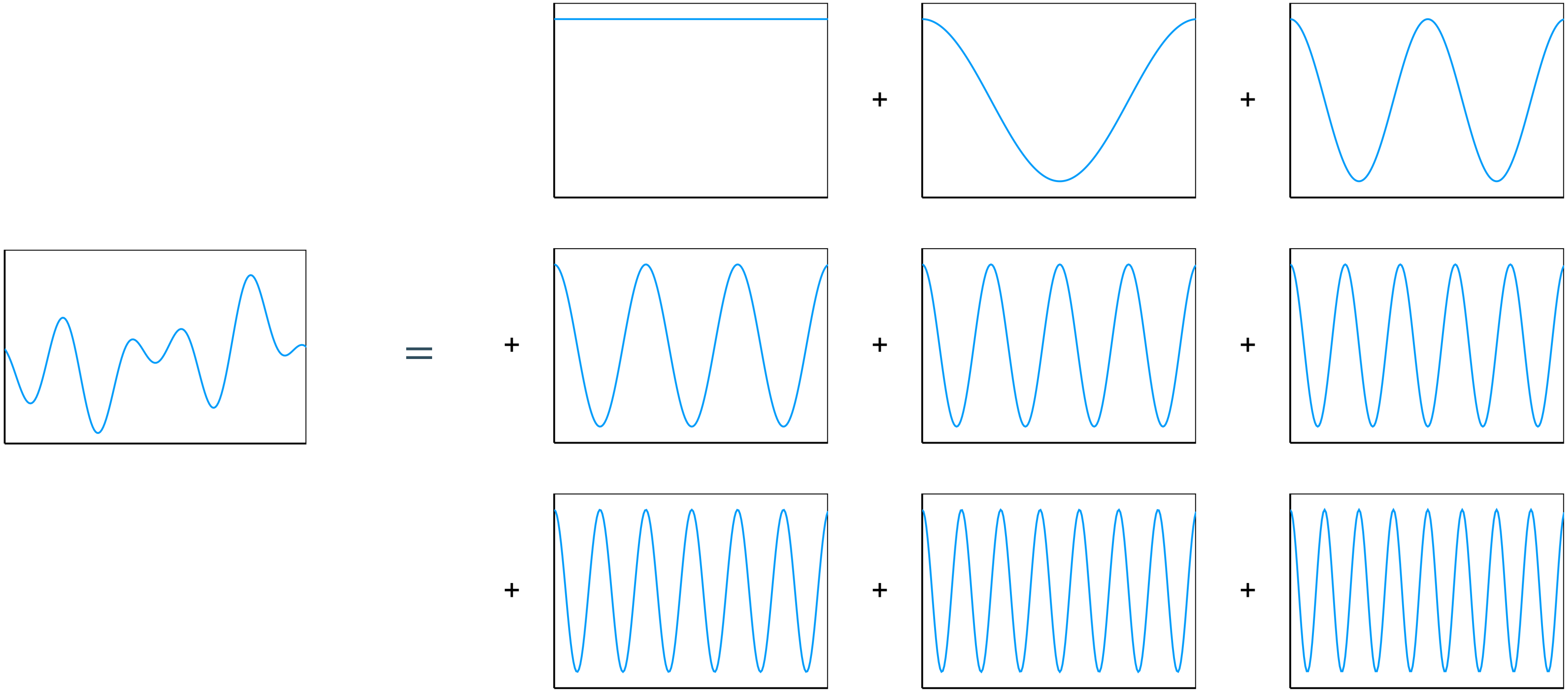
How to approximate?

- **Sum** of **basic** functions:

