Multivariate Newton & Lagrange Interpolation

Michael Hecht

MOSAIC Group

MPI-CBG & CSBD & TU Dresden

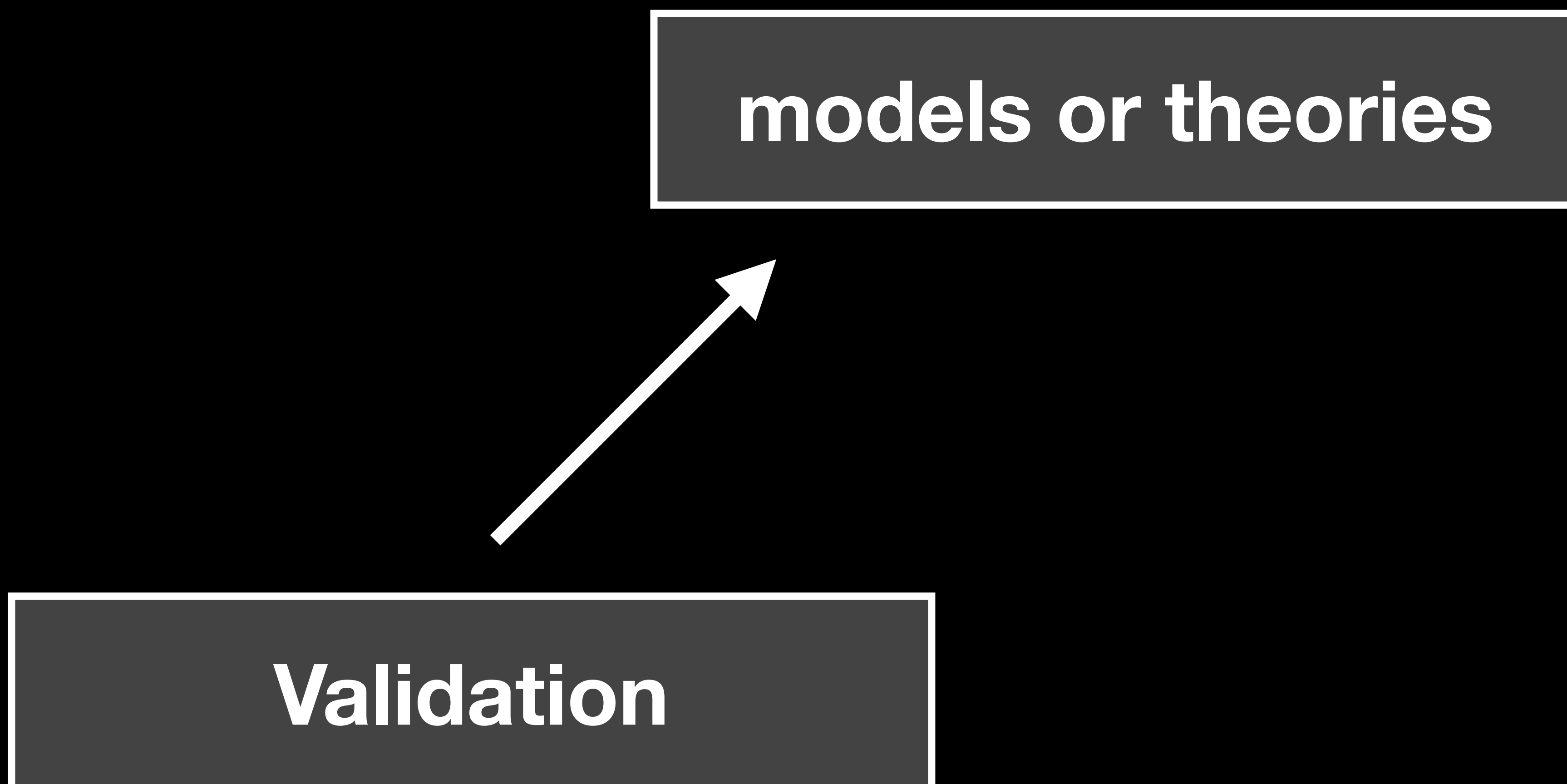
CBG Max Planck Institute of Molecular Cell Biology and Genetics

center for systems biology dresden

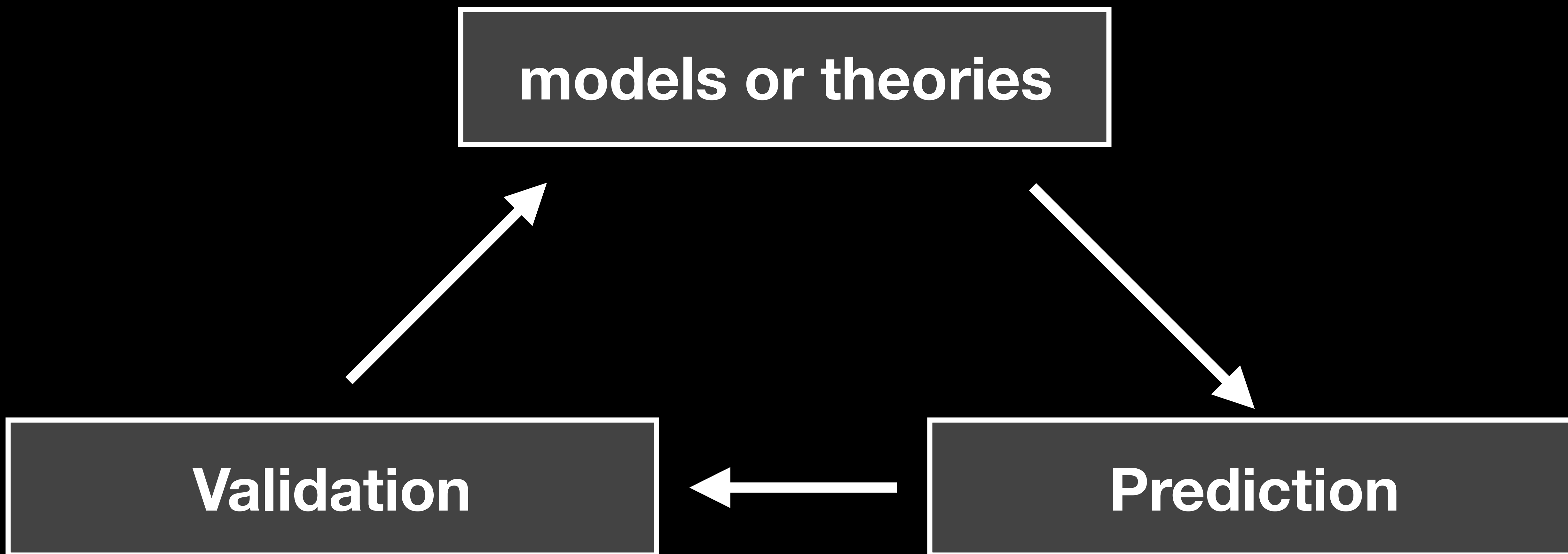
TECHNISCHE UNIVERSITÄT DRESDEN

RESEARCH SEMINAR WU-VIENNA 13.12.19

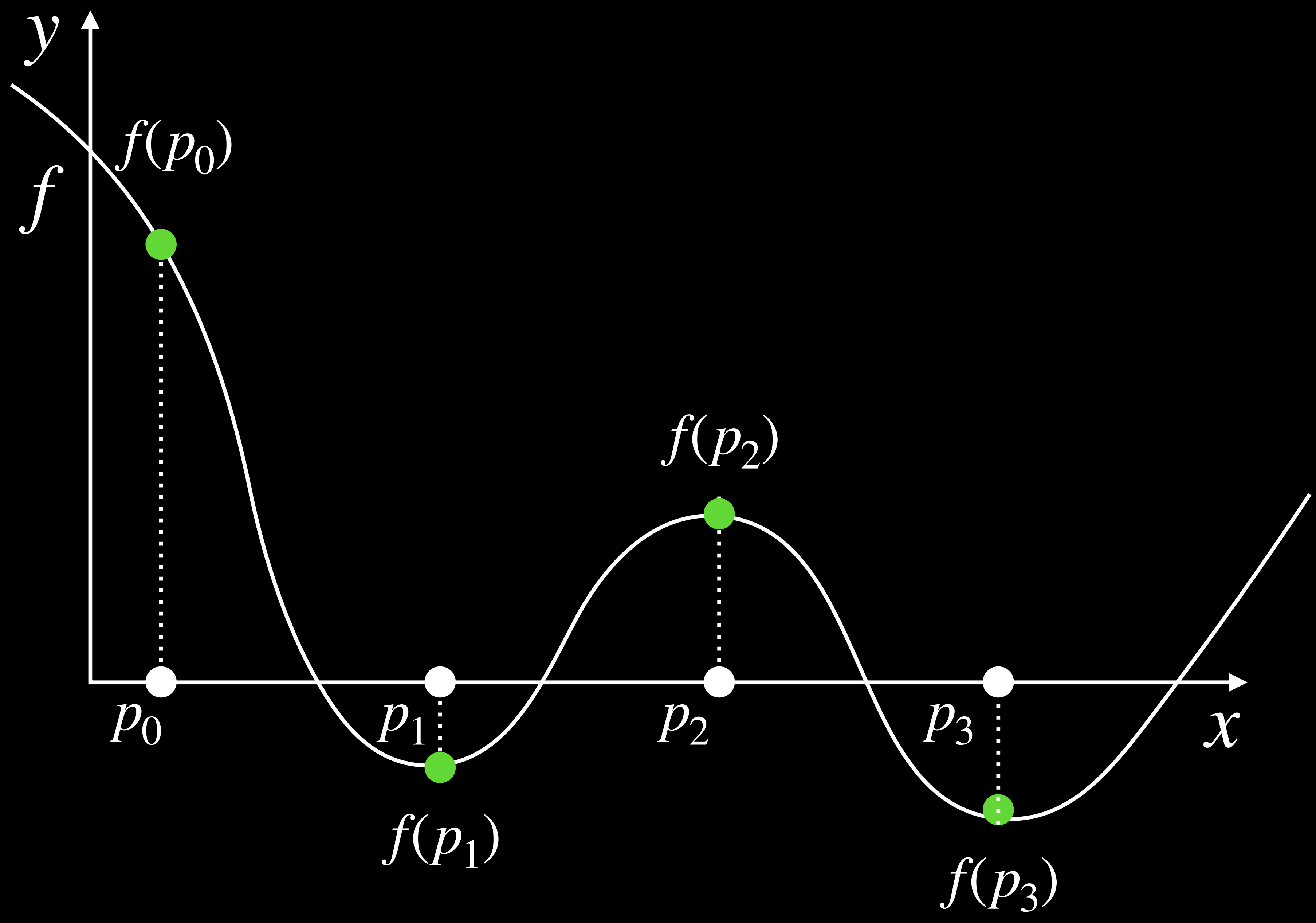
Main principle of Science



Main principle of Science



1D Interpolation



Naive Interpolation

Vandermonde Matrix

$$V_{n,P} = \begin{pmatrix} 1 & p_0 & \cdots & p_0^n \\ \vdots & \vdots & & \vdots \\ 1 & p_n & \cdots & p_n^n \end{pmatrix}, \quad P = \begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}, \quad F = \begin{pmatrix} f(p_0) \\ \vdots \\ f(p_n) \end{pmatrix}$$

$$C = V_{n,P}^{-1} \cdot F, \quad C = (c_0, \ldots, c_n)$$

$$Q_{f,n}(x) = c_0 + c_1 x + \cdots + c_n x^n$$

Runtime $\mathcal{O}(n^3)$ **Storage** $\mathcal{O}(n^2)$ **Evaluation** $\mathcal{O}(n^2)$

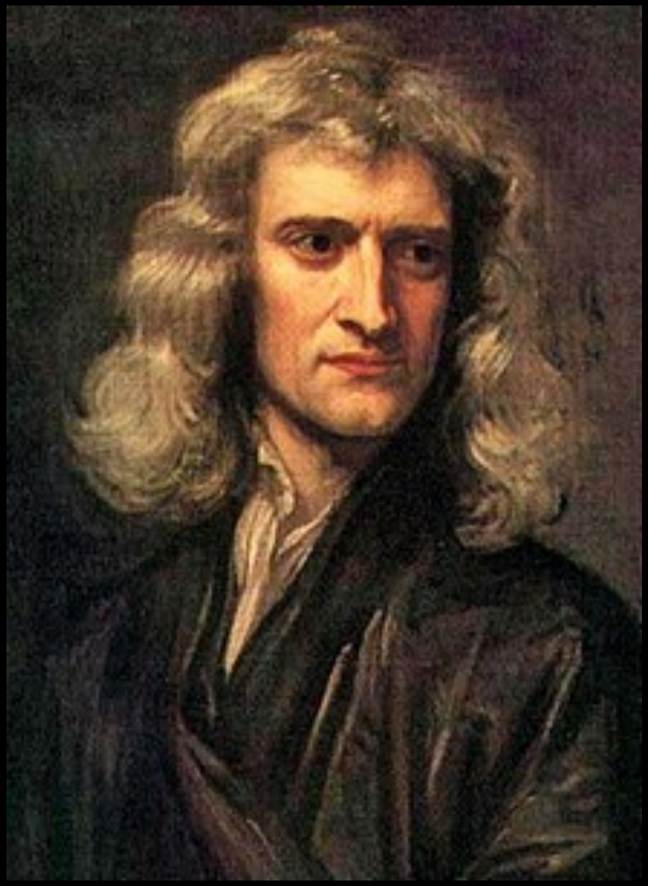
One of the most influential scientists of all time

