The Fast Newton Transform: Multivariate Interpolation in Downward Closed Spaces

CASUS CENTER FOR ADVANCED SYSTEMS UNDERSTANDING
www.casus.science

Phil-Alexander Hofmann and Michael Hecht

Mathematical Foundations of Complex Systems Science, CASUS / HZDR
Mathematical Institute, University Wrocław

SIGMA, CIRM, Marseille, 2024

HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF
TECHNISCHE UNIVERSITÄT DRESDEN
CBG Max Planck Institute of Molecular Cell Biology and Genetics
UFZ HELMHOLTZ Centre for Environmental Research
Uniwersytet Wrocławski
SPONSORED BY THE
Federal Ministry of Education and Research
STAATSMINISTERIUM FÜR WISSENSCHAFT KULTUR UND TOURISMUS
Freistaat SACHSEN

CASUS - Center for Advanced Systems Understanding

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING
www.casus.science

HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

TECHNISCHE UNIVERSITÄT DRESDEN

CBG Max Planck Institute of Molecular Cell Biology and Genetics

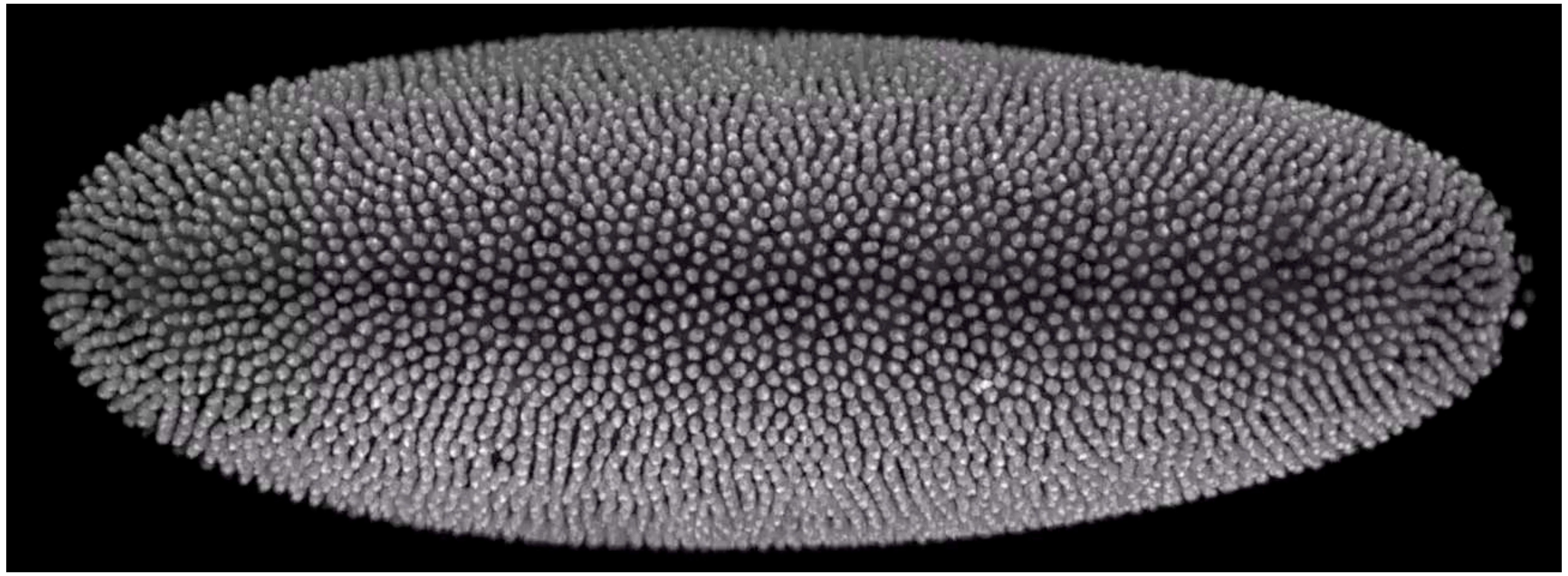
UFZ HELMHOLTZ Centre for Environmental Research