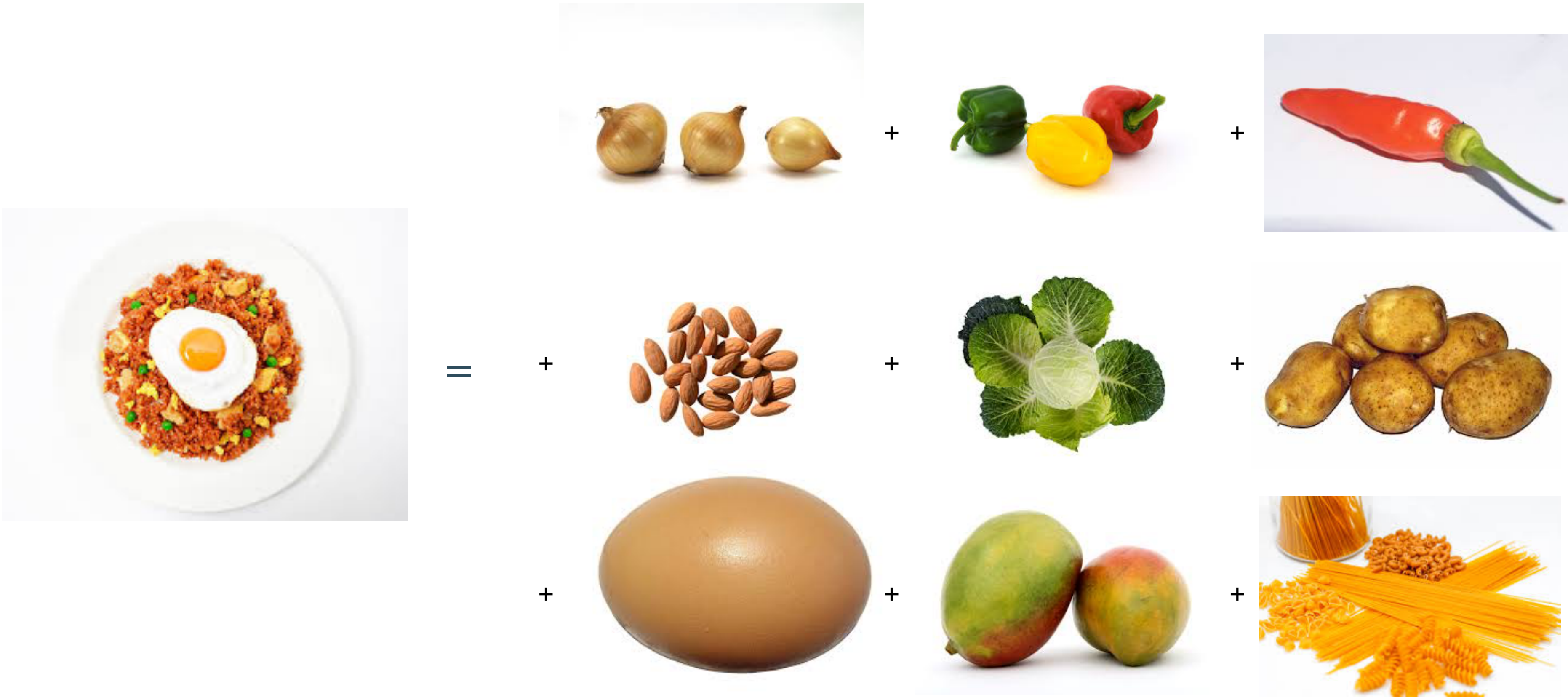
$$f(x) = \sum_k c_k \cos(\pi k x)$$

- **Basic** functions:
  - Simple derivatives
  - Simple integrals
  - Simple …

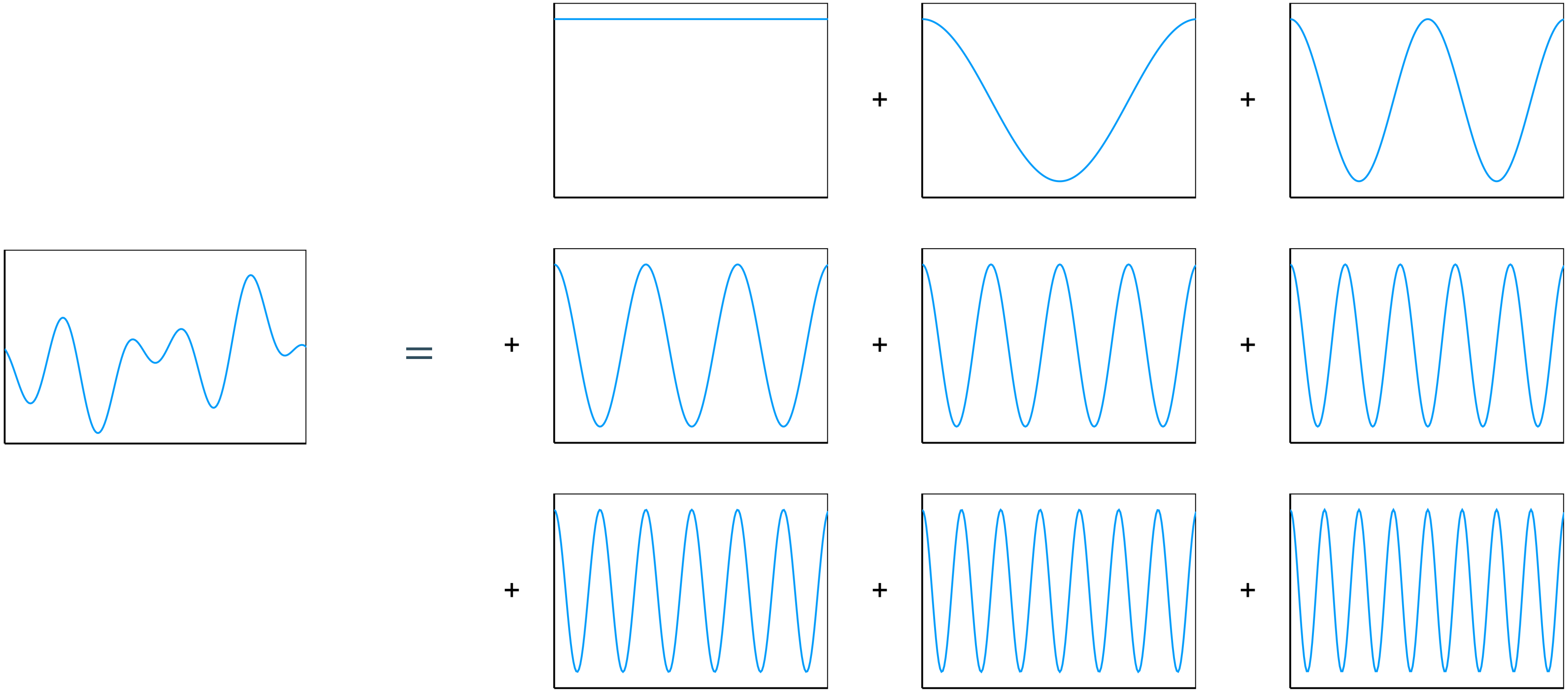
Sum of cosines



Sum of cosines



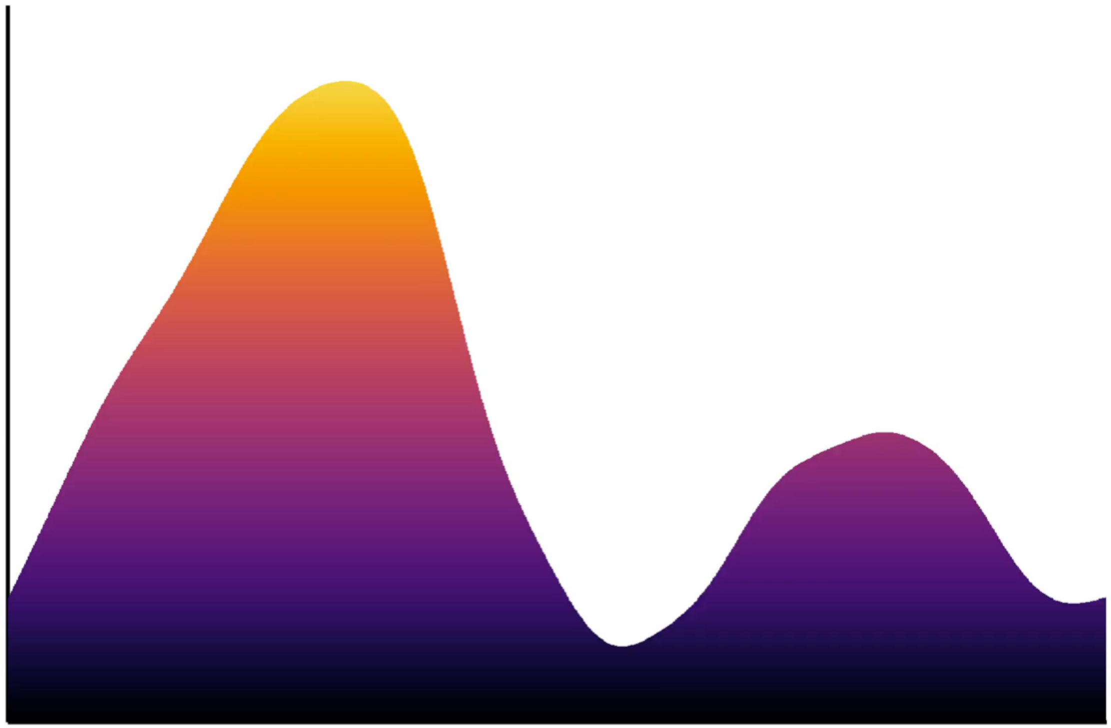
Sum of cosines



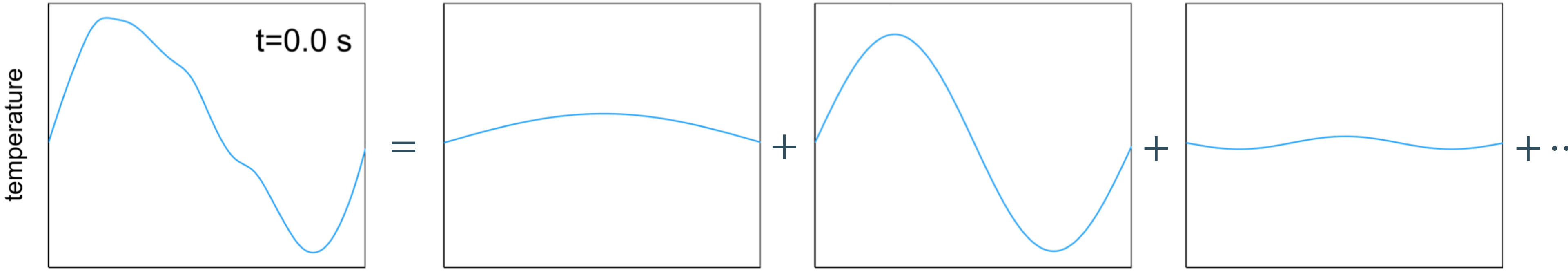
Joseph Fourier

- *Théorie analytique de la chaleur*
- **Every function** can be written as a sum of sines and cosines

Heat equation



Heat equation

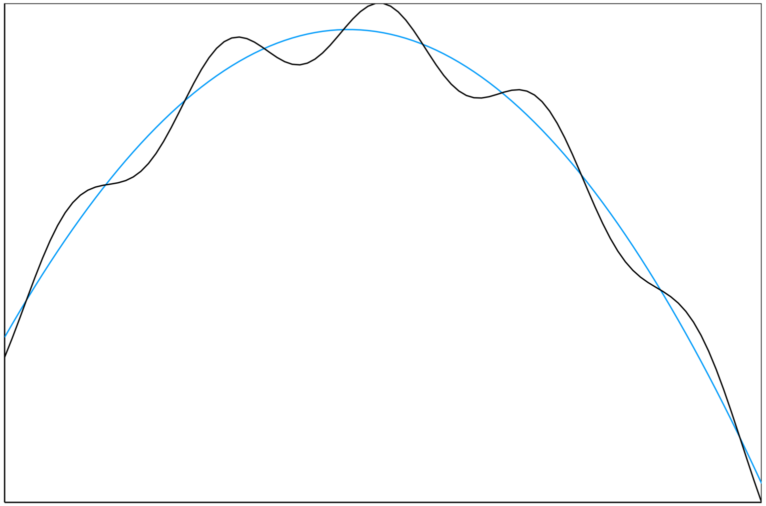


“Good” Function Approximation

“Swiftly and with style” – Mr. Alphonse

Accuracy

- Correctness of approximation for increasing N
- Possible measure: maximum error



- Should go down as fast as possible for increasing N

Speed

- Execution time scales with problem size N
- Linear $O(N)$, quadratic $O(N^2)$ and cubic $O(N^3)$
- Quadratic algorithm: 10 times bigger problem takes 100 times longer

Speed

- Execution time scales with problem size N
- Linear $O(N)$, quadratic $O(N^2)$ and cubic $O(N^3)$
- Quadratic algorithm: 10 times bigger problem takes 100 times longer

N	$O(N)$	$O(N^2)$	$O(N^3)$
1	1 ms	1 ms	1 ms

Speed

- Execution time scales with problem size N
- Linear $O(N)$, quadratic $O(N^2)$ and cubic $O(N^3)$
- Quadratic algorithm: 10 times bigger problem takes 100 times longer

$\boldsymbol{N}$	$\boldsymbol{O(N)}$	$\boldsymbol{O(N^2)}$	$\boldsymbol{O(N^3)}$
1	1 ms	1 ms	1 ms
100	100 ms	10 s	16 minutes

Speed

- Execution time scales with problem size N
- Linear $O(N)$, quadratic $O(N^2)$ and cubic $O(N^3)$
- Quadratic algorithm: 10 times bigger problem takes 100 times longer