Sir Isaac Newton
1643-1726

- **Mathematics**
- **Physics**
- **Optics**
- **Computational Sciences**

Philosophiæ Naturalis Principia Mathematica

Two of the most influential scientists of all time



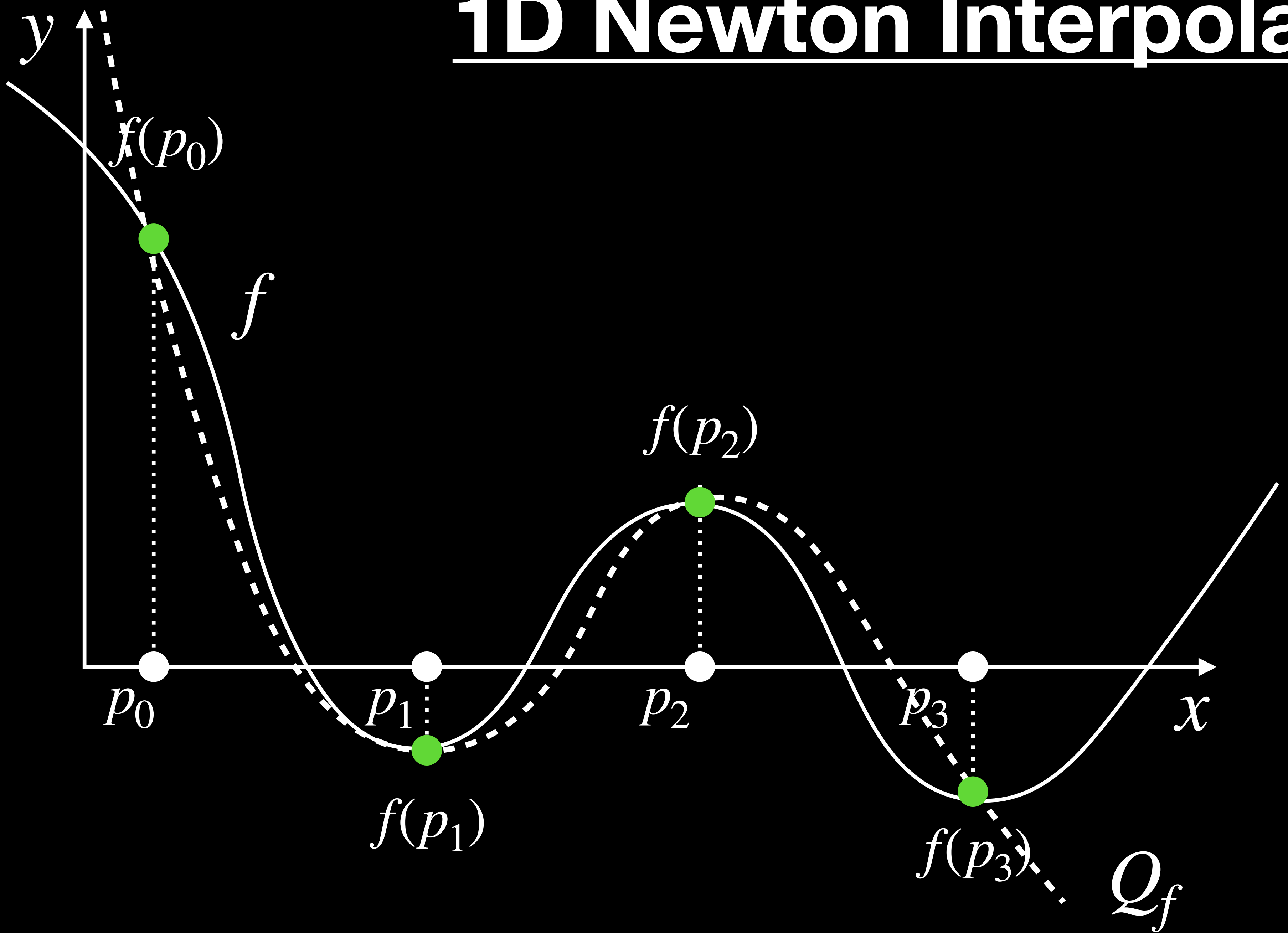
Sir Isaac Newton
1643-1726

Philosophiæ Naturalis Principia Mathematica

Joseph-Louis Lagrange
1736-1813

Mécanique analytique

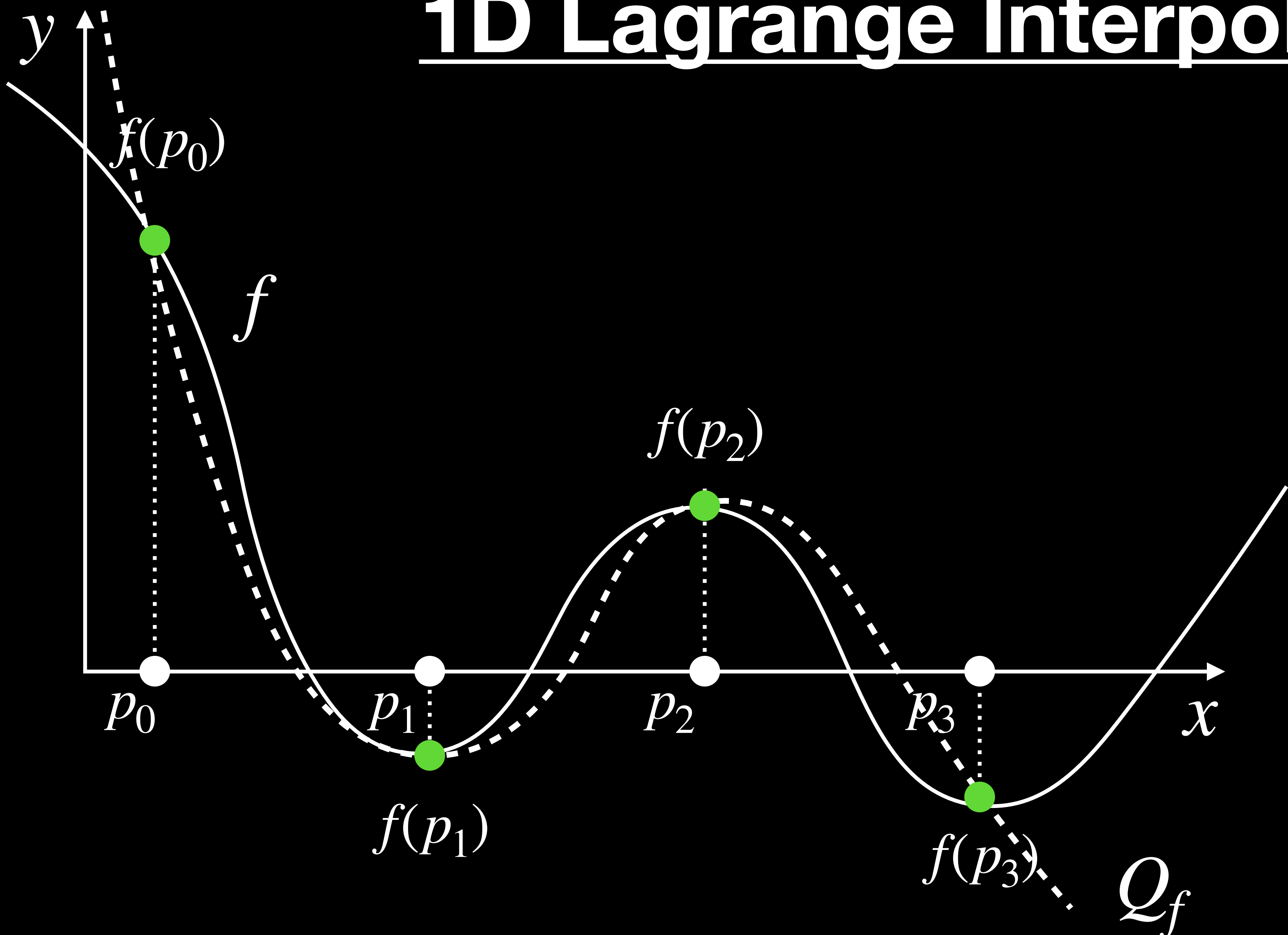
1D Newton Interpolation



Runtime	$\mathcal{O}(n^2)$
Storage	$\mathcal{O}(n)$
Evaluation	$\mathcal{O}(n)$

$$Q_{f,3}(x) = c_0 + c_1(x - p_0) + c_2(x - p_0)(x - p_1) + c_3 \prod_{j=0}^{2} (x - p_j)$$

1D Lagrange Interpolation



Runtime	$\mathcal{O}(n)$
Storage	$\mathcal{O}(n)$
Evaluation	$\mathcal{O}(n)$

$$Q_{f,3}(x) = \prod_{j=0}^{3}(x - p_j)\left(f(p_0)\frac{\omega_0}{x - p_0} + f(p_1)\frac{\omega_1}{x - p_1} + \sum_{i=2}^{n} f(p_i)\frac{\omega_i}{x - p_i}\right)$$

Divided Difference Scheme

Vandermonde Matrix

$$V_{n,P} = \begin{pmatrix} 1 & p_0 & \cdots & p_0^n \\ \vdots & \vdots & & \vdots \\ 1 & p_n & \cdots & p_n^n \end{pmatrix}, \quad P = \begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}, \quad F = \begin{pmatrix} f(p_0) \\ \vdots \\ f(p_n) \end{pmatrix}$$

$$\mathscr{V}_{n,P} : \Pi_n \longrightarrow \mathbb{R}^{n+1}, \quad C \mapsto F, \text{ **s.t.** } C = V_{n,P}^{-1} \cdot F,$$

$$Q_{f,n}(x) = c_0 + c_2 x + \cdots + c_n x^n, \quad F = (Q(p_0), \ldots, Q(p_n))$$

Newton Basis of Π_n

$$N_i(x) = \prod_{j=0}^{i-1} (x - p_j), \quad i = 0, \ldots, n$$

Divided Difference Scheme

Vandermonde Matrix

$$V_{n,P} = \begin{pmatrix} 1 & p_0 & \cdots & p_0^n \\ \vdots & \vdots & & \vdots \\ 1 & p_n & \cdots & p_n^n \end{pmatrix}, \quad P = \begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}, \quad F = \begin{pmatrix} f(p_0) \\ \vdots \\ f(p_n) \end{pmatrix}$$

Vandermonde Matrix w.r.t. Newton Basis

$$W_{n,P} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & (p_1 - p_0) & \cdots & 0 \\ 1 & (p_2 - p_0) & (p_2 - p_0)(p_2 - p_1) & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (p_n - p_0) & \cdots & \prod_{j=0}^{n-1} (p_n - p_j) \end{pmatrix}$$

Divided Difference Scheme

$$[p_0]f := f(p_0)\,, \quad [p_i, \ldots, p_j]f := \frac{[p_i, \ldots, p_{j-1}]f - [p_{i+1}, \ldots, p_j]f}{x_j - x_i}\,,\ j \geq i$$

$$\begin{array}{ccccccccc}
[p_0]f & & & & & & & & \\
 & \searrow & & & & & & & \\
[p_1]f & \rightarrow & [p_0, p_1]f & & & & & & \\
 & \searrow & & \searrow & & & & & \\
[p_2]f & \rightarrow & [p_1, p_2]f & \rightarrow & [p_0, p_1, p_2]f & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & \\
 & \searrow & & \searrow & & & \searrow & & \\
[p_n]f & \rightarrow & [p_{n-1}, p_n]f & \rightarrow & [p_{n-2}, p_{n-1}, p_n]f & \cdots & \rightarrow & [p_0 \ldots p_n]f &
\end{array}$$

Divided Difference Scheme

Vandermonde Matrix

$$V_{n,P} = \begin{pmatrix} 1 & p_0 & \cdots & p_0^n \\ \vdots & \vdots & & \vdots \\ 1 & p_n & \cdots & p_n^n \end{pmatrix}, \quad P = \begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}, \quad F = \begin{pmatrix} f(p_0) \\ \vdots \\ f(p_n) \end{pmatrix}$$

$$\mathscr{V}_{n,P} : \Pi_n \longrightarrow \mathbb{R}^{n+1}, \quad C \mapsto F, \text{ **s.t.** } C = V_{n,P}^{-1} \cdot F,$$

$$Q_{f,n}(x) = c_0 + c_! x + \cdots + c_n x^n, \quad F = (Q(p_0), \ldots, Q(p_n))$$

Lagrange Basis of Π_n

$$L_i(x) = \prod_{j=0, j \neq i}^{n} (x - p_j) \Big/ \prod_{j=0, j \neq i}^{n} (p_i - p_j), \quad i = 0, \ldots, n$$

Divided Difference Scheme

Vandermonde Matrix

$$V_{n,P} = \begin{pmatrix} 1 & p_0 & \cdots & p_0^n \\ \vdots & \vdots & & \vdots \\ 1 & p_n & \cdots & p_n^n \end{pmatrix}, \quad P = \begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}, \quad F = \begin{pmatrix} f(p_0) \\ \vdots \\ f(p_n) \end{pmatrix}$$

$$\mathscr{V}_{n,P} : \Pi_n \longrightarrow \mathbb{R}^{n+1}, \quad C \mapsto F, \text{ **s.t.** } C = V_{n,P}^{-1} \cdot F,$$

$$Q_{f,n}(x) = c_0 + c_1 x + \cdots + c_n x^n, \quad F = (Q(p_0), \ldots, Q(p_n))$$

Lagrange Basis of Π_n

$$L_i(p_j) = \delta_{i,j}$$

Divided Difference Scheme

Vandermonde Matrix

$$V_{n,P} = \begin{pmatrix} 1 & p_0 & \cdots & p_0^n \\ \vdots & \vdots & & \vdots \\ 1 & p_n & \cdots & p_n^n \end{pmatrix}, \quad P = \begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}, \quad F = \begin{pmatrix} f(p_0) \\ \vdots \\ f(p_n) \end{pmatrix}$$

Vandermonde Matrix w.r.t. Lagrange Basis

$$W_{n,P} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 0 & 1 & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Classical Lagrange Form

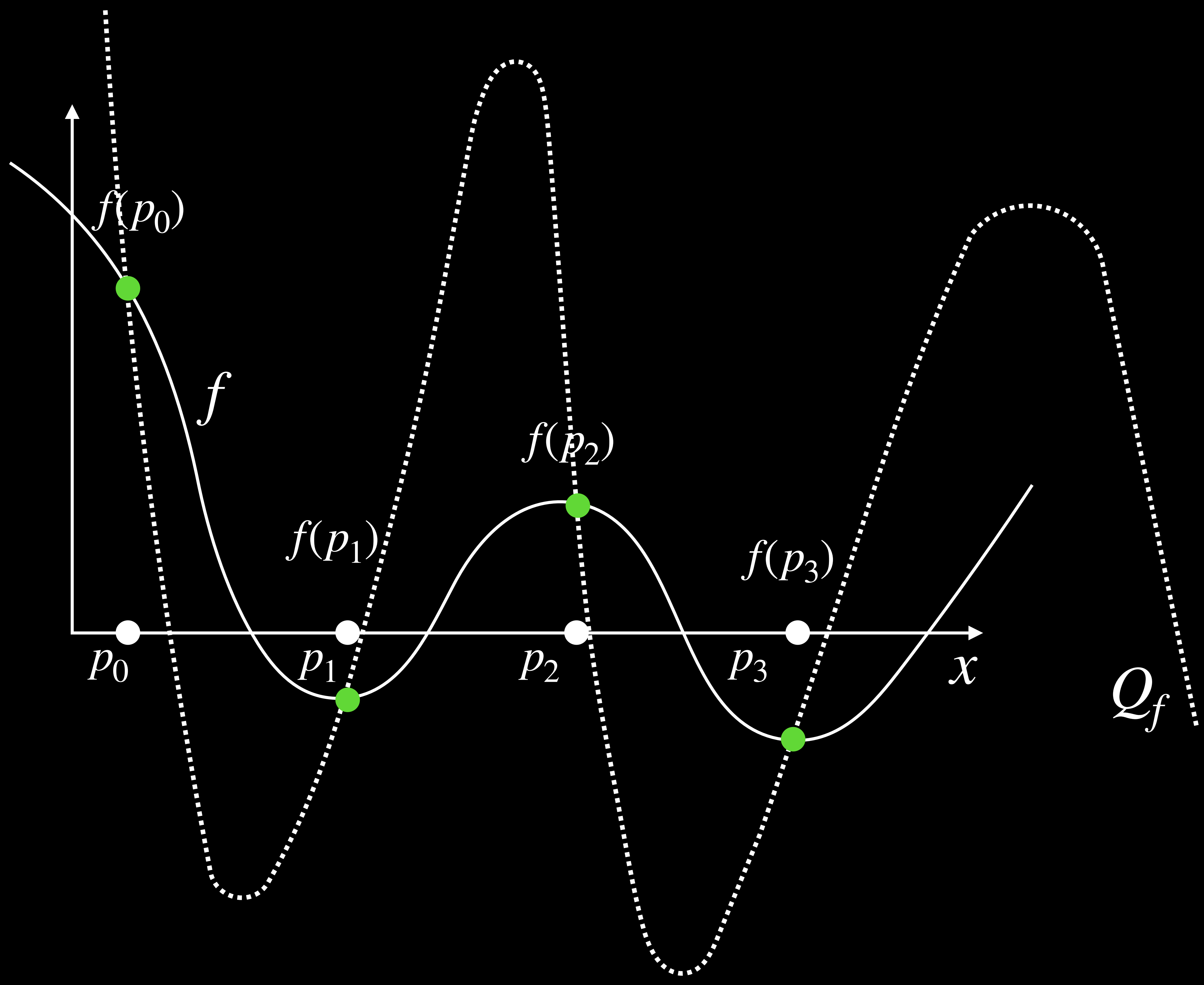
$$Q_{f,n}(x) = \sum_{i=0}^{n} f(p_i) L_i(x)$$

Barycentrical Lagrange Form

$$Q_{f,n}(x) = \prod_{j=0}^{n} (x - p_j) \sum_{i=0}^{n} f(p_i) \frac{\omega_i}{x - p_i}$$

Runtime	$\mathcal{O}(n)$
Storage	$\mathcal{O}(n)$
Evaluation	$\mathcal{O}(n)$

Runge's Phenomena



Approximation Theory

$$|f(x) - Q_{f,n}(x)| \leq \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - p_i), \quad x, \xi_x \in \Omega$$

Approximation Theory

$$|f(x) - Q_{f,n}(x)| \leq \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - p_i)\,, \quad x, \xi_x \in \Omega$$

Chebyshev nodes

$$\mathrm{Cheb}_n = \left\{ \cos\left(\frac{2k+1}{2(n+1)}\pi\right), k = 0, \ldots, n \right\}$$

Approximation Theory

$$|f(x) - Q_{f,n}(x)| \leq \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - p_i), \quad x, \xi_x \in \Omega$$

Chebyshev nodes

$$\mathrm{Cheb}_n = \left\{ \cos\left(\frac{2k+1}{2(n+1)} \pi \right), k = 0, \ldots, n \right\}$$

$$|f(x) - Q_f(x)| \leq \frac{f^{(n+1)}(\xi_x)}{2^n (n+1)!}, \qquad x, \xi_x \in \Omega$$

Approximation Theory

$$|f(x) - Q_{f,n}(x)| \leq \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - p_i), \quad x, \xi_x \in \Omega$$

Chebyshev nodes

$$\text{Cheb}_n = \left\{ \cos\left(\frac{2k+1}{2(n+1)} \pi \right), k = 0, \ldots, n \right\}$$

$$|f(x) - Q_f(x)| \leq \frac{f^{(n+1)}(\xi_x)}{2^n (n+1)!}, \qquad x, \xi_x \in \Omega$$

Approximation

$$Q_{f,n} \xrightarrow[n \to \infty]{} f, \quad \forall f \in H^1(\Omega, \mathbb{R})$$

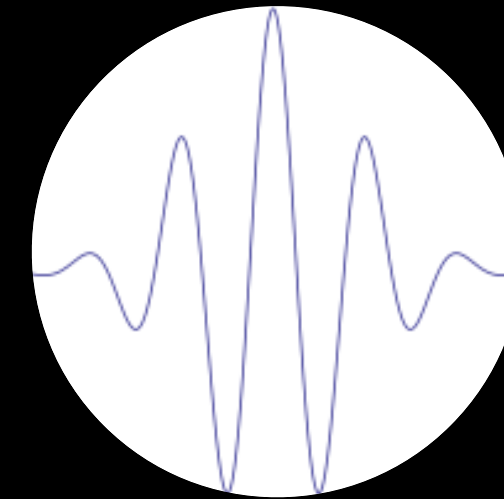
Computer … someone who computes

Complex computations

- physics & astrophysics
- engineering
- aeronautics
- economics
- etc

NACA High Speed Computer Room (1949)