Uniwersytet Wrocławski

SPONSORED BY THE
Federal Ministry of Education and Research
STAATSMINISTERIUM FÜR WISSENSCHAFT KULTUR UND TOURISMUS
Freistaat SACHSEN

Complex Systems Across Disciplines

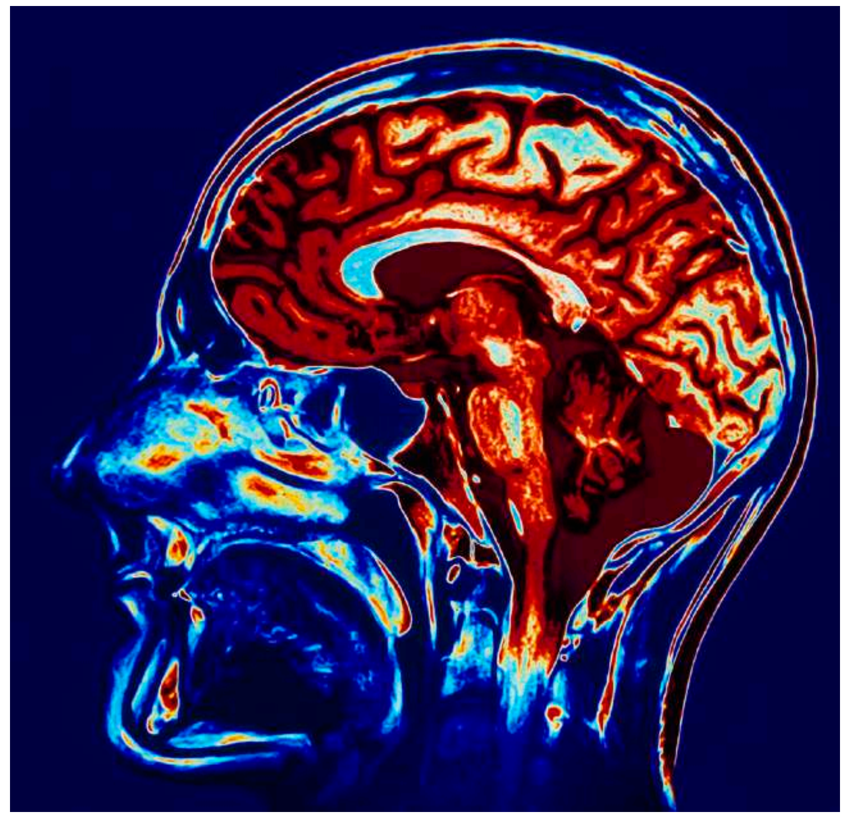
bio-physics and bio-chemistry



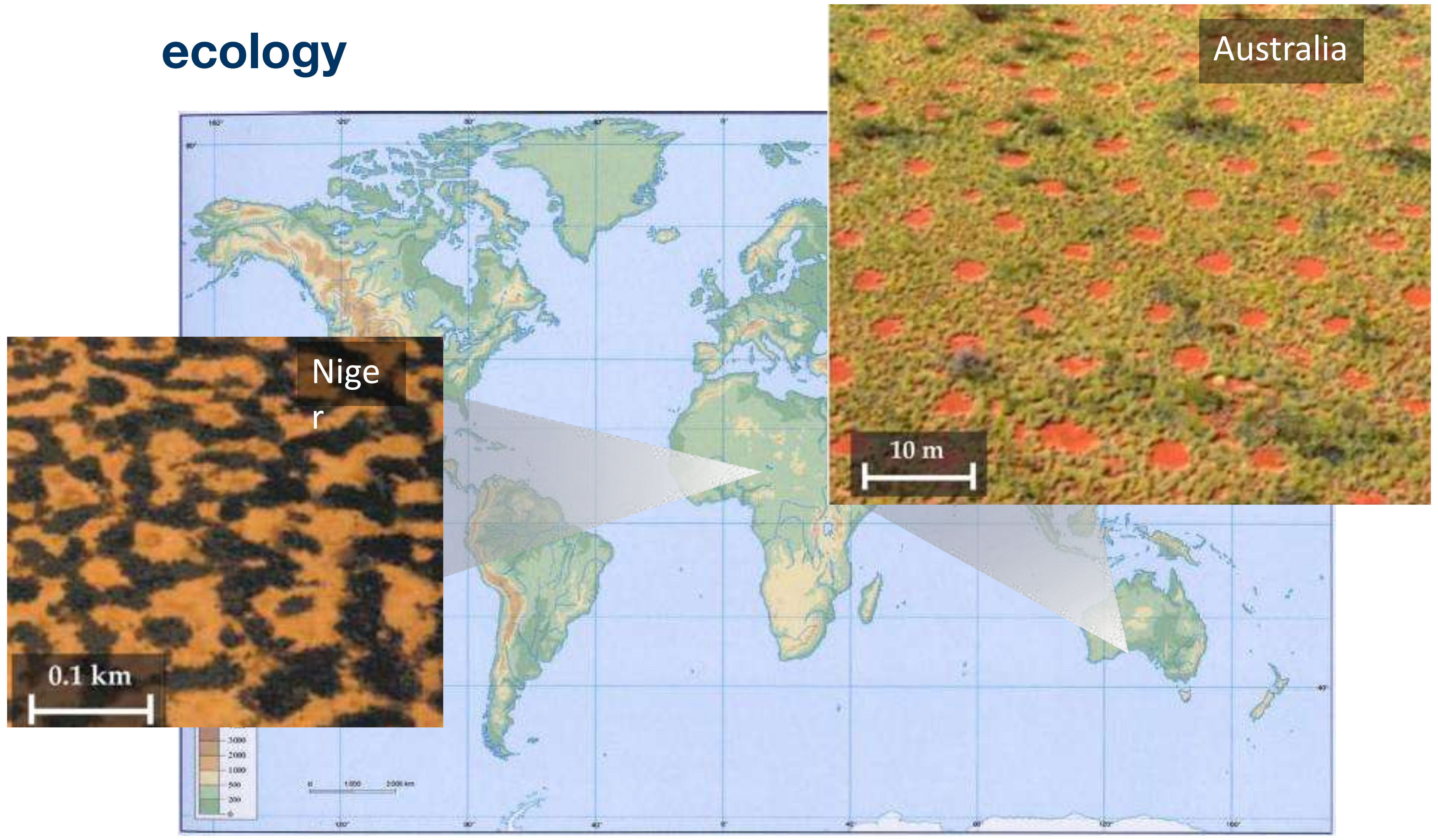
Drosophila Embryo development by Loic Royer