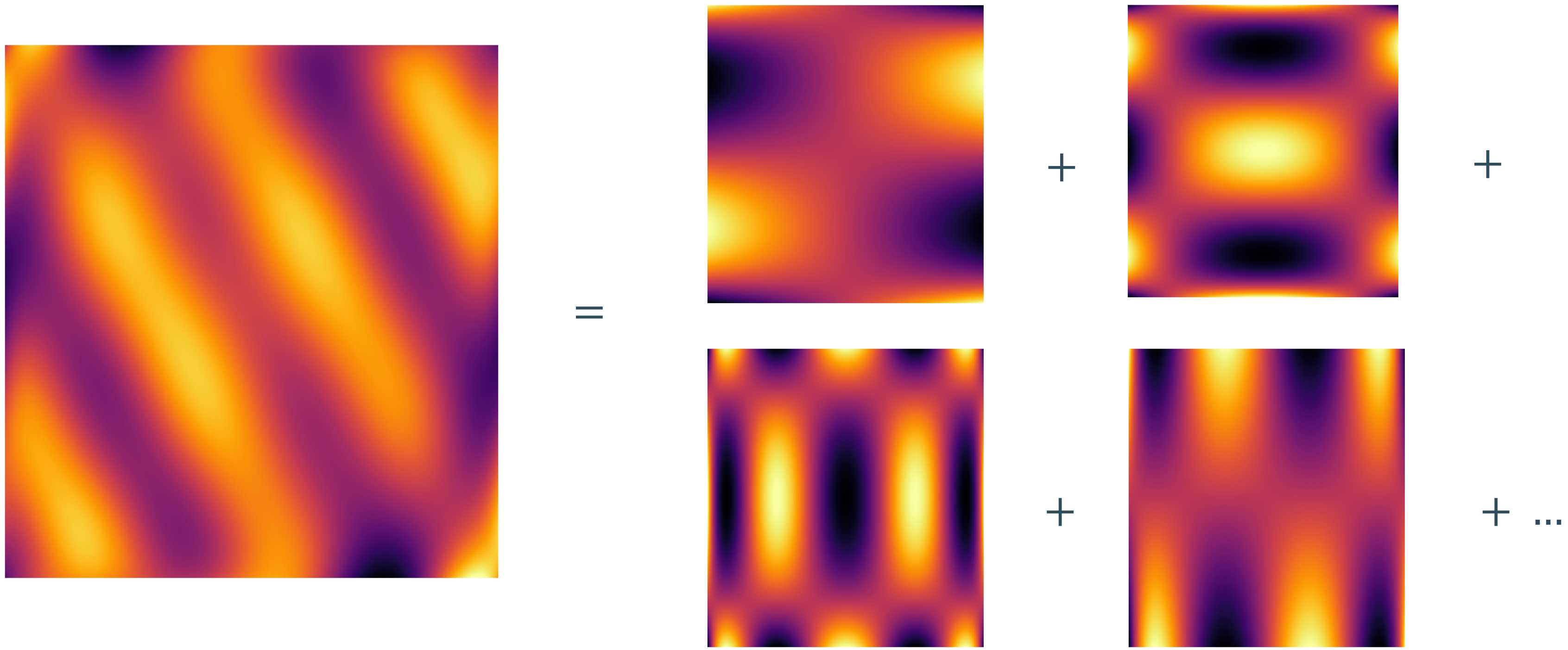
N	$O(N)$	$O(N^2)$	$O(N^3)$
1	1 ms	1 ms	1 ms
100	100 ms	10 s	16 minutes
10000	10 s	3 days	32 years

Frame vs Basis

“I’m nothing if not redundant! I also repeat myself.” – R. Fish

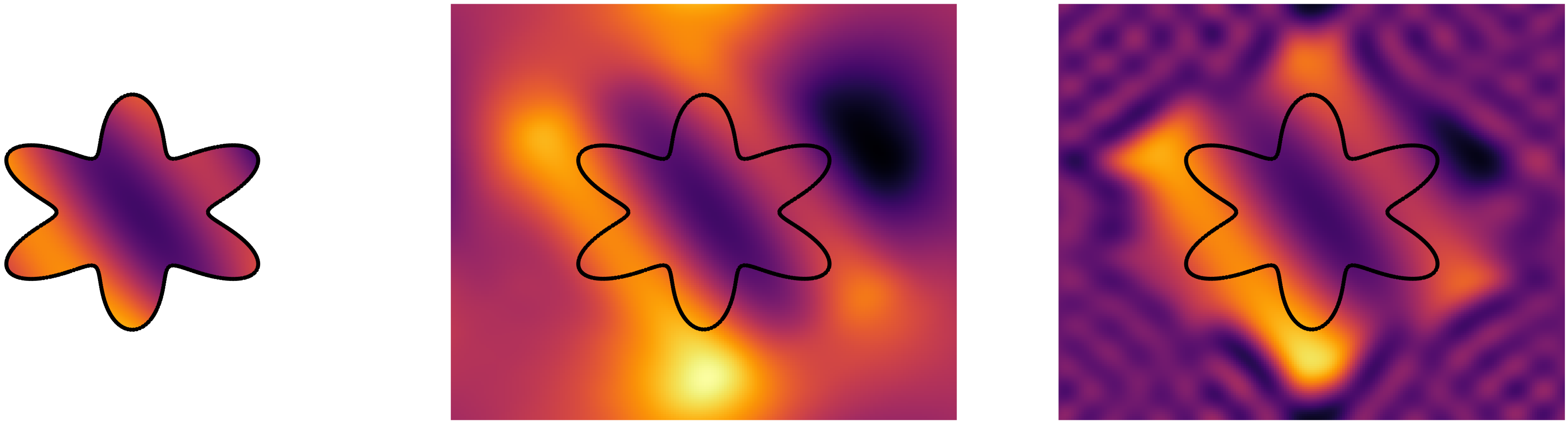
Basis

- Unique representation, straightforward and efficient
- Very good for smooth functions on intervals/rectangles



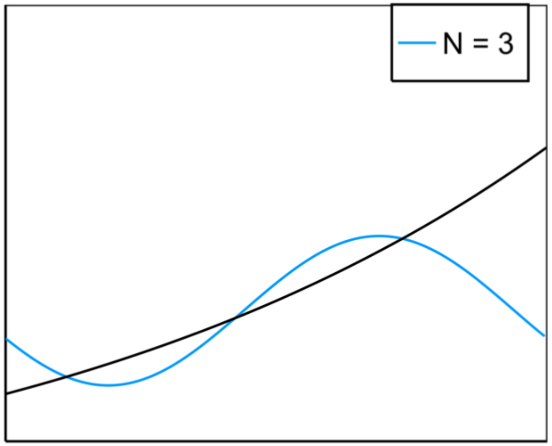
Basis

- Unique representation, straightforward and efficient
- Very good for smooth functions on intervals/rectangles