The Curse of Dimensionality



Newton / Lagrange Interpolation

$$Q_f(x) = \sum_{i=0}^{n} c_i N_i(x) = \sum_{i=0}^{n} d_i L_i(x)$$

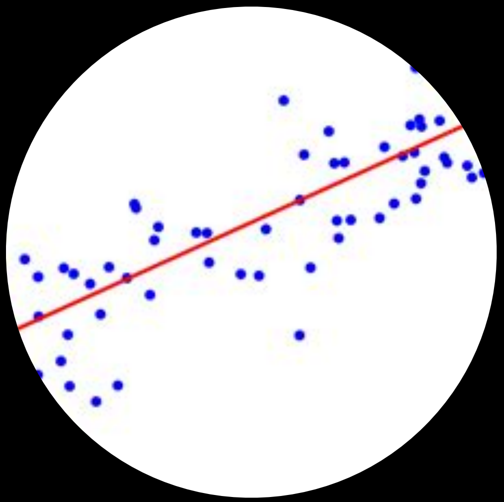
Spline/Wavelet Interpolation & FFTs

$$Q_f(x) = \sum_{p \in G} c_p \gamma_p(x)$$

- Numerically accurate & fast $\mathcal{O}(n^2)/\mathcal{O}(n)$
- Convergence to the ground truth $Q_{f,n} \xrightarrow[n \to \infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Only in 1D

- Fast Runtime $\mathcal{O}(\log(N)N)\,, N = r^m$
- Convergence to ground truth $Q_{f,n} \xrightarrow[n \to \infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Feasible in low dimensions

The Curse of Dimensionality



Newton / Lagrange Interpolation

$$Q_f(x) = \sum_{i=0}^{n} c_i N_i(x) = \sum_{i=0}^{n} d_i L_i(x)$$

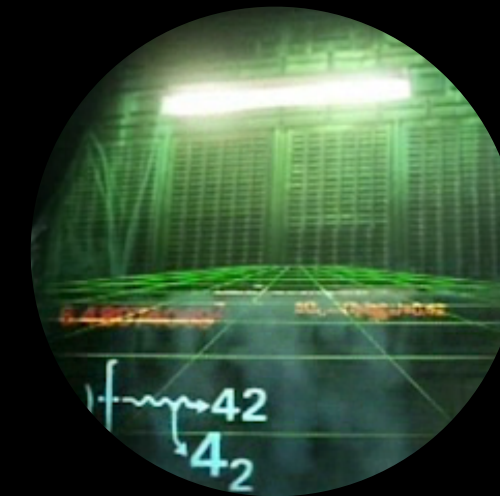
- Numerically accurate & fast $\mathcal{O}(n^2)/\mathcal{O}(n)$
- Convergence to the ground truth $Q_{f,n} \xrightarrow[n\to\infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Only in 1D

Linear Regression in mD

$$Q_f(x) \approx c_0 + \sum_{i=1}^{m} c_i x_i$$

- Numerically accurate & fast
- Only linear approximation
- Interpolant is easy to understand
- Allows further analysis/computation
- Feasible in high dimensions

The Curse of Dimensionality

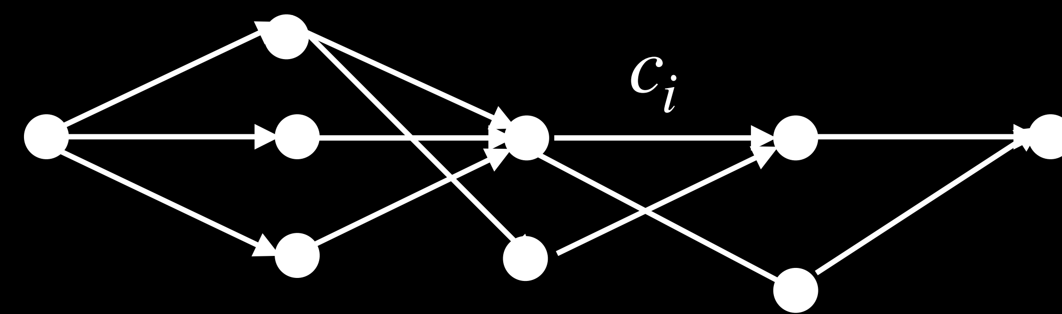


Newton / Lagrange Interpolation

Machine Learning

$$Q_f(x) = \sum_{i=0}^{n} c_i N_i(x) = \sum_{i=0}^{n} d_i L_i(x)$$

$Q_f(x) =$



- Numerically accurate & fast $\mathcal{O}(n^2)/\mathcal{O}(n)$
- Convergence to the ground truth $Q_{f,n} \xrightarrow[n\to\infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Only in 1D

- Numerically accurate & fast
- No Convergence to ground truth $Q_{f,n} \nrightarrow f$
- Interpolant is hard to understand
- Hampers further analysis/computation
- Feasible in high dimensions

Multivariate Polynomial Interpolation

Multivariate Vandermonde Matrix

$$V_{m,n,P} = \begin{pmatrix} 1 & p_{0,1} & p_{0,2} & \cdots & p_{0,m} & p_0^{\odot 2} & \cdots & p_0^{\odot n} \\ 1 & & \vdots & & & \vdots & \ddots & \vdots \\ 1 & p_{N,1} & p_{N,2} & \cdots & p_{N,m} & p_N^{\odot 2} & \cdots & p_N^{\odot n} \end{pmatrix}, \quad C = V_{m,n,P}^{-1} F$$

Unisolvent Nodes

How to choose P such that $V_{m,n,P}$ becomes (numerically) invertible ?

Multivariate Polynomial Interpolation

Multivariate Vandermonde Matrix

$$V_{m,n,P} = \begin{pmatrix} 1 & p_{0,1} & p_{0,2} & \cdots & p_{0,m} & p_0^{\odot 2} & \cdots & p_0^{\odot n} \\ 1 & & \vdots & & & \vdots & \ddots & \vdots \\ 1 & p_{N,1} & p_{N,2} & \cdots & p_{N,m} & p_N^{\odot 2} & \cdots & p_N^{\odot n} \end{pmatrix}, \quad N = \binom{m+n}{n}$$

- Numerically in-accurate & slow $\mathcal{O}(N^3)$
- No Convergence to the ground truth $Q_{f,n} \nrightarrow f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Only in low dimensions

Coefficients in normal form

$$C = V_{m,n,P}^{-1} F\,,$$

$$F = \big(f(p_0), \ldots, f(p_{N(m,n)})\big)$$

$$N \in \mathcal{O}(m^n)$$

Multivariate Polynomial Interpolation

Multivariate Vandermonde Matrix

$$V_{m,n,P} = \begin{pmatrix} 1 & p_{0,1} & p_{0,2} & \cdots & p_{0,m} & p_0^{\odot 2} & \cdots & p_0^{\odot n} \\ 1 & & \vdots & & & \vdots & \ddots & \vdots \\ 1 & p_{N,1} & p_{N,2} & \cdots & p_{N,m} & p_N^{\odot 2} & \cdots & p_N^{\odot n} \end{pmatrix}, \quad V_{m,n,P}C = 0$$

Unisolvent Nodes

$$\ker V_{m,n,P} = 0 \Leftrightarrow \nexists Q \in \Pi_{m,n} \setminus \{0\}, \quad Q(P) = 0$$

Multivariate Polynomial Interpolation

Multivariate Vandermonde Matrix

$$V_{m,n,P} = \begin{pmatrix} 1 & p_{0,1} & p_{0,2} & \cdots & p_{0,m} & p_0^{\odot 2} & \cdots & p_0^{\odot n} \\ 1 & & \vdots & & & \vdots & \ddots & \vdots \\ 1 & p_{N,1} & p_{N,2} & \cdots & p_{N,m} & p_N^{\odot 2} & \cdots & p_N^{\odot n} \end{pmatrix}, \quad C = V_{m,n,P}^{-1} F$$

Unisolvent Nodes

$$\ker V_{m,n,P} = 0 \Leftrightarrow \nexists Q \in \Pi_{m,n} \setminus \{0\}, \quad Q(P) = 0$$

$$|f(x) - Q_f(x)| \leq \frac{\partial_{x_i}^{\alpha_i+1} f(\xi_x)}{2^{\alpha_i}(n+1)!}, \qquad x, \xi_x \in \Omega, \alpha \in \mathbb{N}^m, |\alpha| = n$$

Multivariate Polynomial Interpolation

Multivariate Vandermonde Matrix

$$V_{m,n,P} = \begin{pmatrix} 1 & p_{0,1} & p_{0,2} & \cdots & p_{0,m} & p_0^{\odot 2} & \cdots & p_0^{\odot n} \\ 1 & & \vdots & & & \vdots & \ddots & \vdots \\ 1 & p_{N,1} & p_{N,2} & \cdots & p_{N,m} & p_N^{\odot 2} & \cdots & p_N^{\odot n} \end{pmatrix}, \quad C = V_{m,n,P}^{-1} F$$

Unisolvent Nodes

$$\ker V_{m,n,P} = 0 \Leftrightarrow \nexists Q \in \Pi_{m,n} \setminus \{0\}, \quad Q(P) = 0$$

Approximation

$$Q_f \xrightarrow[n \to \infty]{} f, \quad \forall f \in H^k(\Omega, \mathbb{R}), k > m/2$$

Multivariate Polynomial Interpolation

Multivariate Vandermonde Matrix

$$V_{m,n,P} = \begin{pmatrix} 1 & p_{0,1} & p_{0,2} & \cdots & p_{0,m} & p_0^{\odot 2} & \cdots & p_0^{\odot n} \\ 1 & & \vdots & & & \vdots & \ddots & \vdots \\ 1 & p_{N,1} & p_{N,2} & \cdots & p_{N,m} & p_N^{\odot 2} & \cdots & p_N^{\odot n} \end{pmatrix}, \quad C = V_{m,n,P}^{-1} F$$

Unisolvent Nodes

$$\ker V_{m,n,P} = 0 \Leftrightarrow \nexists Q \in \Pi_{m,n} \setminus \{0\}, \quad Q(P) = 0$$

Approximation

$$Q_f \xrightarrow[n \to \infty]{} f, \quad \forall f \in H^k(\Omega, \mathbb{R})\,, k > m/2$$

Multivariate Newton Basis

Runtime $\mathcal{O}(N^2)$ **Storage** $\mathcal{O}(N)$

Theorem 1

Let $m, n \in \mathbb{N}$ and $H \subseteq \mathbb{R}^m$ be a hyperplane of co-dimension 1.

- If $P_1 \subseteq H$ is unisolvent w.r.t. $m-1, n$ on H
- $P_2 \subseteq \mathbb{R}^m \backslash H$ is unisolvent w.r.t. $m, n-1$ on $\mathbb{R}^m$

Then $P = P_1 \cup P_2$ is unisolvent w.r.t. m, n on $\mathbb{R}^m$.

Theorem 1

Let $m, n \in \mathbb{N}$ **and** $H \subseteq \mathbb{R}^m$ **be a hyperplane of co-dimension 1.**

- **If** $P_1 \subseteq H$ **is unisolvent w.r.t.** $m-1, n$ **on** H
- $P_2 \subseteq \mathbb{R}^m \backslash H$ **is unisolvent w.r.t.** $m, n-1$ **on** $\mathbb{R}^m$

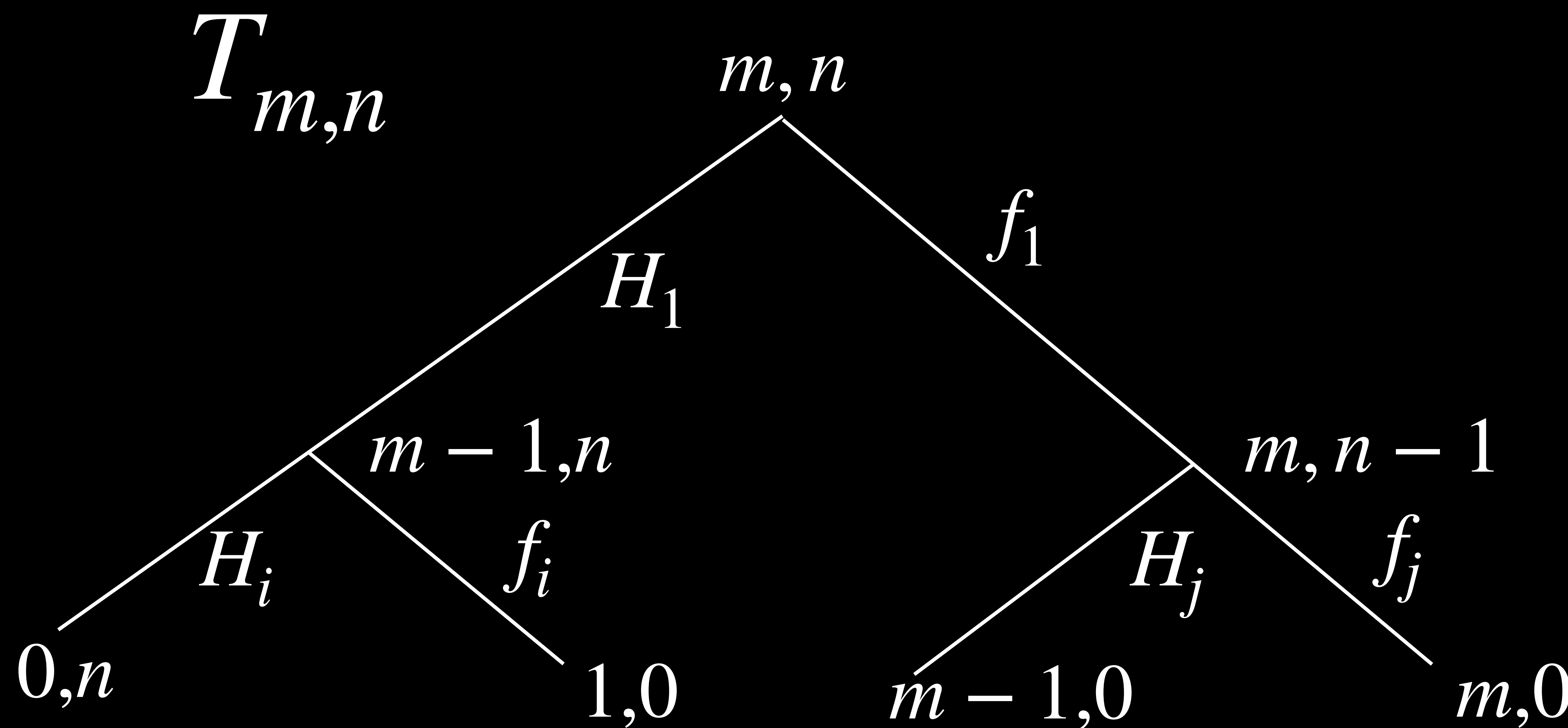
Then $P = P_1 \cup P_2$ **is unisolvent w.r.t.** m, n **on** $\mathbb{R}^m$**.**

Theorem 2

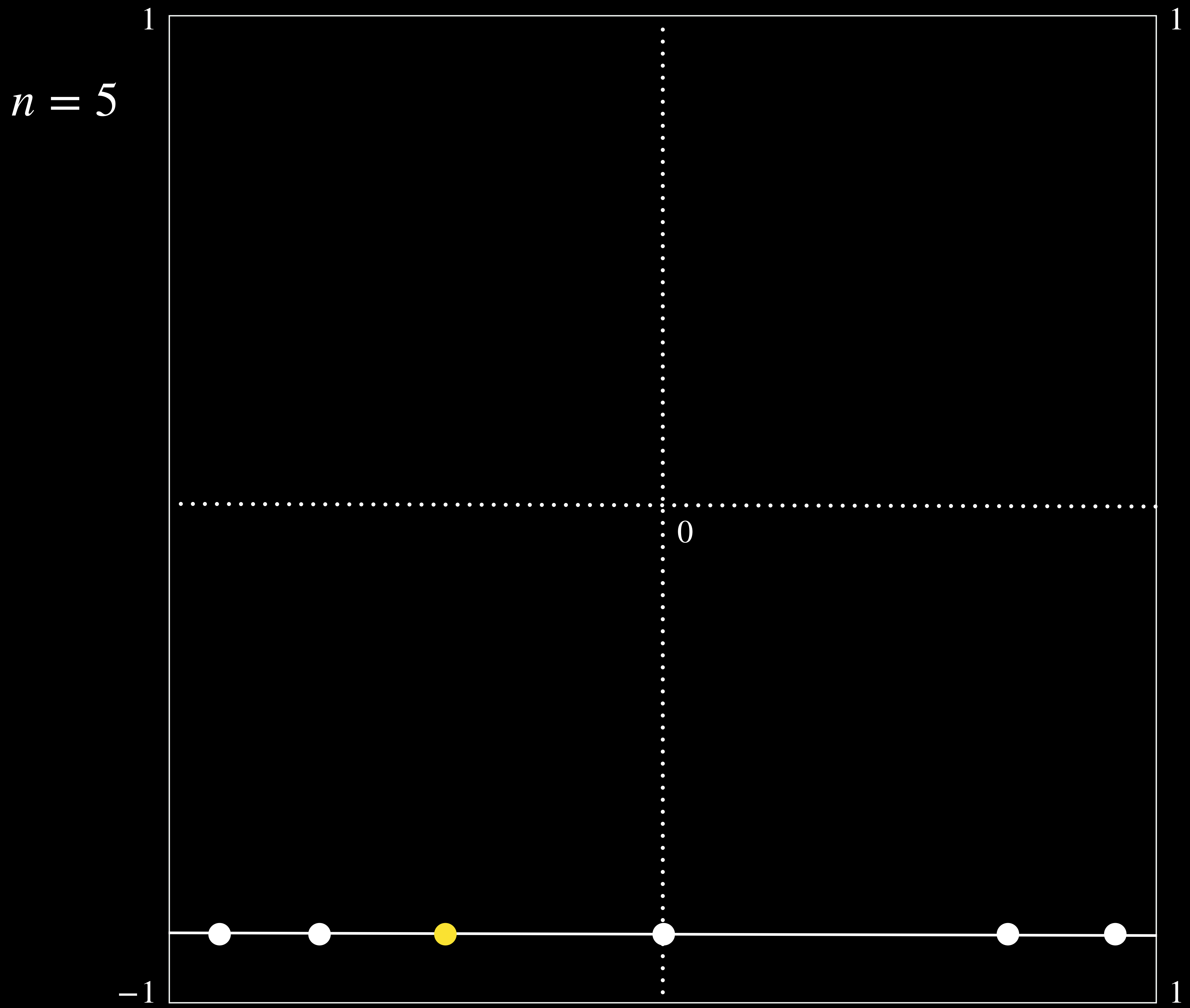
Let $m, n \in \mathbb{N}$ **,** $f : [-1,1]^m \longrightarrow \mathbb{R}$ **be a function,** $H = Q_H^{-1}(0)$ **be a hyperplane.**