droplet simulations / fluid dynamics

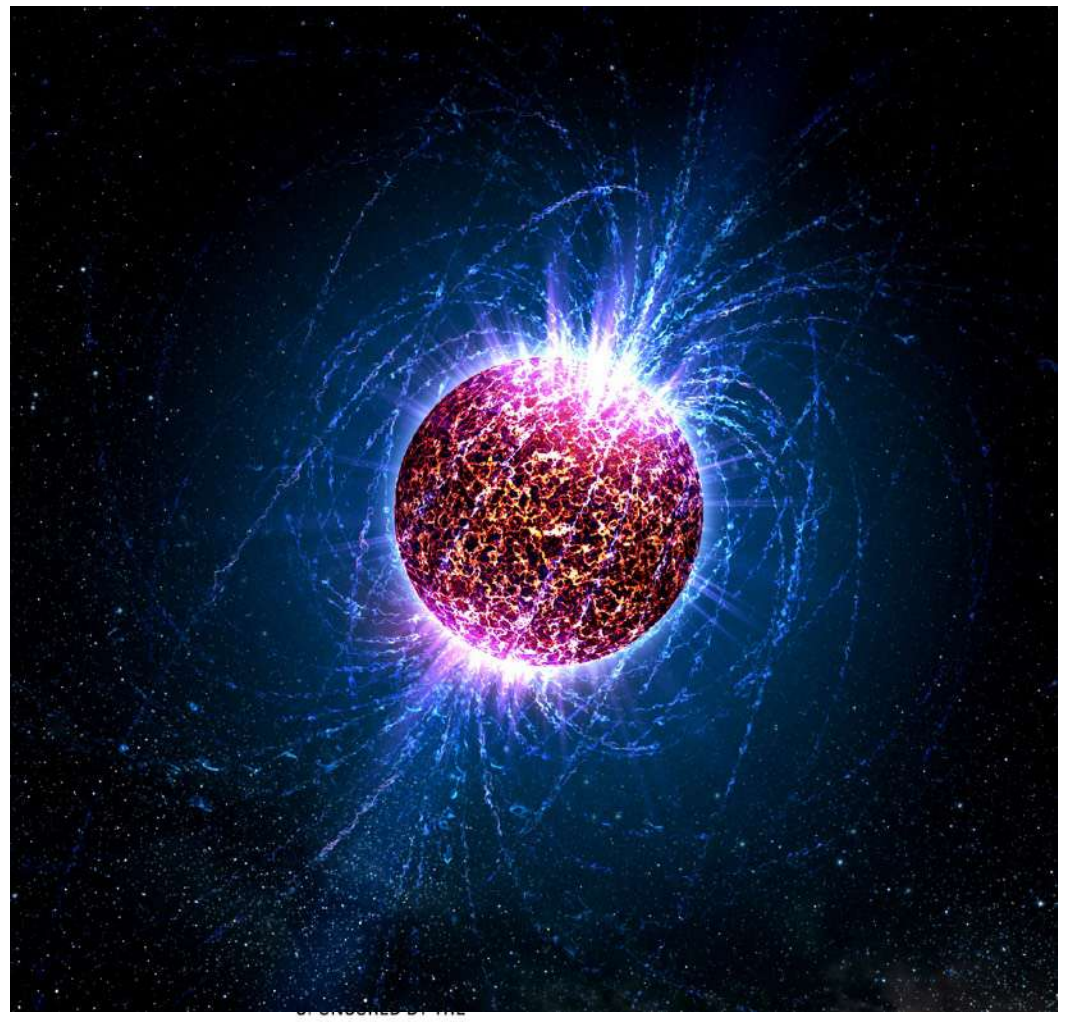
neuro science



ecology



astro and quantum physics



HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

TECHNISCHE UNIVERSITÄT DRESDEN

CBG Max Planck Institute of Molecular Cell Biology and Genetics

UFZ HELMHOLTZ Centre for Environmental Research

Uniwersytet Wrocławski

Federal Ministry of Education and Research

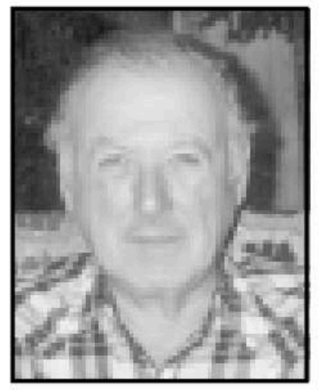
STAATSMINISTERIUM FÜR WISSENSCHAFT KULTUR UND TOURISMUS

Freistaat SACHSEN

from *SIAM News*, Volume 33, Number 4 (2000)

The Best of the 20th Century: Editors Name Top 10 Algorithms

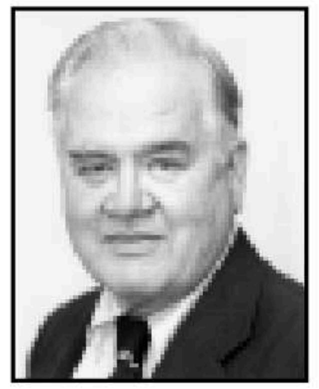
By Barry A. Cipra



James Cooley

1965: James Cooley of the IBM T.J. Watson Research Center and John Tukey of Princeton University and AT&T Bell Laboratories unveil the **fast Fourier transform**.