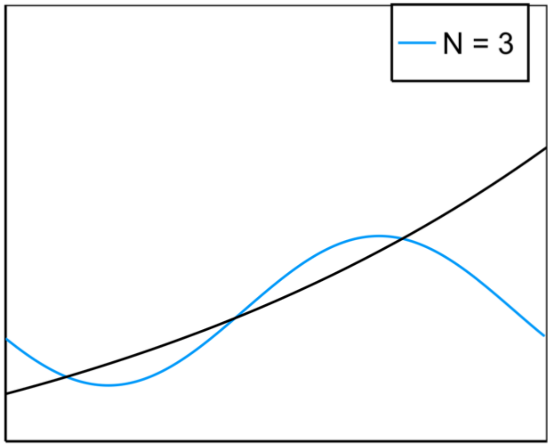
Redundancy helps

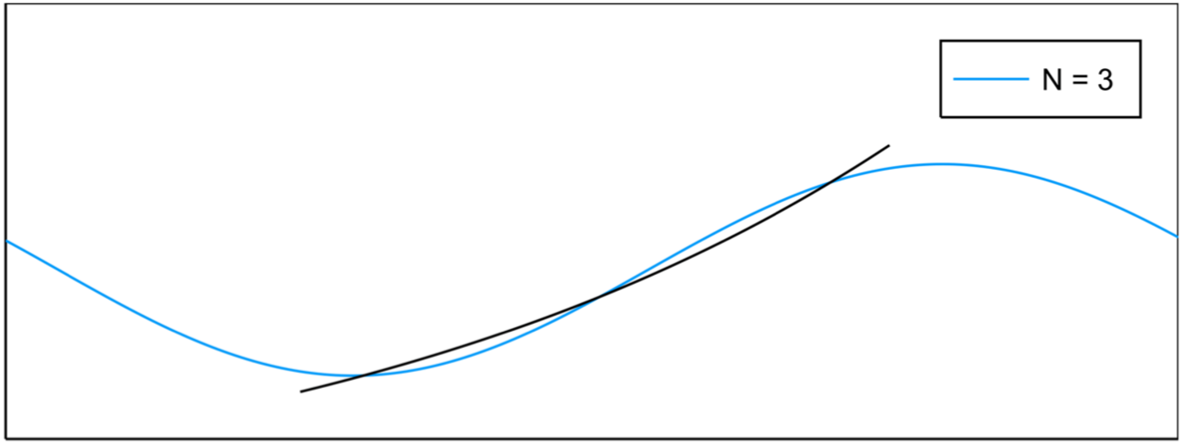
- Endpoints don't match up
- Many functions needed!



Redundancy helps

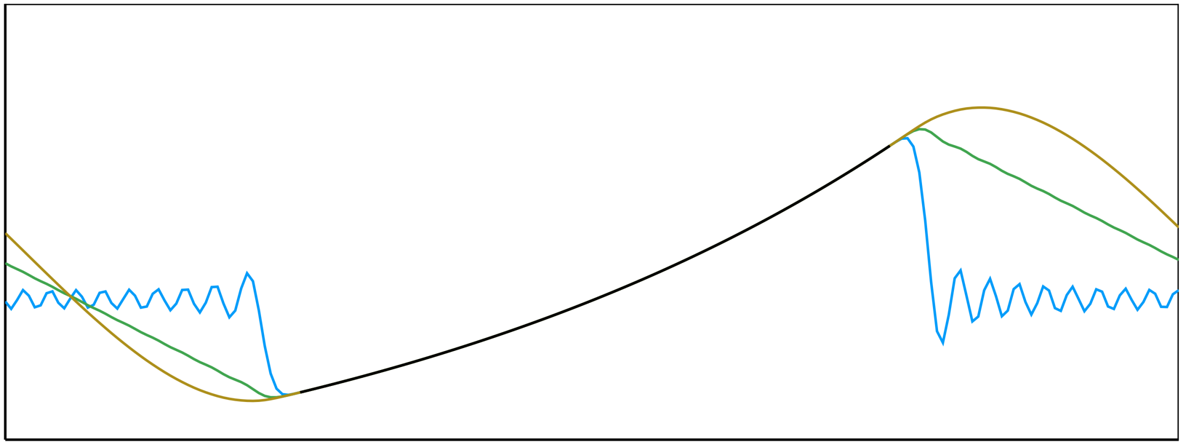
- Endpoints don't match up
- Many functions needed!





Redundancy helps

- Extending the domain → redundancy → accuracy



Frames

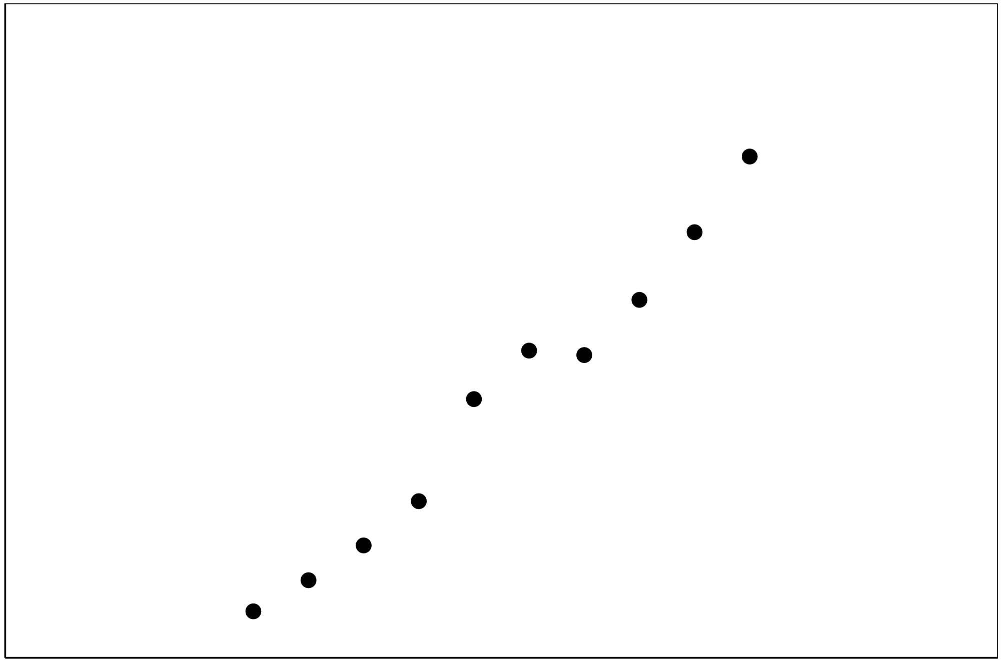
- Orthonormal basis Riesz basis Frame
- Restriction frame
- Sum frame

If solution with reasonable norm exists, it can be found through regularized projection.