- **If** Q_1 **fits** f **w.r.t.** $(P_1, m-1, n)$ **on** H
- Q_2 **fits** $f_1 = (f - Q_1)/Q_H$, $Q_H^{-1}(0) = H$ **w.r.t.** $P_2, m, n-1$ **on** $\mathbb{R}^m$

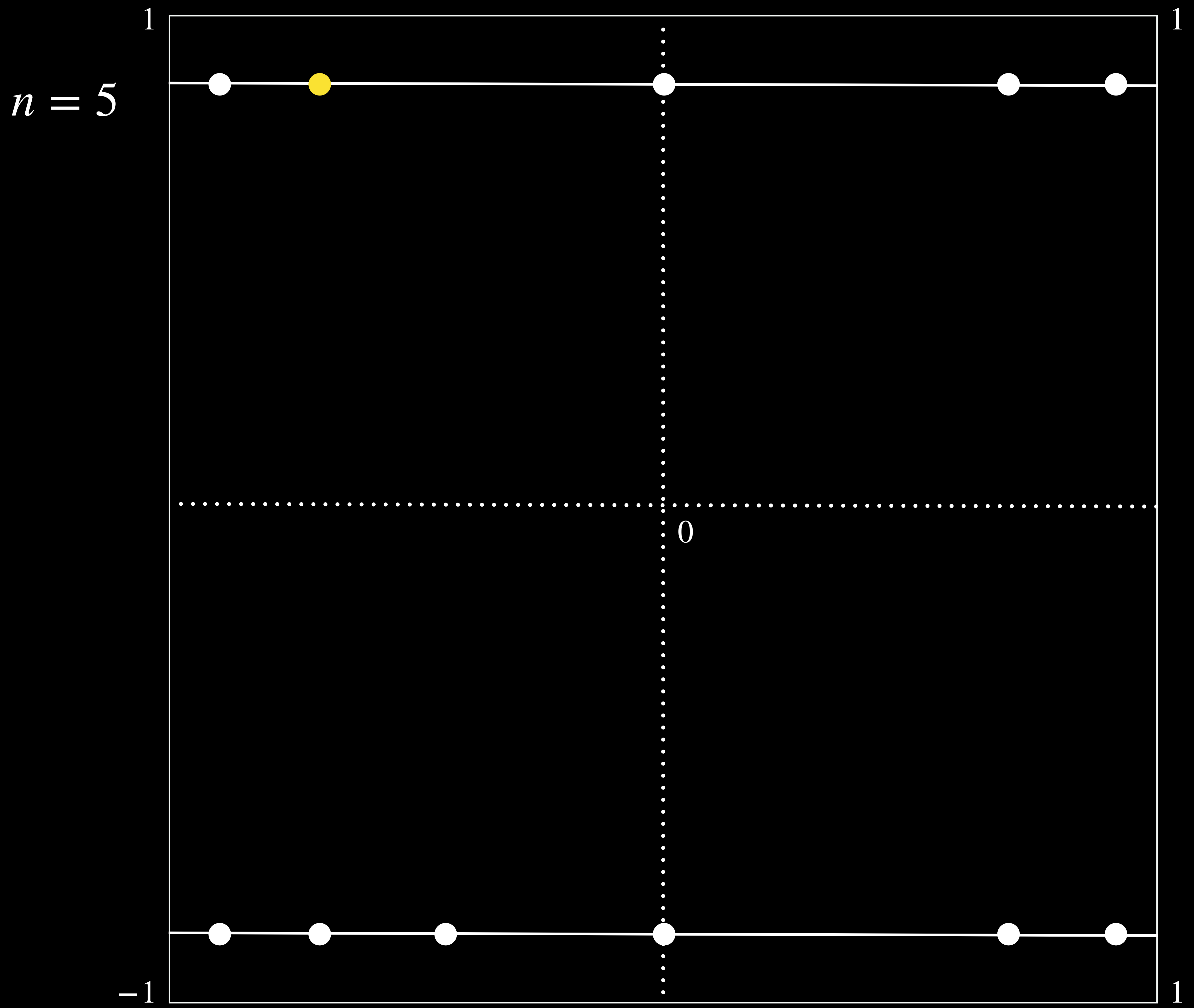
Then $Q_{m,n,f} = Q_1 + Q_H Q_2$ **fits** f **w.r.t.** P, m, n **on** $\mathbb{R}^m$



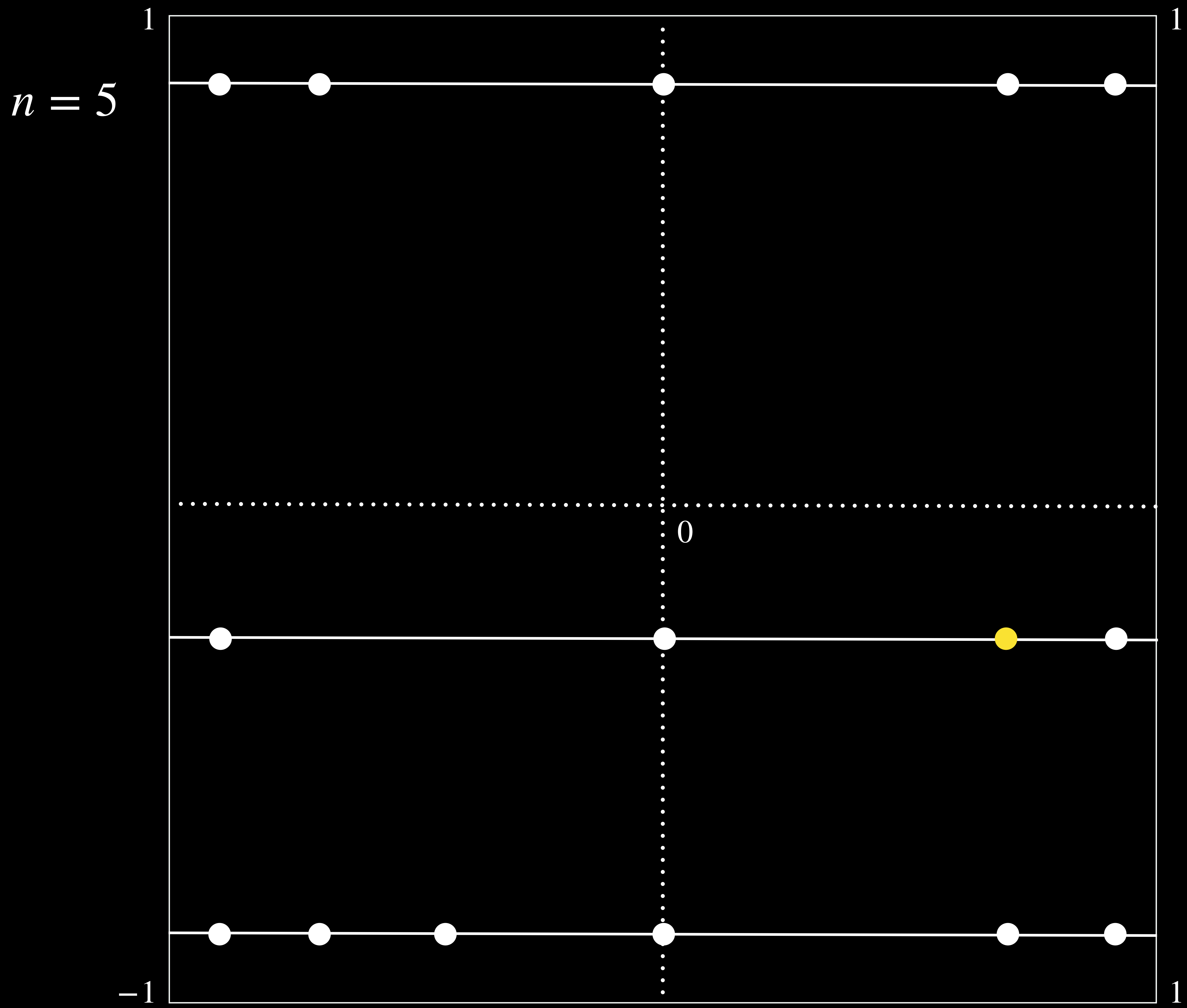
- **Recursive Decomposition of the Interpolation Problem**
- **In 1D this yields exactly the classical Newton Interpolation**
- **Runtime** $\mathcal{O}(N^2)\,, N \in \mathcal{O}(m^n)$