Easily the most far-reaching algo-rithm in applied mathematics, the FFT revolutionized signal processing. The underlying idea goes back to Gauss (who needed to calculate orbits of asteroids), but it was the Cooley–Tukey paper that made it clear how easily Fourier transforms can be computed. Like Quicksort, the FFT relies on a divide-and-conquer strategy to reduce an ostensibly $O(N^2)$ chore to an $O(N \log N)$ frolic. But unlike Quick- sort, the implementation is (at first sight) nonintuitive and less than straightforward. This in itself gave computer science an impetus to investigate the inherent complexity of computational problems and algorithms.



John Tukey

1987: Leslie Greengard and Vladimir Rokhlin of Yale University invent the **fast multipole algorithm**.

This algorithm overcomes one of the biggest headaches of N-body simulations: the fact that accurate calculations of the motions of N particles interacting via gravitational or electrostatic forces (think stars in a galaxy, or atoms in a protein) would seem to require $O(N^2)$ computations—one for each pair of particles. The fast multipole algorithm gets by with $O(N)$ computations. It does so by using multipole expansions (net charge or mass, dipole moment, quadrupole, and so forth) to approximate the effects of a distant group of particles on a local group. A hierarchical decomposition of space is used to define ever-larger groups as distances increase. One of the distinct advantages of the fast multipole algorithm is that it comes equipped with rigorous error estimates, a feature that many methods lack.

Dongarra, J. and F. Sullivan (2000). Top ten algorithms of the century. Computing in Science and Engineering 2(1), 22–23.

What new insights and algorithms will the 21st century bring? The complete answer obviously won't be known for another hundred years. One thing seems certain, however. As Sullivan writes in the introduction to the top-10 list, "The new century is not going to be very restful for us, but it is not going to be dull either!"

How to compute Multivariate Function Expansions that closely approximate the ground truth function and its derivatives FAST ?

Geometric approximation rates of Fourier expansions for analytic periodic functions (FFT)

$$\theta(z) \approx \theta_n(z) = \sum_{\|\alpha\|_\infty \le n} c_\alpha e^{2\pi i \alpha \cdot z}, \; \| \theta - \theta_n \|_{C^0(\Omega)} = \mathcal{O}(r^{-n}), r > 1$$

Gauss, C. F. (1886). Theoria interpolationis methodo nova tractata Werke band 3, 265–327. *Göttingen: Königliche Gesellschaft der Wissenschaften*.
Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*

Geometric approximation rates of the Coulomb / Gravitation field expansion in (FMM)s

$$\phi(z) \approx \phi_n(z) = Q\log(z) + \sum_{k=1}^{n} \frac{a_k}{z^k}, \; \| \phi - \phi_n \|_{C^0(\Omega)} = \mathcal{O}(|z|^{-n}), |z| > 1\,.$$

Greengard, L., & Rokhlin, V. (1987). A fast algorithm for particle simulations. *Journal of computational physics*

How to compute Multivariate Function Expansions that closely approximate the ground truth function and its derivatives FAST ?

Geometric approximation rates of Fourier expansions for analytic periodic functions (FFT)

$\theta(z) \approx \theta$

Gauss, C. F. (1886). ... *...senschaften*.
Cooley, J. W., & Tuk...

Geometric app...