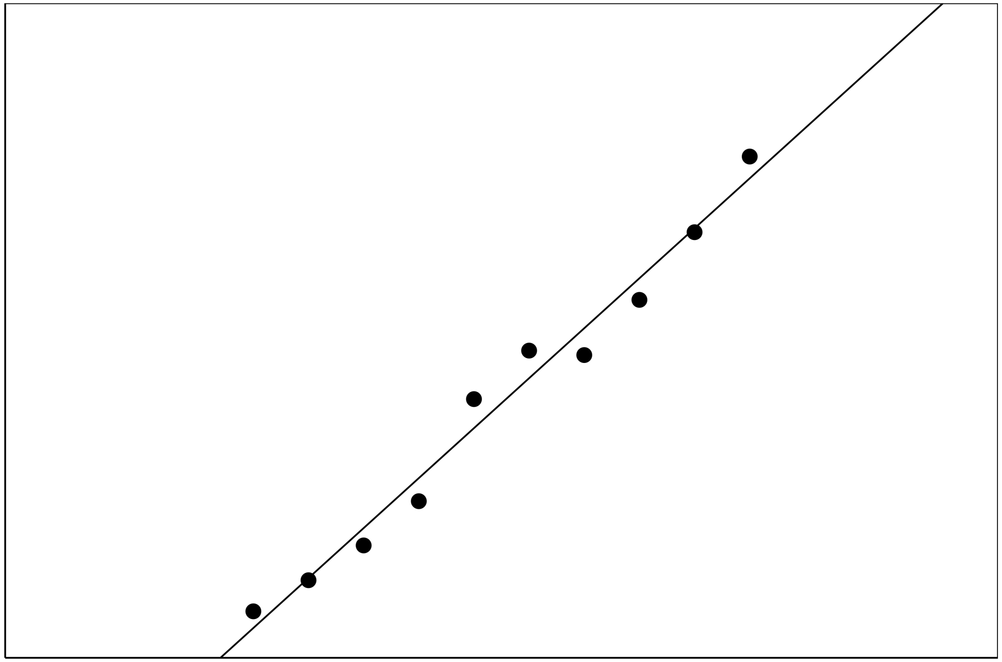
Efficient Algorithms

“Efficiency is intelligent laziness” – D. Dunham

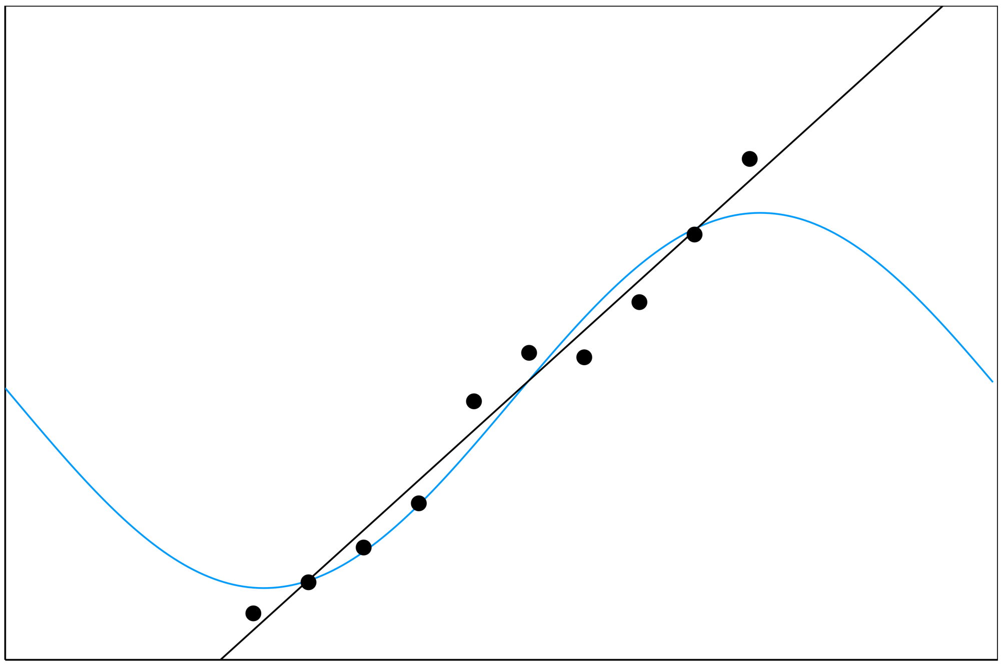
Least squares



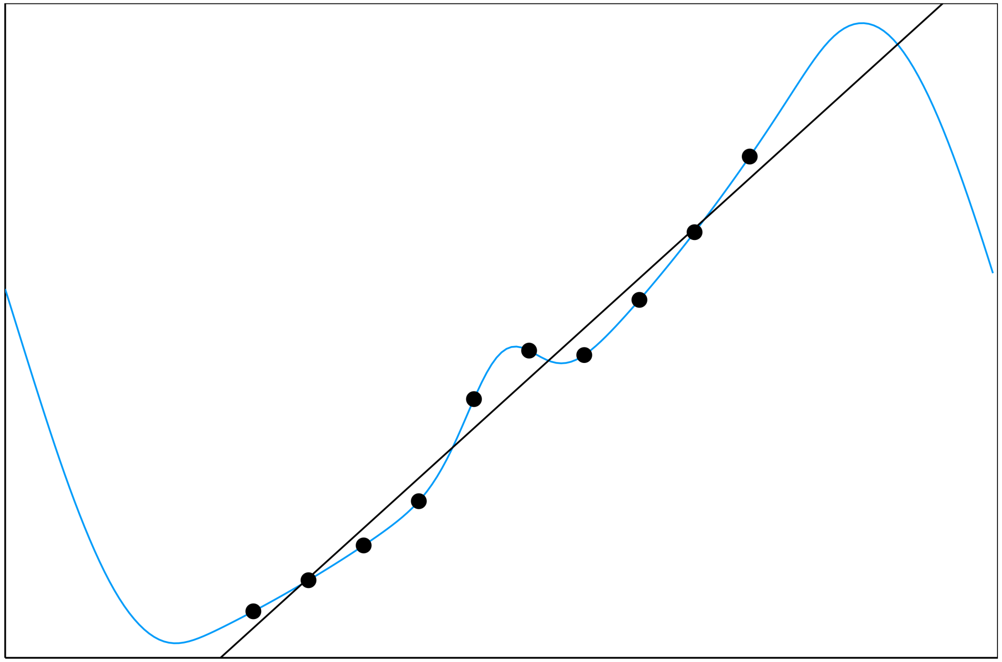
Least squares



Least squares



Least squares



Least squares

- Solve system