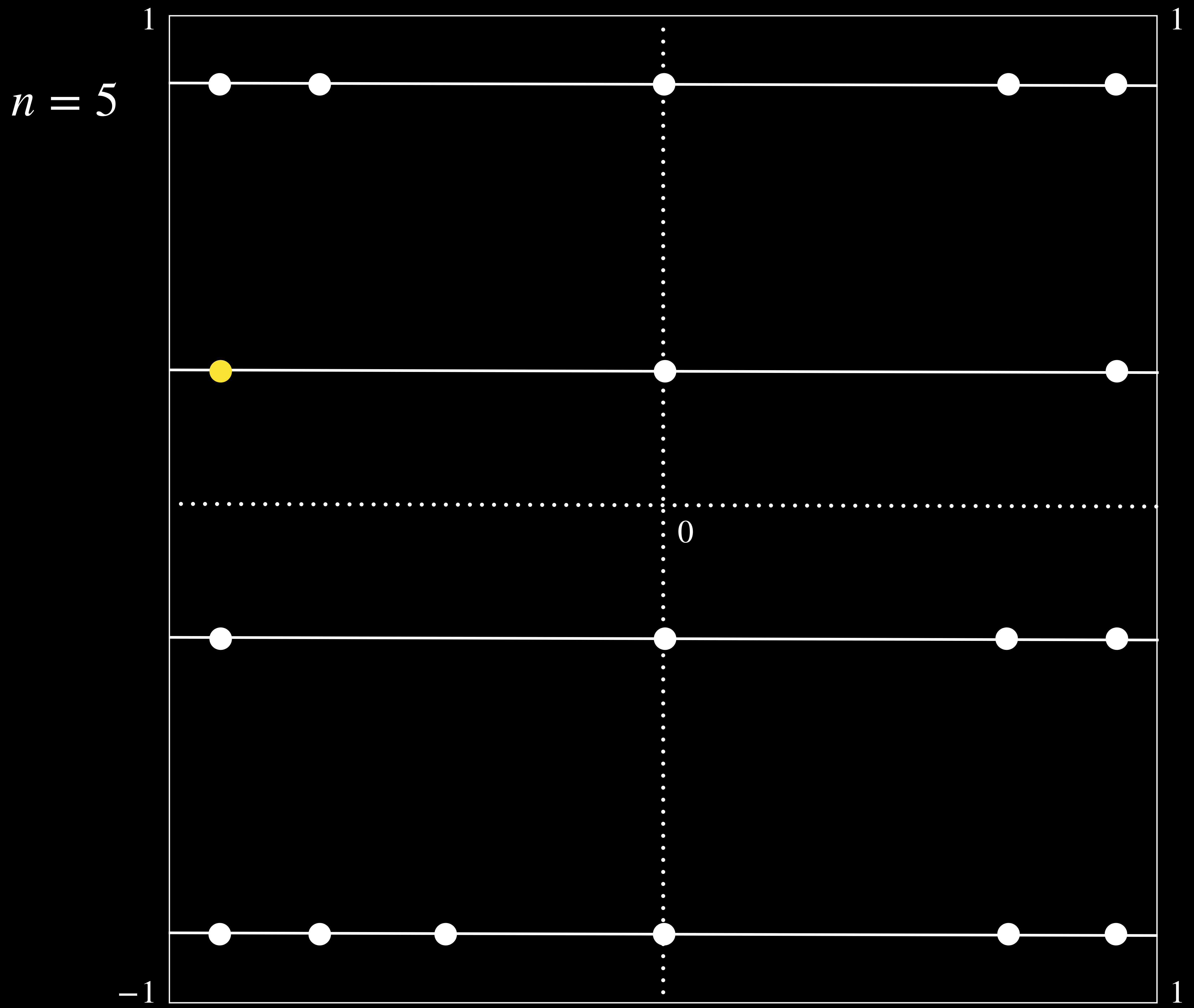
$n = 5$
1
1
0
−1
1



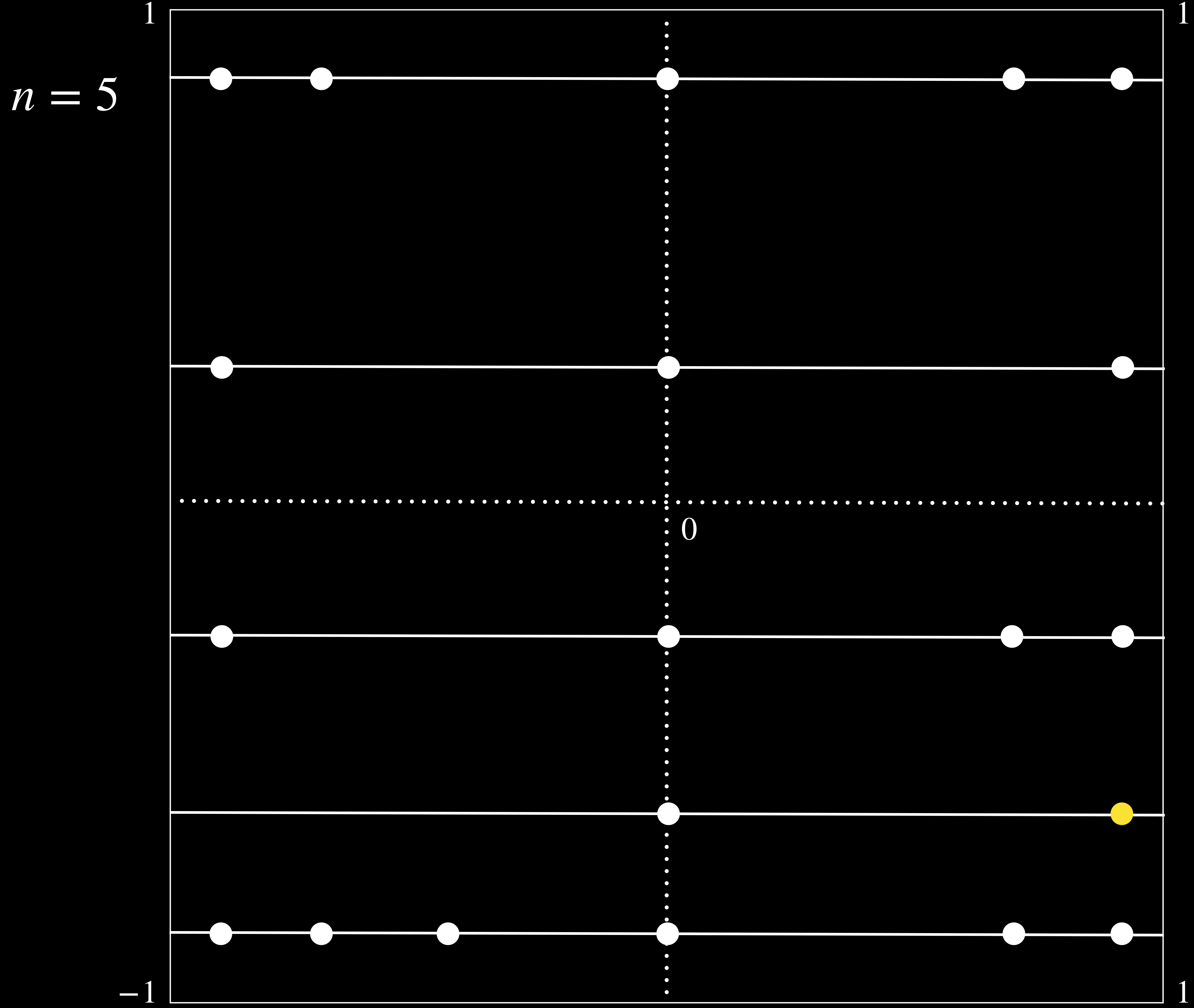
n = 5
1
1
0
−1
1



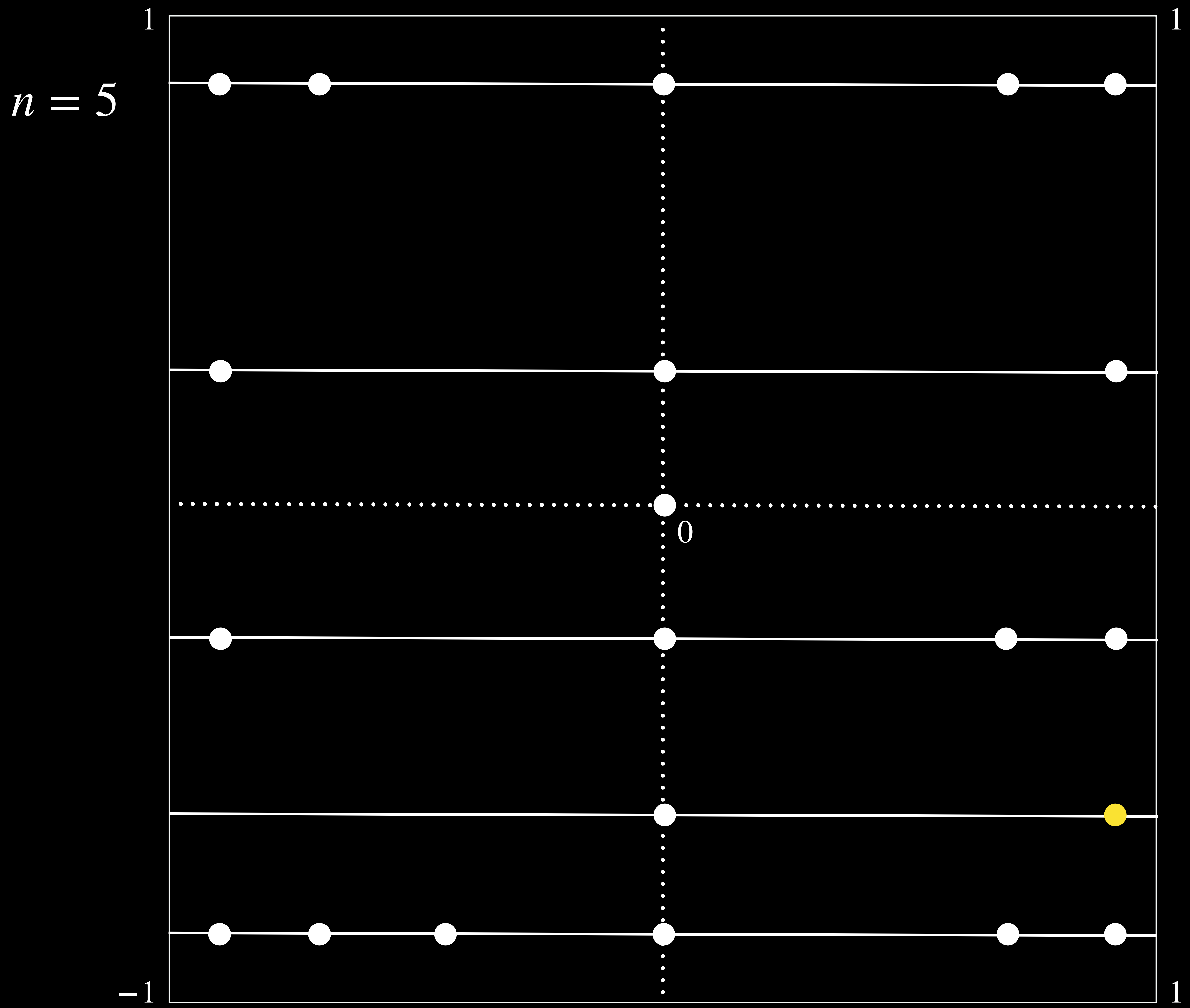
$n = 5$
1
1
0
−1
1



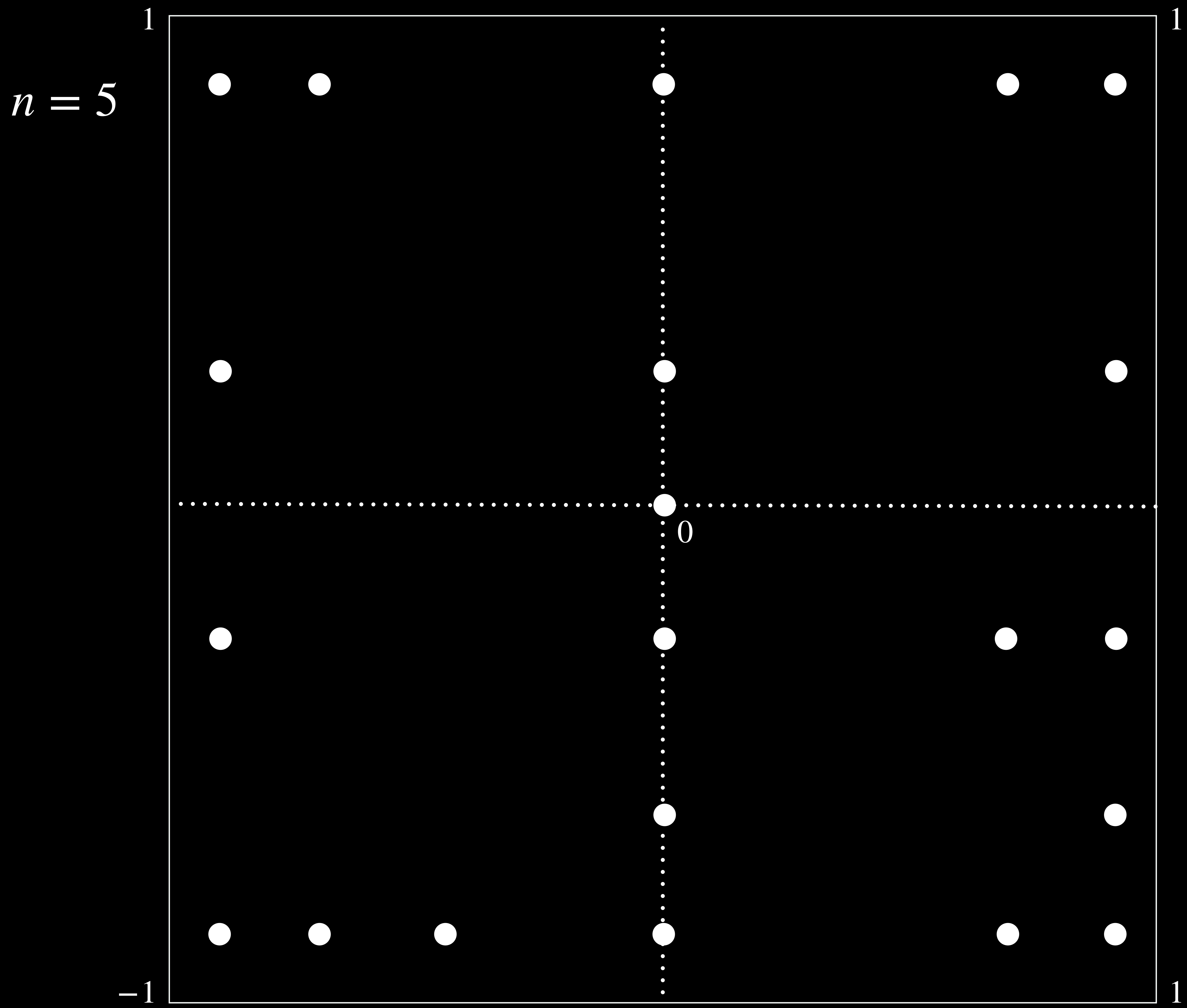
$n = 5$
1
1
0
−1
1



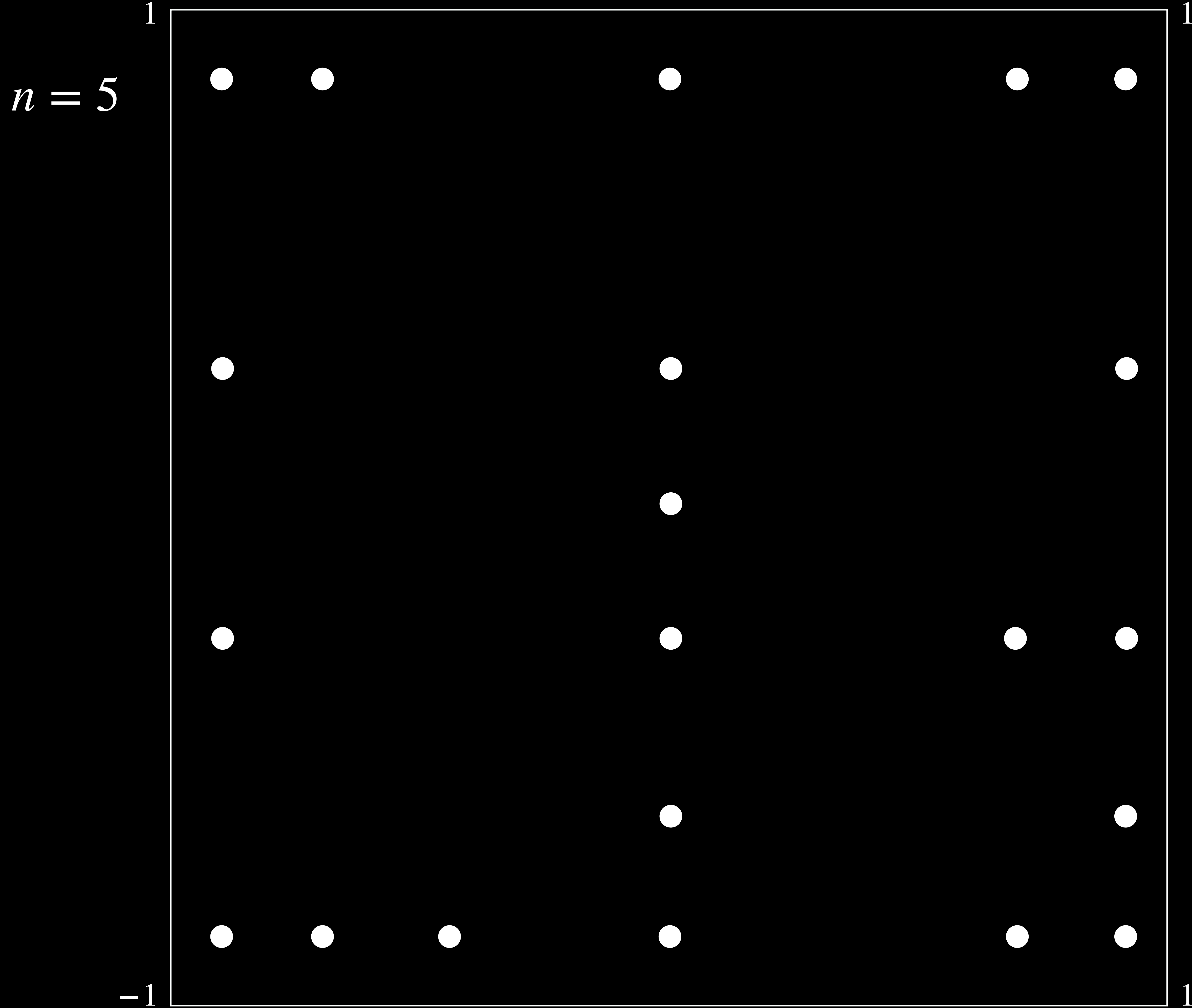
n = 5
1
1
0
−1
1



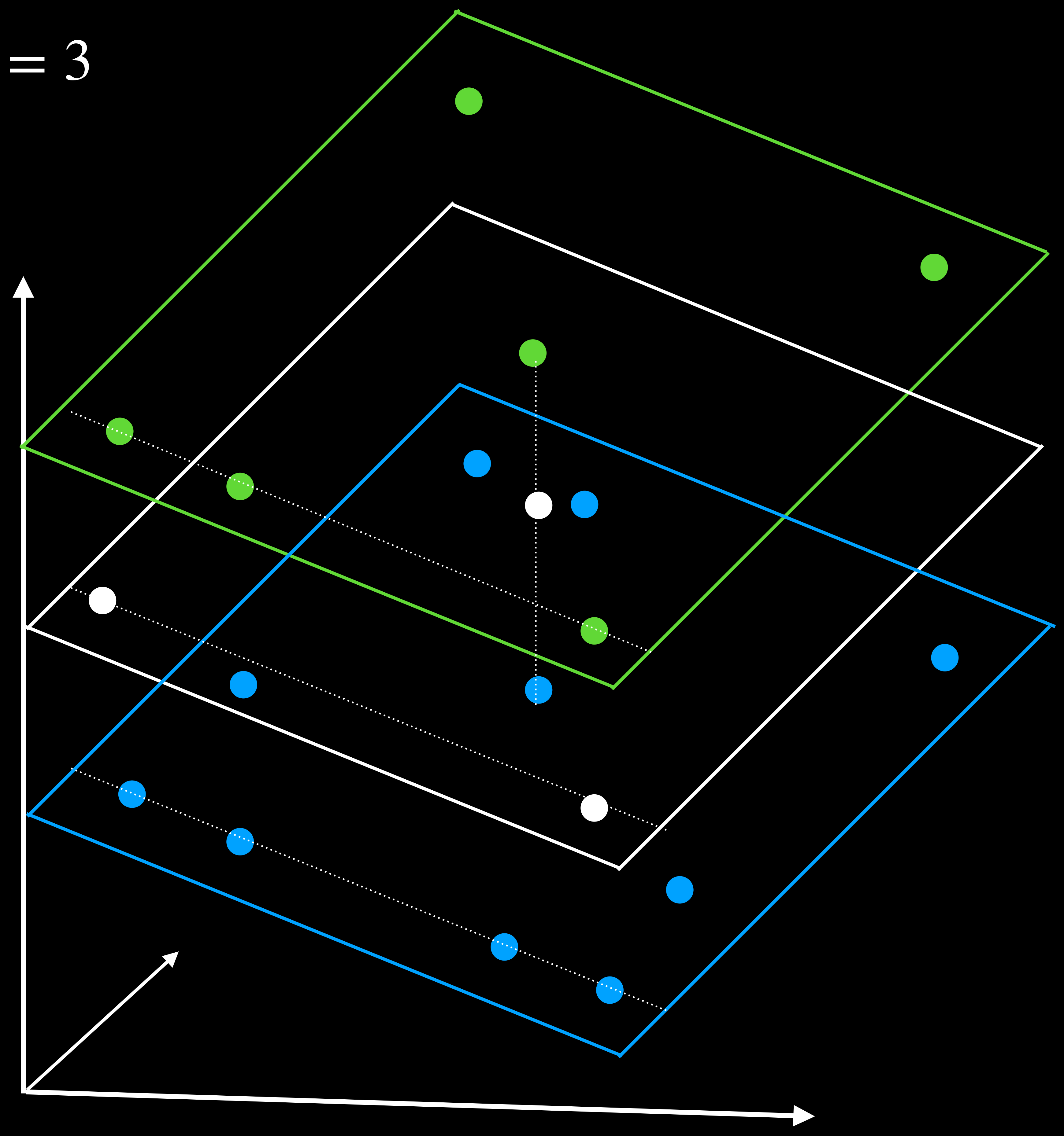
1
1
$n = 5$
0
−1
1



$n = 5$
1
1
0
−1
1


1
1
n = 5
−1
1

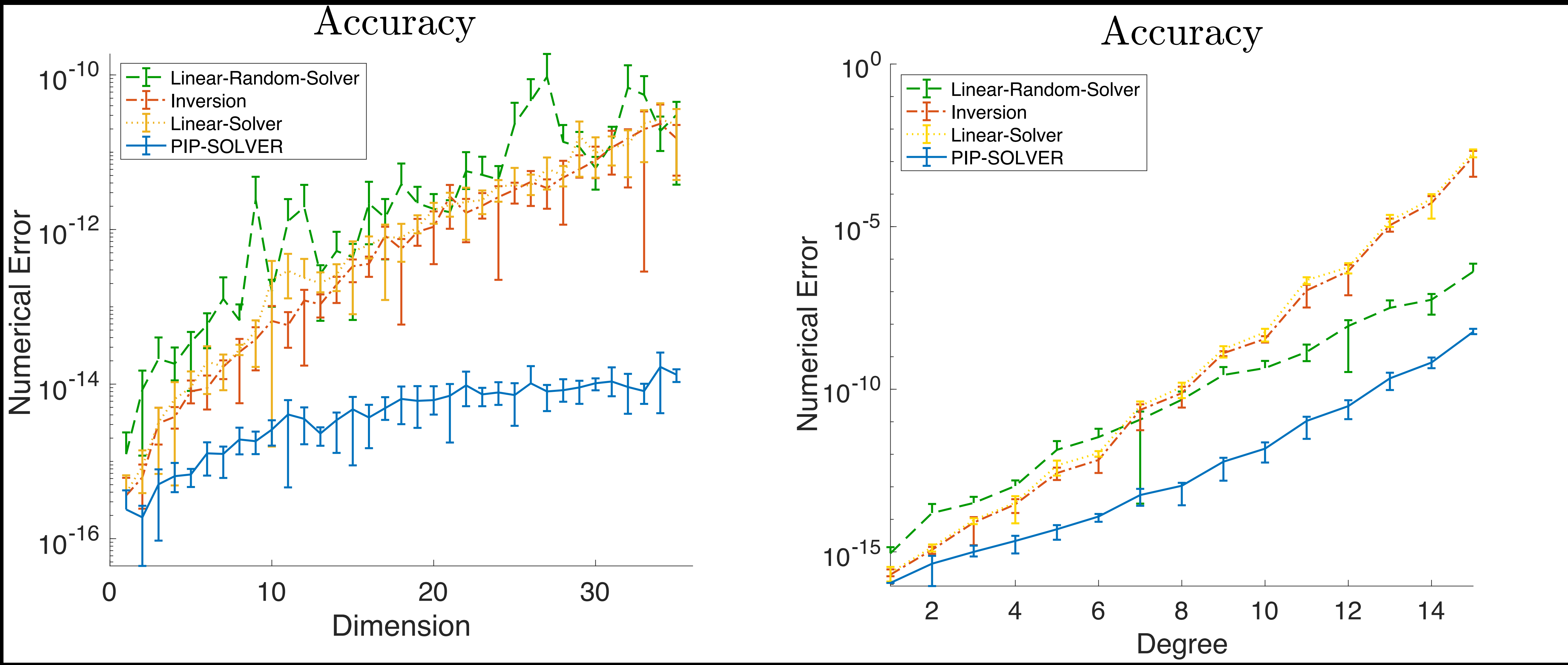
$m = 3, n = 3$



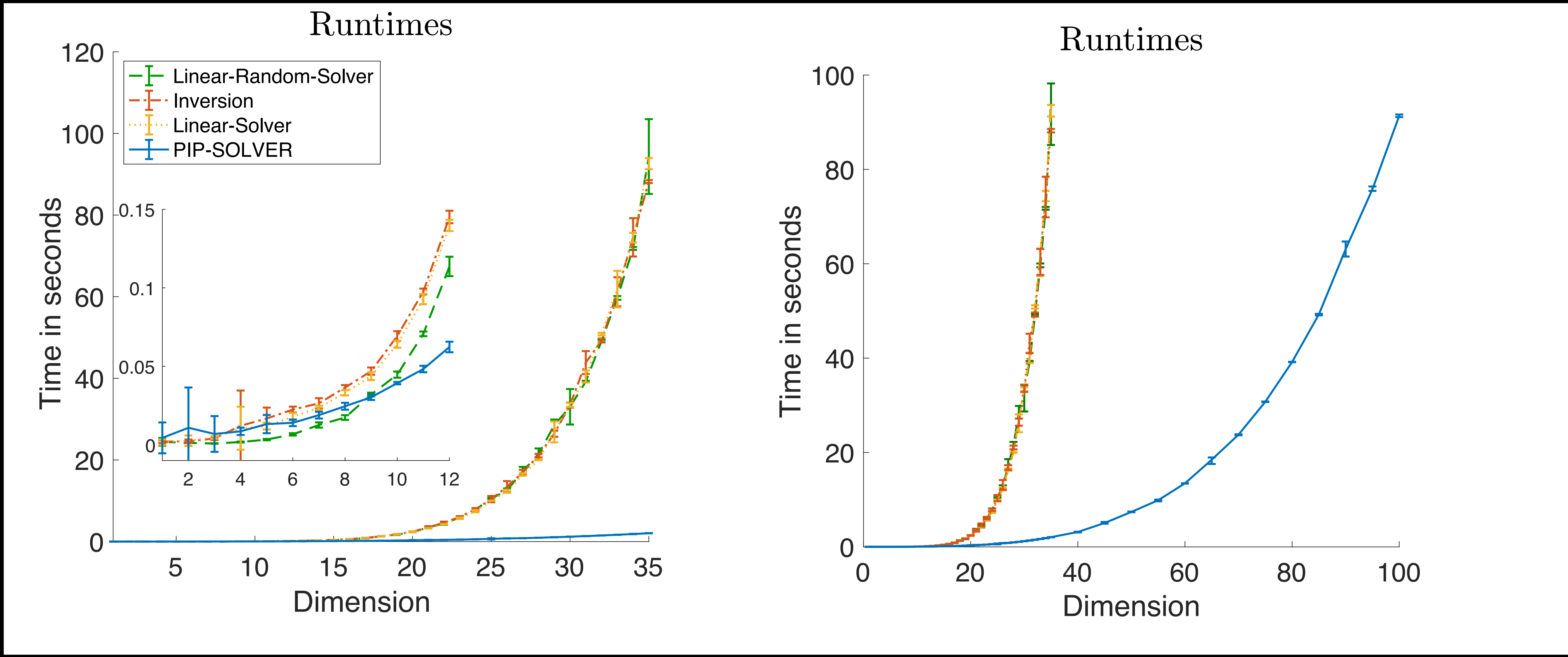
Numerical Error

Fixed Degree n=3

Fixed Dimension m=5



Fixed degree n=3



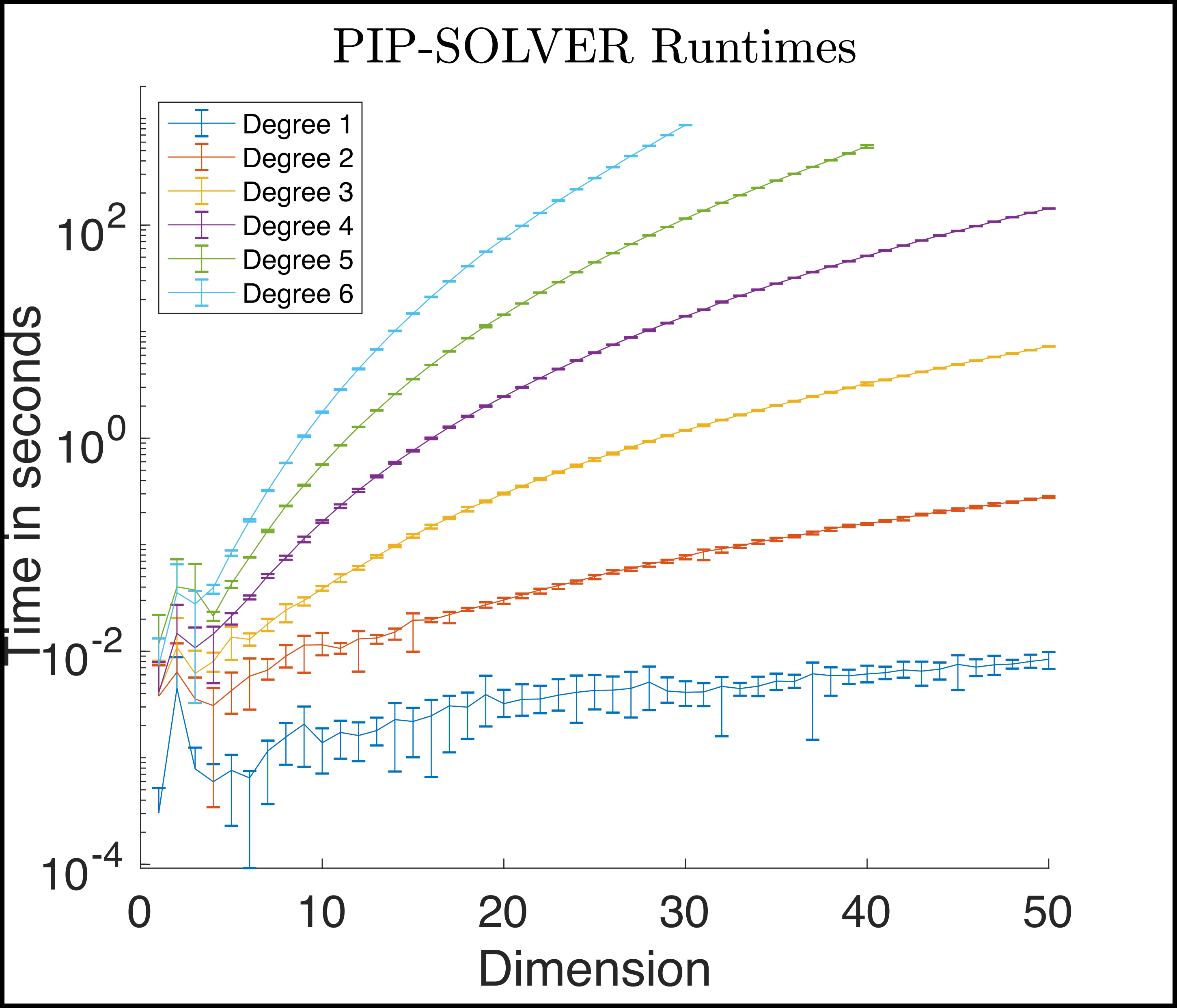
- Matlab Prototype vs Matlab-Inversion/Linear-Solver packages

Fitting the runtimes for $n = 3$ w.r.t $pN(m,n)^q$.

Algorithm	Intervals	Pre-factor p	Exponent q
Inversion	$m = 1,\ldots,35$	$p = 0.010737$	$q = 2.2982$
Linear Solver	$m = 1,\ldots,35$	$p = 0.007607$	$q = 2.2907$
PIP-SOLVER	$m = 1,\ldots,35$	$p = 0.007696$	$q = 1.2006$
PIP-SOLVER	$m = 1,\ldots,100$	$p = 0.003410$	$q = 1.2258$

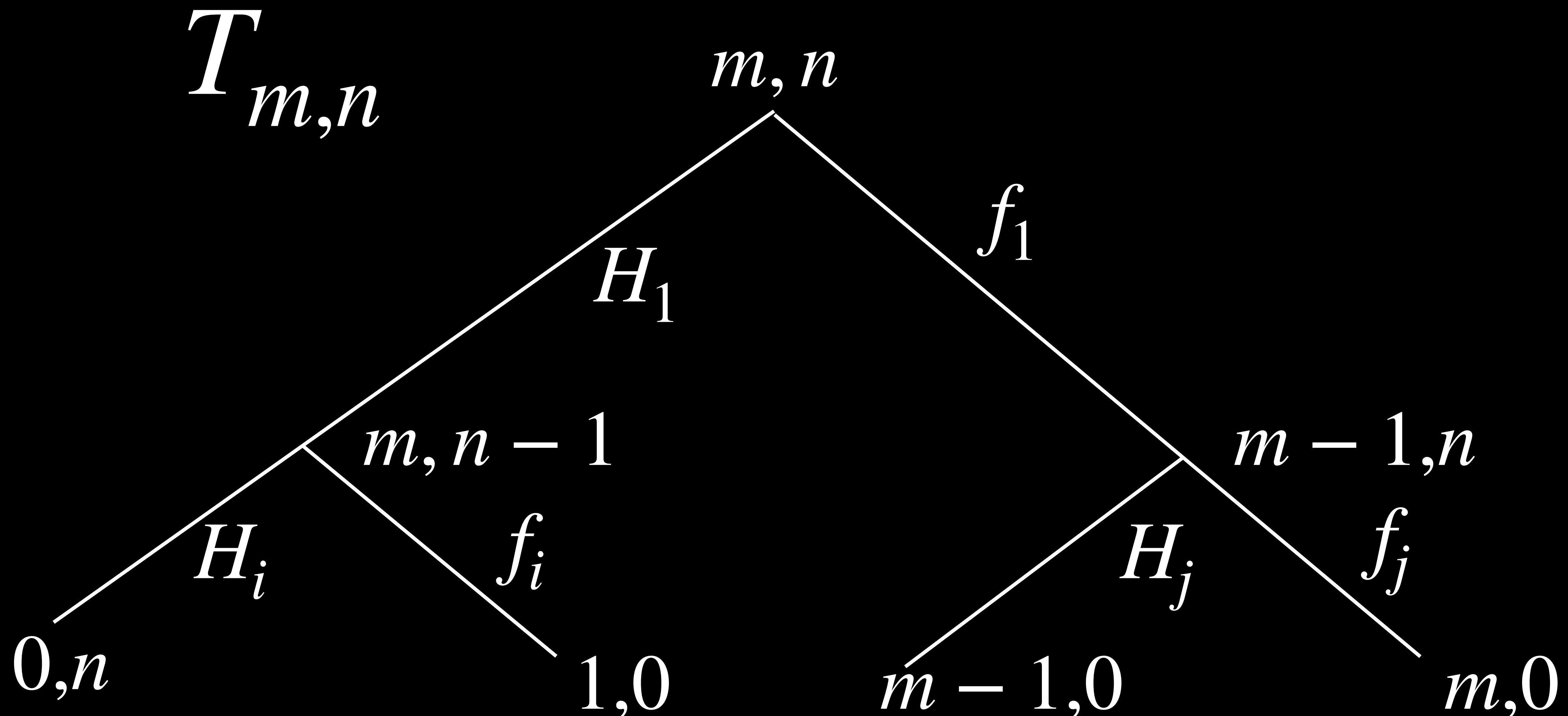
- **Linear storage amount allows to solve large instances, which could not be computed by the alternatives**

Runtimes for different degrees



Degree	Pre-factor p	Exponent q
1	0.0035219	1.0450
2	0.021732	1.1257
3	0.0031317	1.2096
4	0.0021351	1.1861
5	0.0017234	1.1478
6	0.0035746	1.1336

Fitting model $pN(m,n)^q$



- **Recursive Decomposition of the Interpolation Problem**
- **Formulating** ***barycentrical mD Lagrange Interpolation***
- **Runtime** $\mathcal{O}((m+n)N) = \mathcal{O}(m^n/n!)$

  $m = n \implies \mathcal{O}((m+n)N) = \mathcal{O}(\log(N)N)\,.$

Multivariate Lagrange Interpolation

Internship report

Vladimir Sivkin

Lomonossow-University Moskau

August 2019

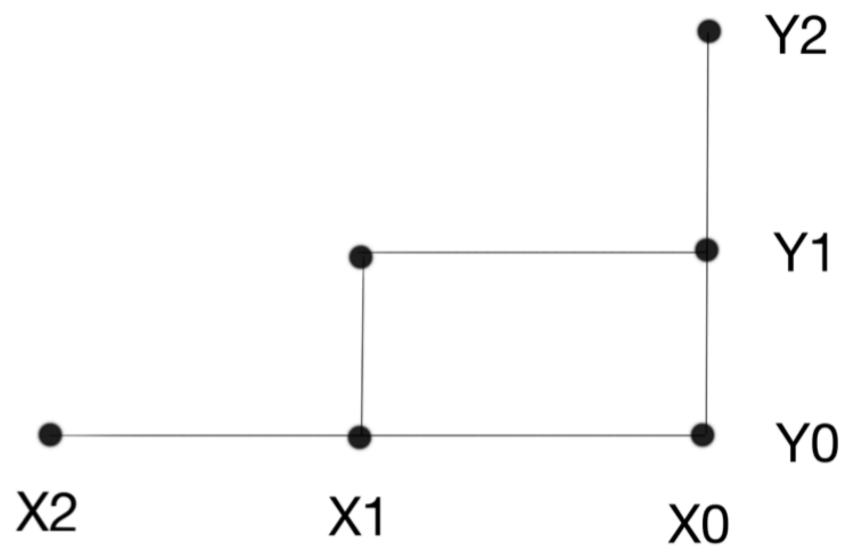
Case m=2, n=2.

To keep simple we start from $m = 2$. Denote by $l_{i,j}(x, y)$ a (desired) polynomial s.t.

$$l_{i,j}(x_p, y_q) = \delta_i^p \delta_j^q, \forall (x_p, y_q) \in P_{m,n}, \dim l = m, \deg l = n.$$

Let us show how to construct it.

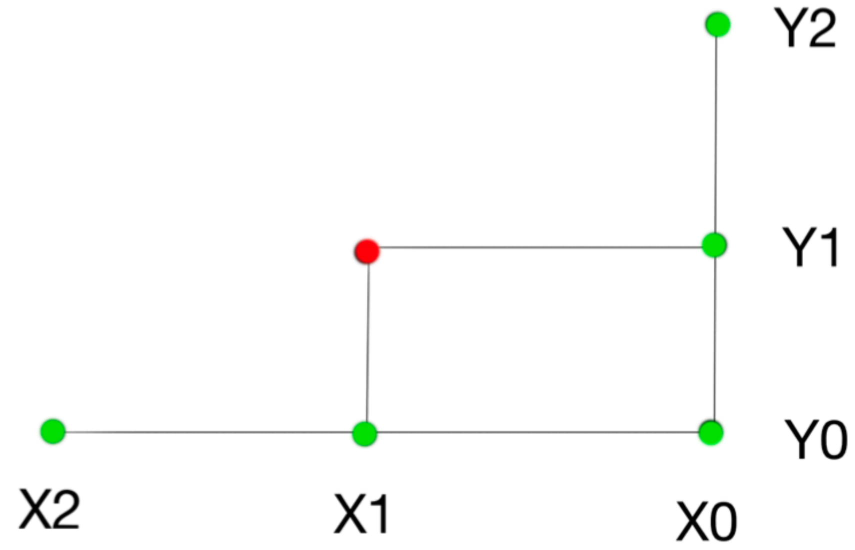
$$l_{2,0}(x, y) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$



Case m=2, n=2.

The next should be corrected by additional term.

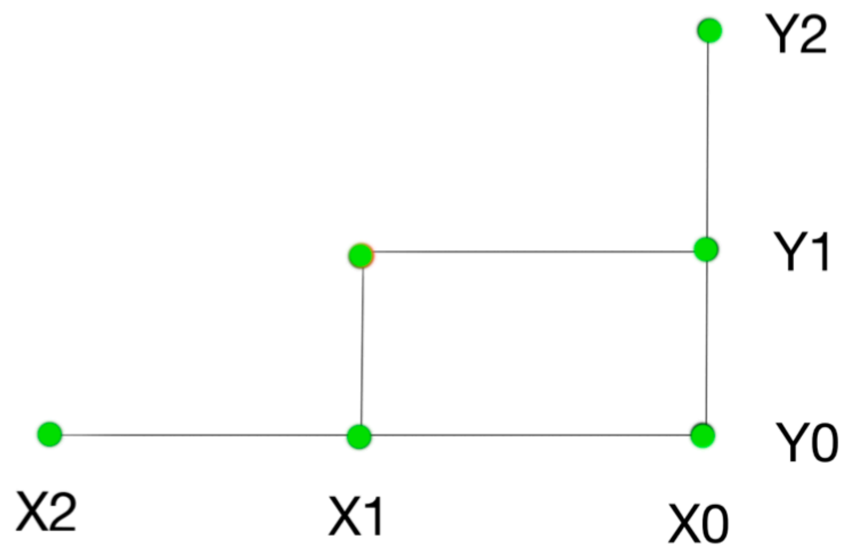
$$l_{1,0}(x,y) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + \dots$$



Case m=2, n=2.