$$Ax \approx b$$

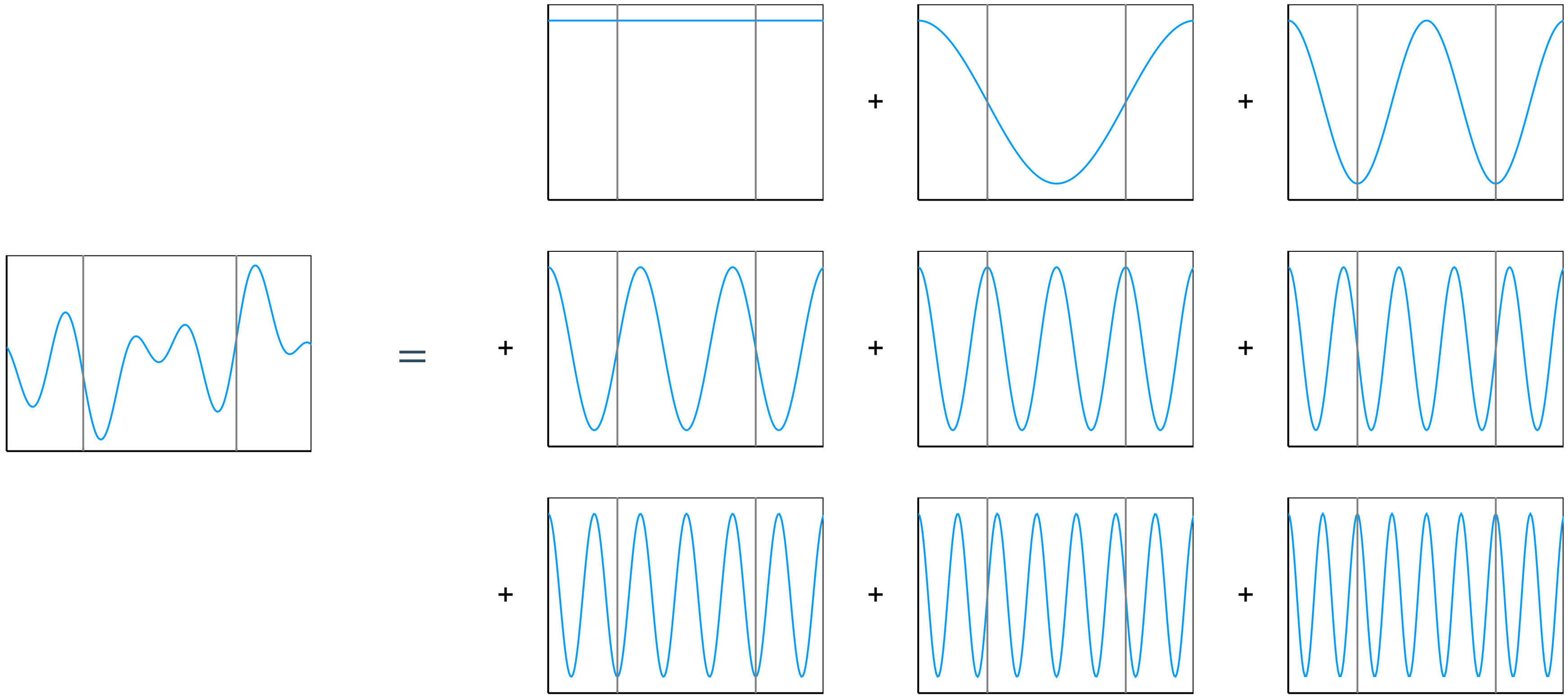
$$A_{i,j} = \varphi_i(x_j), \qquad b_j = f(x_j)$$

- Truncated Singular Value Decomposition

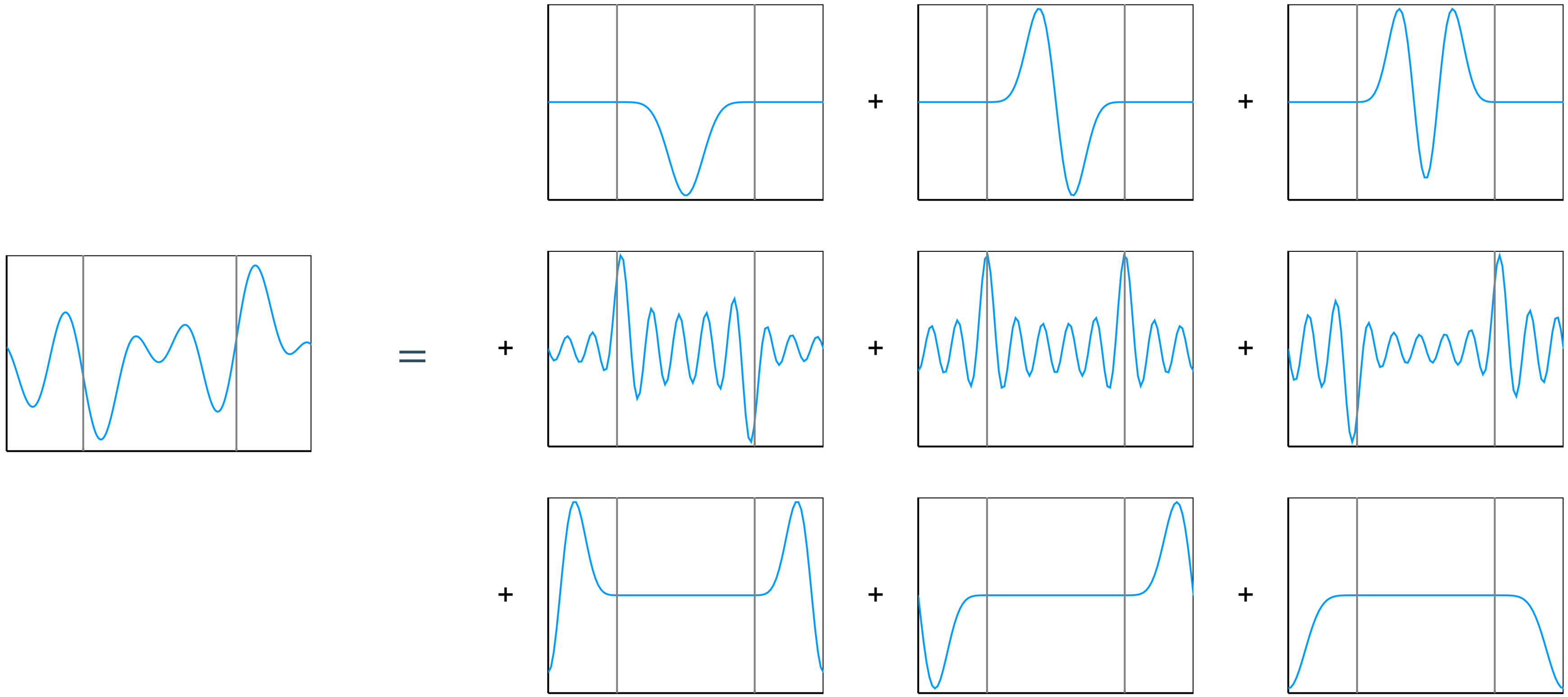
$$A = U_\epsilon \Sigma_\epsilon V_\epsilon^*$$
$$x = V_\epsilon \Sigma_\epsilon^{-1} U_\epsilon b$$