The next is corrected by additional term.

$$l_{1,0}(x,y) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} - \frac{(x-x_0)(y-y_0)}{(x_1-x_0)(y_1-y_0)}$$

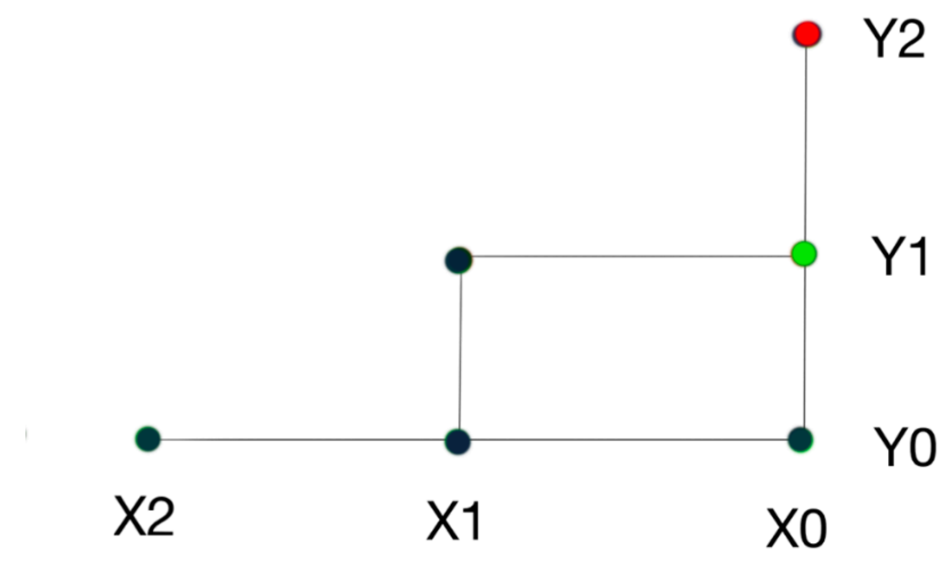


Case m=2, n=2.

This element should be corrected.

$$l_{0,1}(x,y) = \frac{(y-y_0)(x-x_1)}{(y_1-y_0)(x_0-x_1)} + \frac{(y-y_0)(y-y_1)}{(y_1-y_0)(y_1-y_2)} =$$

$$= \frac{y-y_0}{y_1-y_0}\left[\frac{x-x_1}{x_0-x_1} + \frac{y-y_1}{y_1-y_2}\right]$$



General md–Case

$$L_J(x) = \sum_{|I|=n, i_k \geq j_k, \forall k} r^m_{i_m - 1}(x_m) \ldots r^2_{i_2 - 1}(x_2) \frac{r^1_{i_1}(x_1)}{x_1 - p_{1,j_1}} w^m_{j_m, i_m} \ldots w^2_{j_2, i_2} w^1_{j_1, i_1 + 1},$$

i) $x = (x_1, \ldots, x_m)$

ii) $\{p_{i,j} : j = 0 \ldots n\}$ are the generating nodes

iii) $r^j_{i_j}(x_j) = \prod_{k=0}^{i_j}(x_j - p_{j,k})$,

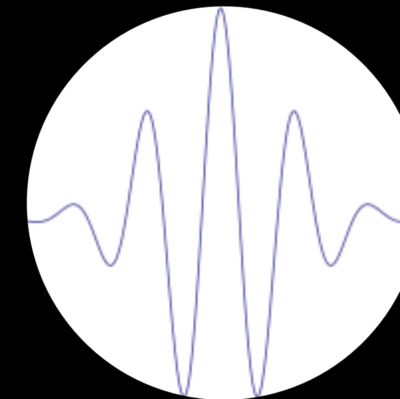
iv) $w^i_{j,k} = \prod_{l=0, l \neq j}^{k} \frac{1}{p_{i,j} - p_{i,l}}$.

The Curse of Dimensionality

mD Newton / Lagrange Interpolation

$$Q_f(x) = \sum_{\alpha, |\alpha| \leq n} c_\alpha N_\alpha(x) = \sum_{\alpha, |\alpha| \leq n} d_\alpha L_\alpha(x)$$

- Numerically accurate & fast $\mathcal{O}(N(m,n)^2)$
- Convergence to the ground truth $Q_{f,n} \xrightarrow[n\to\infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- In high Dimensions



Spline/Wavelet Interpolation & FFTs

$$Q_f(x) = \sum_{p \in G} c_p \gamma_p(x)$$

- Numerically accurate & fast $\mathcal{O}(M \log(M))$
- Convergence to ground truth $Q_{f,n} \xrightarrow[n\to\infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Feasible in low dimensions

The Curse of Dimensionality

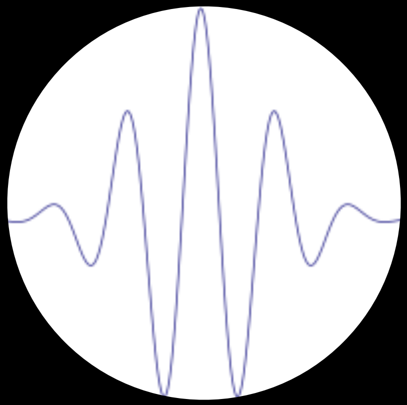
mD Newton / Lagrange Interpolation

$$Q_f(x) = \sum_{\alpha, |\alpha| \le n} c_\alpha N_\alpha(x) = \sum_{\alpha, |\alpha| \le n} d_\alpha L_\alpha(x)$$

$$\mathcal{O}(N(m,n)^2) \;/\; \mathcal{O}((m+n)N(m,n))$$

$$N(m,n) = \binom{m+n}{n} \in \mathcal{O}(m^n/n!)$$

$$m = n \Longrightarrow \mathcal{O}((m+n)N) = \mathcal{O}(\log(N)N)$$



Spline/Wavelet Interpolation & FFTs

$$Q_f(x) = \sum_{p \in G} c_p \gamma_p(x)$$

$$\mathcal{O}(M \log(M))$$

$$M = r^m \,,\; r \gg n$$

lp — degree

$$Q(x) = \sum_{\|\alpha\|_p \le n} c_\alpha x^\alpha , \quad x^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_m^{\alpha_m} ,$$

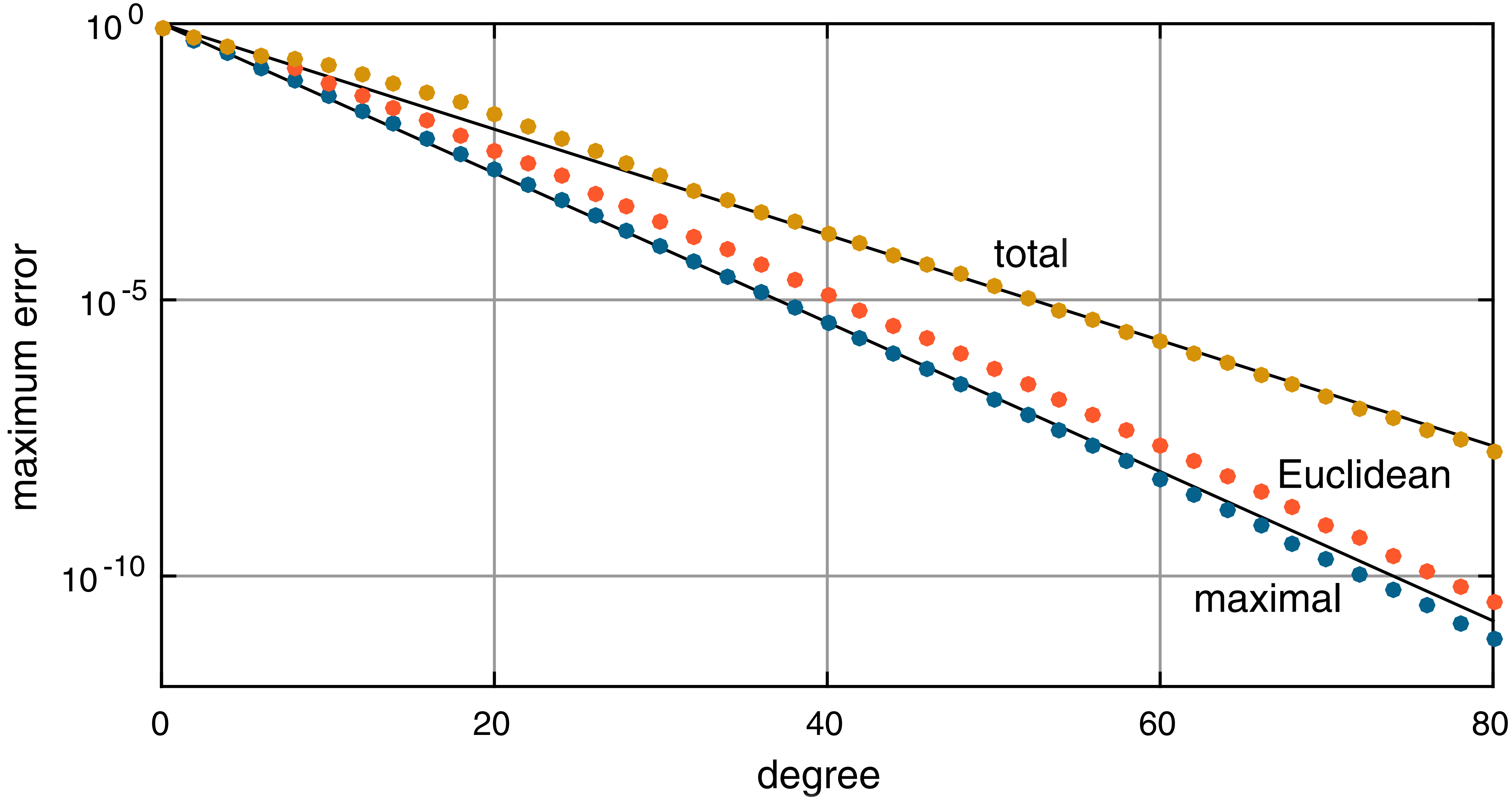
$$\|\alpha\|_p^p = \sum_{i=1}^{m} |\alpha_i|^p$$

Example:

$$x_1^2 \cdot x_2 , \quad n = 1 , p = 1$$

$$x_1^2 \cdot x_2 , \quad n = 2 , p = 2$$

$$x_1^n \cdot x_2^n \cdots x_m^n , \quad n \in \mathbb{N} , p = \infty$$



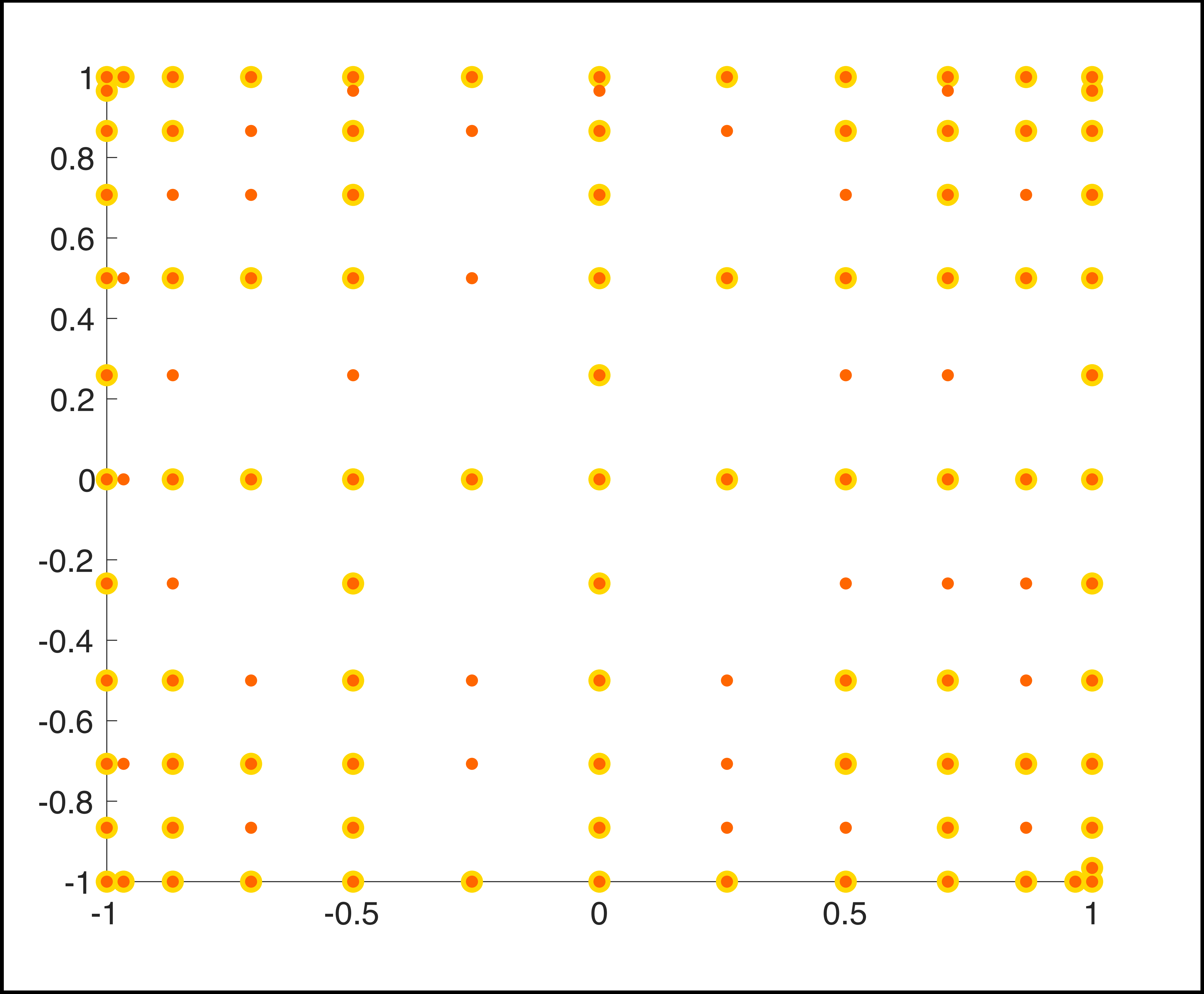
Theorem 4.2. *If f satisfies Assumption* A, *then*

$$\inf_{d(p)\le n} \|f-p\|_{[-1,1]^s} = \begin{cases} O_\varepsilon(\rho^{-n/\sqrt{s}}), & \text{if } d = d_T, \\ O_\varepsilon(\rho^{-n}), & \text{if } d = d_E, \\ O_\varepsilon(\rho^{-n}), & \text{if } d = d_{\max}, \end{cases}$$

where $\rho = h + \sqrt{1+h^2}$. *(defines the Newton Ellipse)*

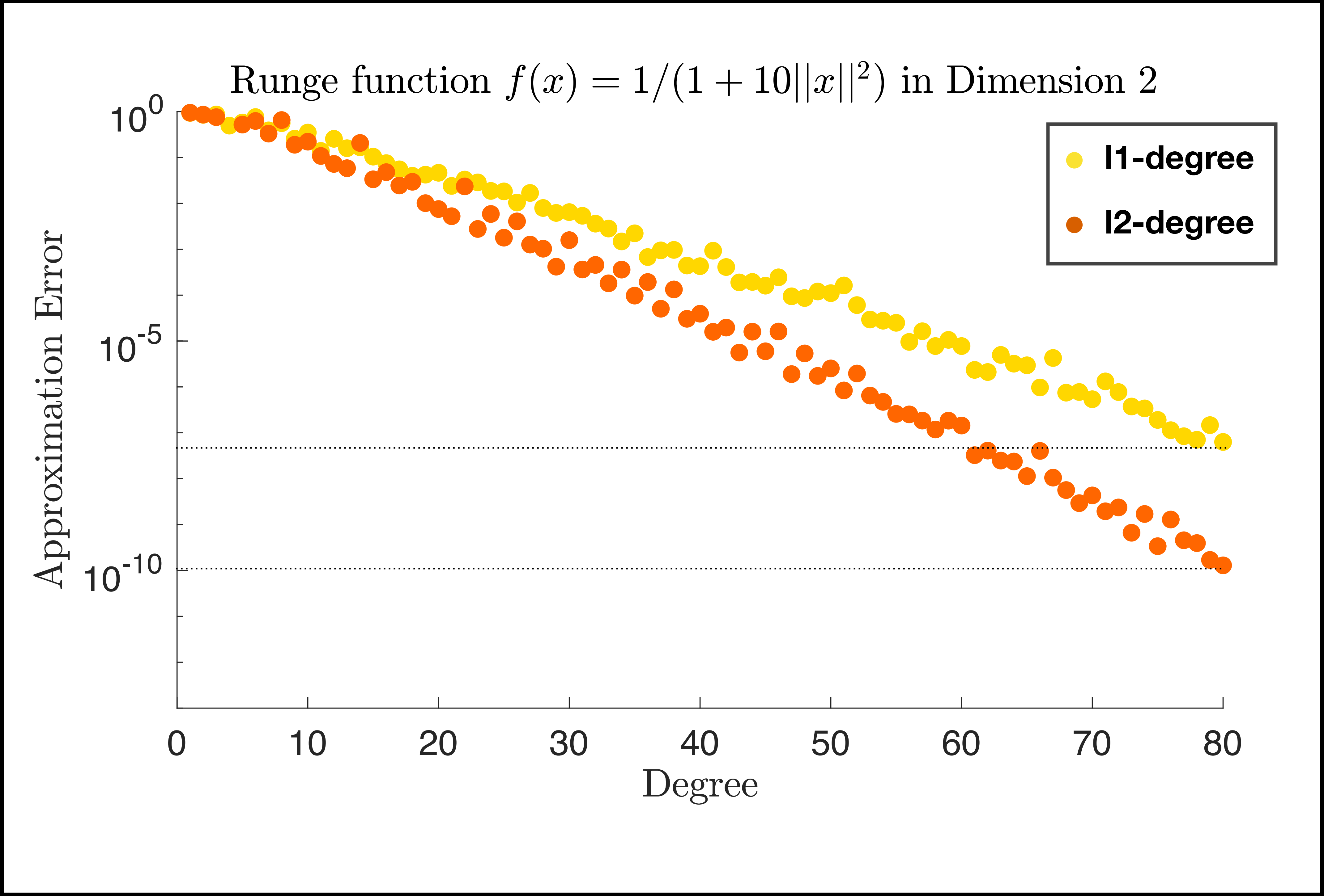
LLoyd N. Trefethen.: Multivariate polynomial approximation in the hypercube. Proceedings of the American Mathematical Society 145 (4837-4844), 11 (2017)

Leja ordered Chebyshev nodes in 2D w.r.t. l1/l2-degree for n=12



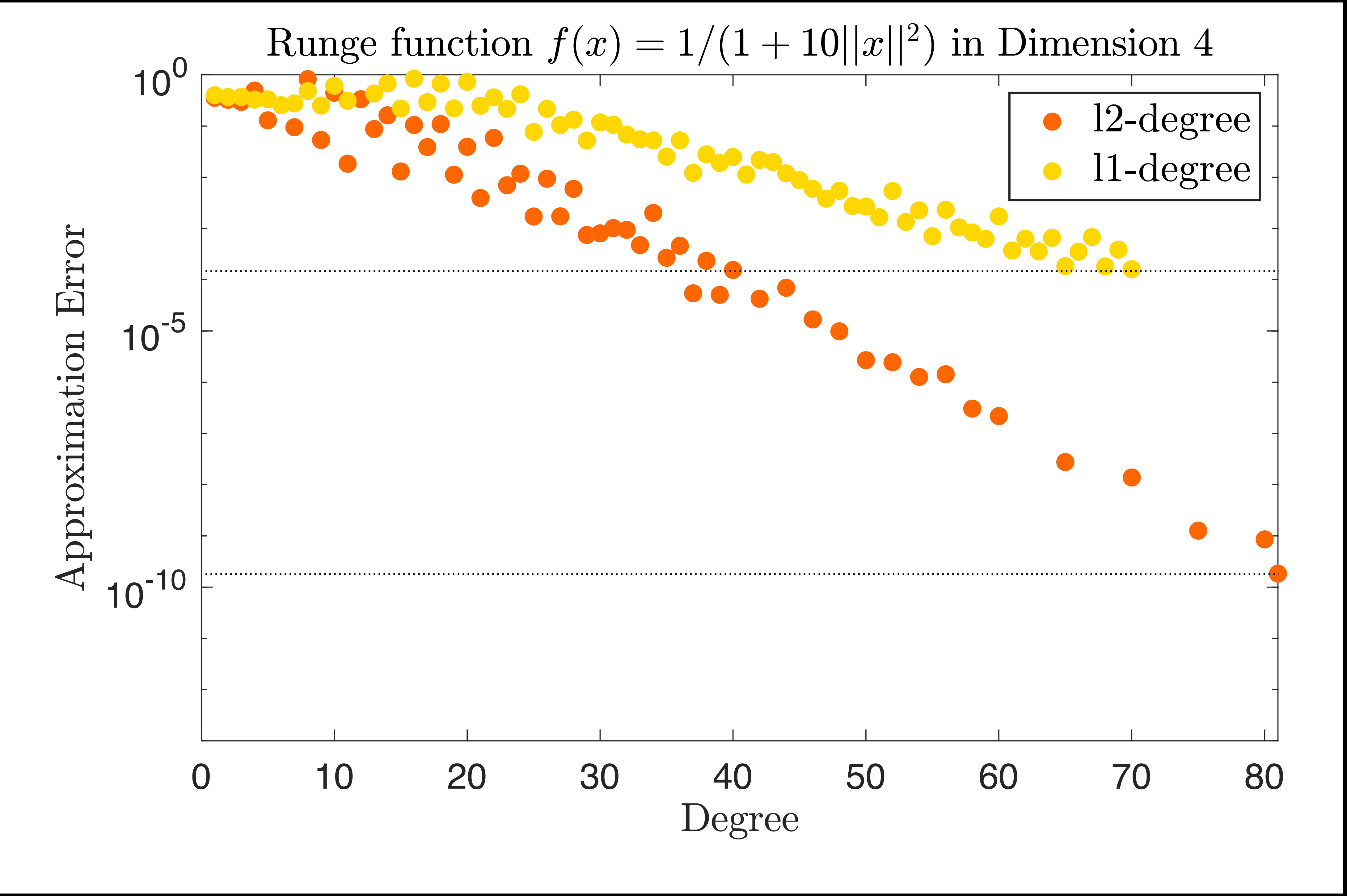
Jannik Michelfeit

mD Newton Interpolation



$$\frac{N_\infty(2,80)}{N_2(2,80)} \approx 1.5$$

mD Newton Interpolation



$$\frac{N_\infty(4,80)}{N_2(4,80)} \approx 3 \qquad \frac{N_\infty(10,n)}{N_2(10,n)} \approx 402$$

Summary

Unisolvent Nodes

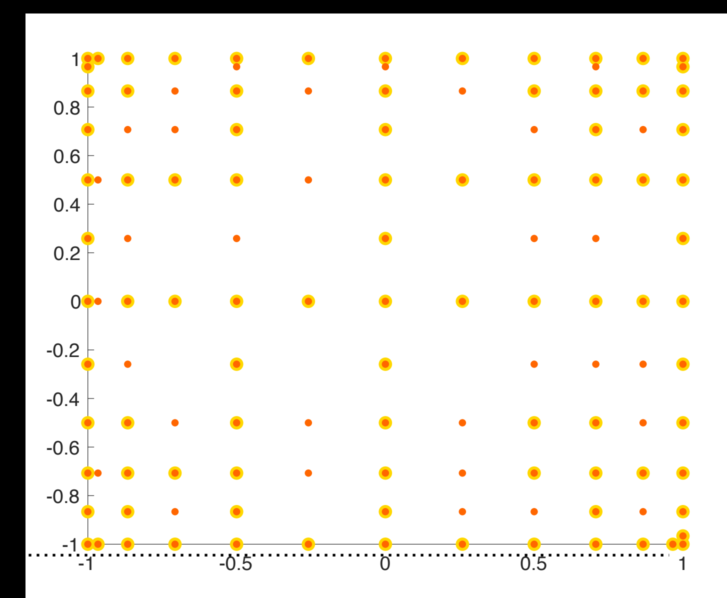
How to choose P such that $V_{m,n,P}$ becomes (numerically) invertible ?

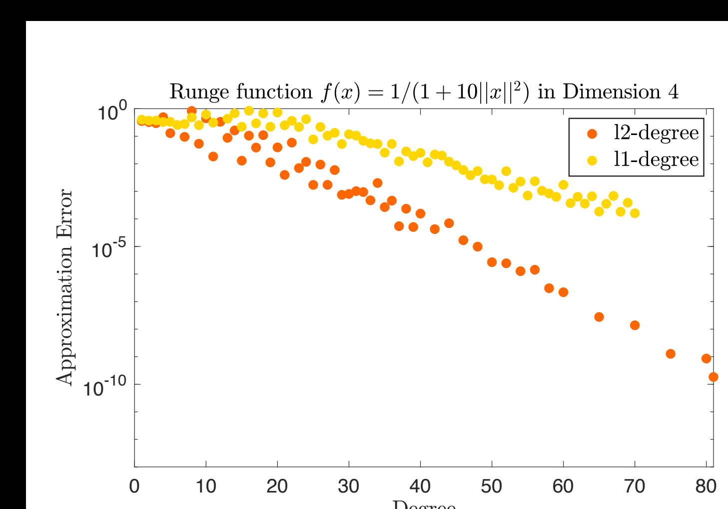
mD Newton / Lagrange Interpolation

$$Q_f(x) = \sum_{\alpha,|\alpha|\leq n} c_\alpha N_\alpha(x) = \sum_{\alpha,|\alpha|\leq n} d_\alpha L_\alpha(x)$$

$$Q_{f,n} \xrightarrow[n\to\infty]{} f \quad \forall f \in H^k(\Omega)\,,\ k > m/2$$

Sparse Chebyshev—grid





Adaptive Optics & Phase Reconstruction

Leslie Greengard

Charles L. Epstein

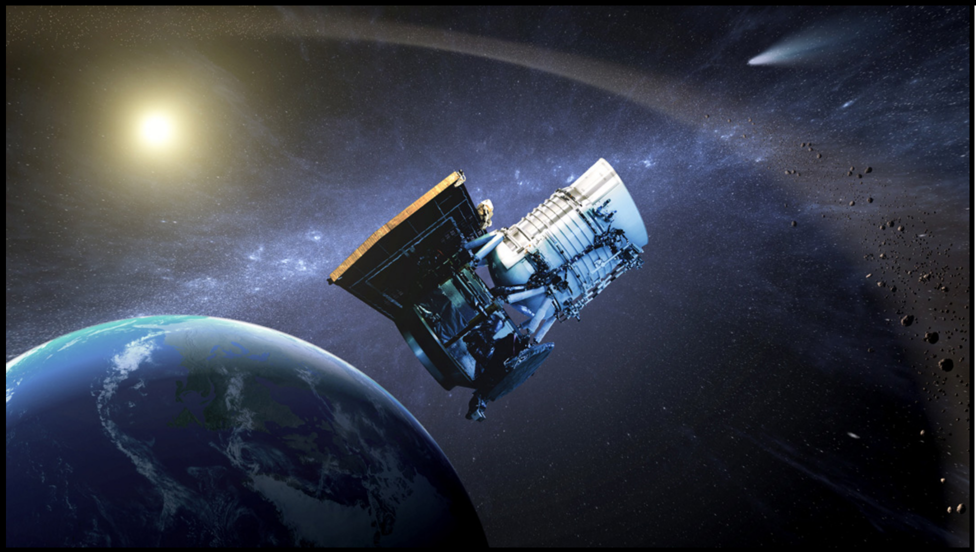
Newtons Telescope

Michael Bussmann

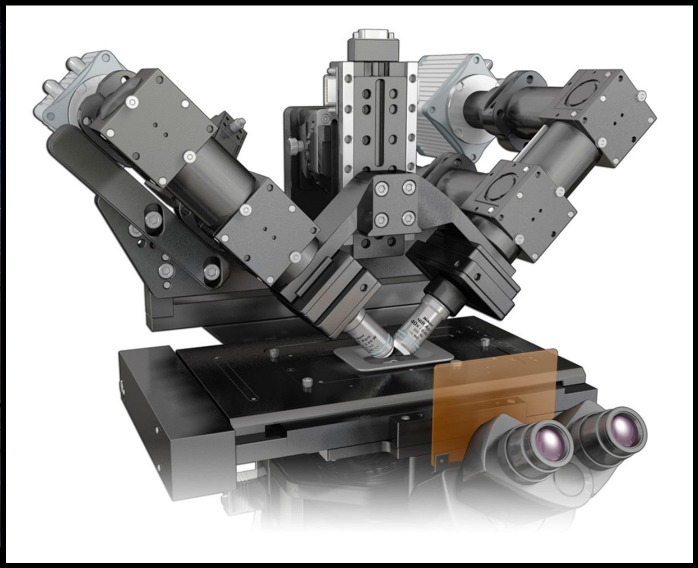
Gene Myers

A deeper view into Space, Biology and the Universe at all.

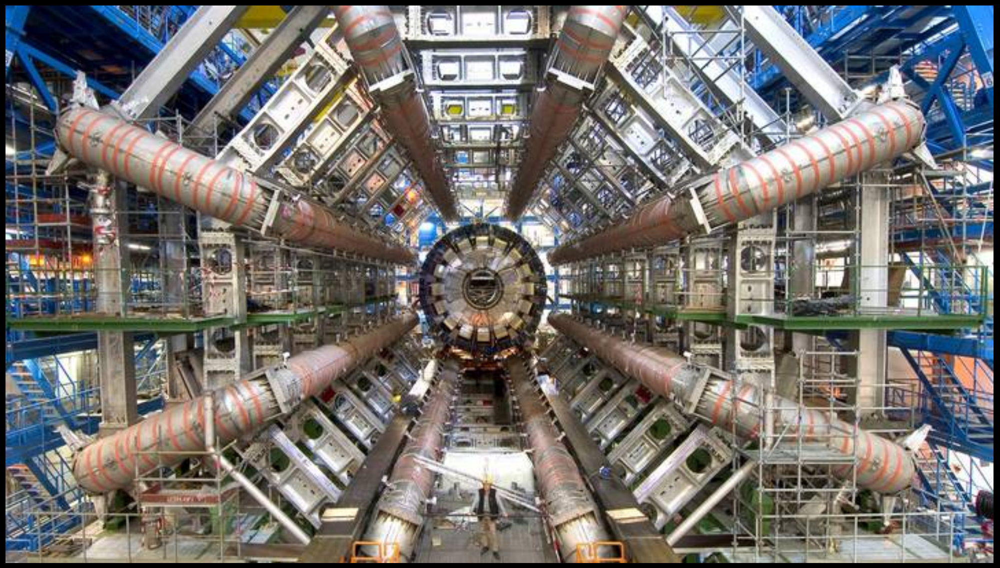
Sharper Selfies



Keppler Telescope



Light Microscopy



CERN Detector

Applications & Further Developments

Fourier Interpolation
&
Faster FFT's

Leslie Greengard

Manas Rachh

Multivariate Polynomial Regression
&
Numerical Integration

Christian L. Mueller

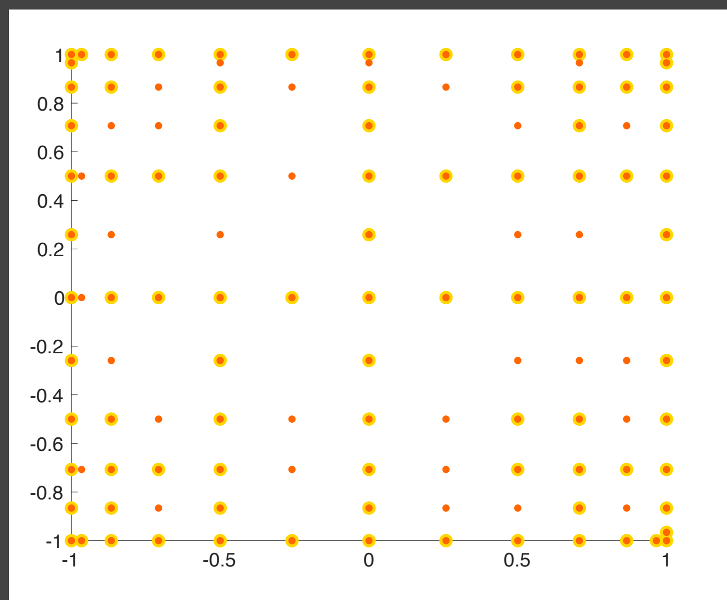
Spectral Particle Methods
&
FFT's for Strong Oscillating Signals

Ivo F. Sbalzarini

Michael Bussmann

Numerical Integration (Outlook)

Sparse Chebyshev—grid



mD—Lagrange Polynomials

$$L_\alpha(p) = \delta_{p_\alpha,p}$$

$$f(x) \approx Q_f(x) = \sum_{\|\alpha\|_p \le n} f(p_\alpha) L_\alpha(x)$$

$$\int_\Omega f(x)dx \approx \sum_{\|\alpha\|_p \le n} f(p_\alpha) \int_\Omega L_\alpha(x)dx = \sum_{\|\alpha\|_p \le n} f(p_\alpha) l_\alpha$$

Runtime $\mathcal{O}(N_p(m,n))\,, \quad N_p(m,n) \in \mathcal{O}(m^n/n!)\,, p = 1$

Acknowledgements

FLATIRON INSTITUTE
Center for Computational Mathematics

Leslie Greengard

Christian L. Mueller

Manas Rachh

TECHNISCHE UNIVERSITÄT DRESDEN

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

CBG
Max Planck Institute of Molecular Cell Biology and Genetics

center for systems biology dresden

Ivo F. Sbalzarini

Michael Bussmann

Jannik Michelfeit

Karl B. Hoffmann

Bevan L. Cheeseman

Vladimir Sivkin

Thank You !

Multivariate Newton Interpolation on arXiv

Email: hecht@mpi-cbg.de