- $O(N^3)$

Rewrite frame



Rewrite frame



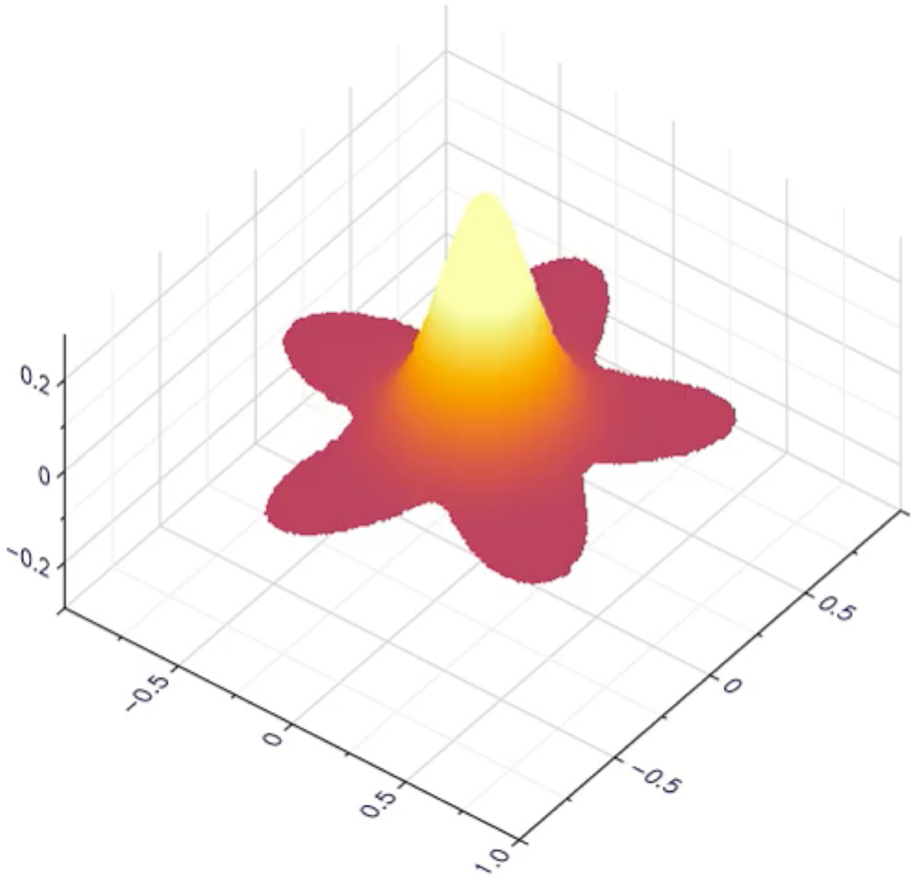
Fast algorithms

- Convert **large difficult** problem into **small difficult** problem and **large easy** problem

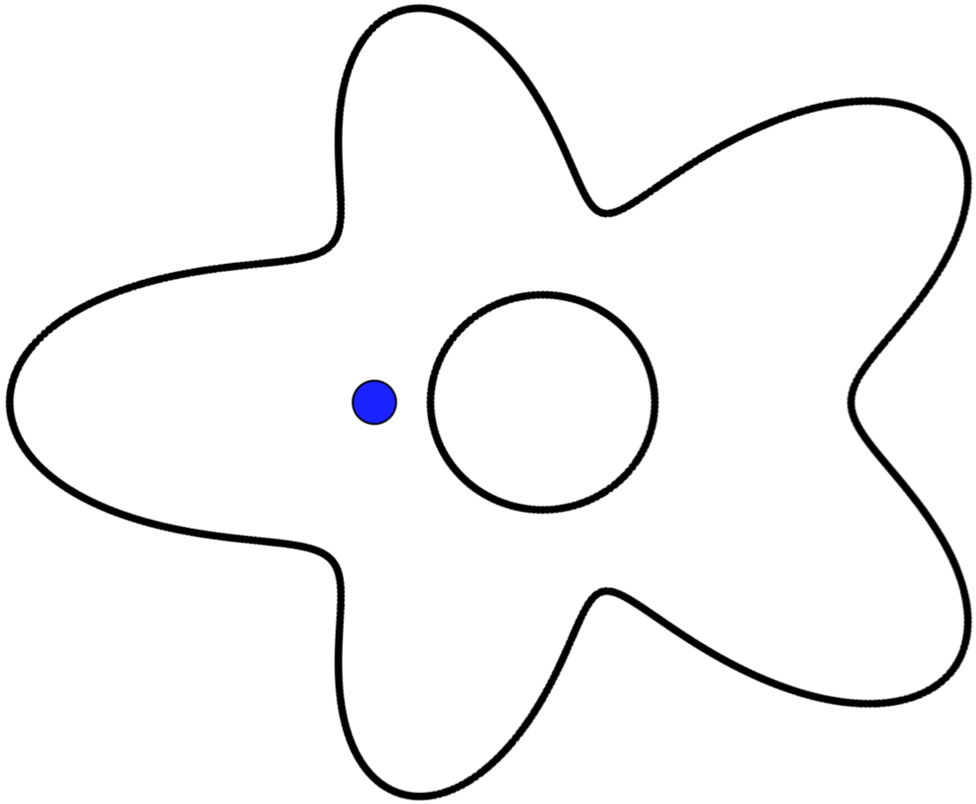
Examples

“I hope there’s pudding!” – J.K. Rowling

Drop of water



WiFi reception



WiFi reception