ϕ

Greengard, L., & Ro... *...*

Limitations

Periodicity of FFTs !
Quadratic decay of the Coulomb field in FMMs !

NO RESISTANCE to the Curse of Dimensionality !

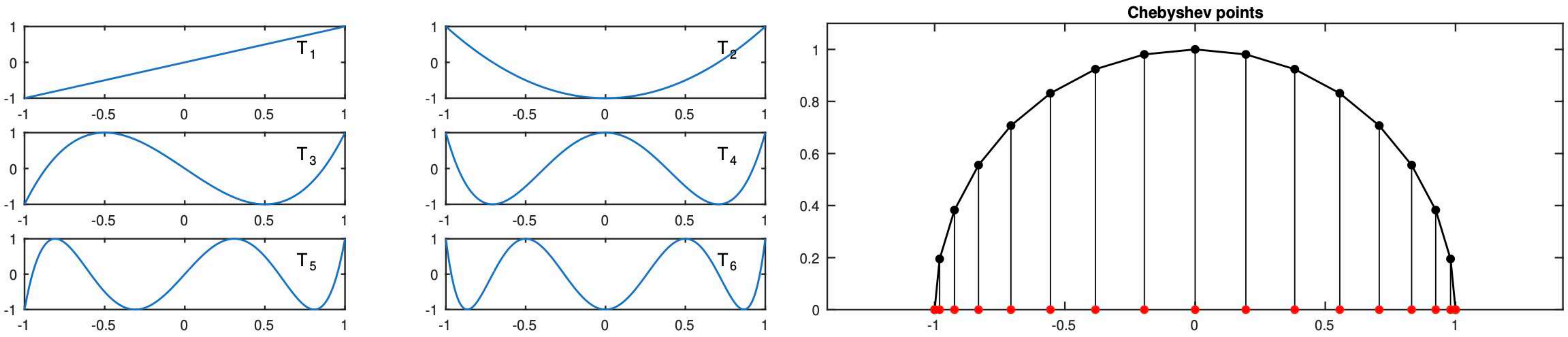
Polynomials are <u>not</u> feasible for computations ?!

"By the change of variables $x = \cos(\theta)$, one can show that interpolation by polynomials in Chebyshev points is equivalent to interpolation of periodic functions by series of sines and cosines in equispaced points. The latter is the subject of discrete Fourier analysis, and one cannot help noting that whereas there is widespread suspicion that it is not safe to compute with polynomials, nobody worries about the Fast Fourier Transform! In the end, that this may be the biggest difference between Fourier and polynomial interpolants, the difference in their reputations." (Trefethen, 2019).

Nick Trefethen

Harvard John A. Paulson School of Engineering and Applied Sciences

Mathematical Institute, University of Oxford



(a) T_k for degree $n = 1, \ldots 6$

(b) Chebyshev-Lobatto points of order 17.

Figure 1: Chebyshev polynomials of first kind and Chebyshev-Lobatto points, from Trefethen (2019).

Trefethen, L. N. (2019). Approximation theory and approximation practice, SIAM.

Polynomials are <u>not</u> feasible for computations ?!

Geometric approximation of analytic functions in 1D

<u>**Theorem:**</u> Let $f : [-1,1] \to \mathbb{R}$, p_n^* its best polynomial approximation of degree $n \in \mathbb{N}$. Then

$$\|f - p_n{}^*\|_\infty = \mathcal{O}(\rho^{-n}), \quad \rho > 1$$

if and only if, f is the restriction of a function $F : E_\rho \subset \mathbb{C} \to \mathbb{C}$ holomorphic in the open *Bernstein ellipse* E_ρ. Consequently,

$$\|f - Q_n\|_\infty = \mathcal{O}(\rho^{-n+1}), \quad \rho > 1$$

applies for the resulting interpolant Q_n in *Chebyshev-Lobatto-nodes*. **Moreover**, even the k-th order deriavitves are approximated as

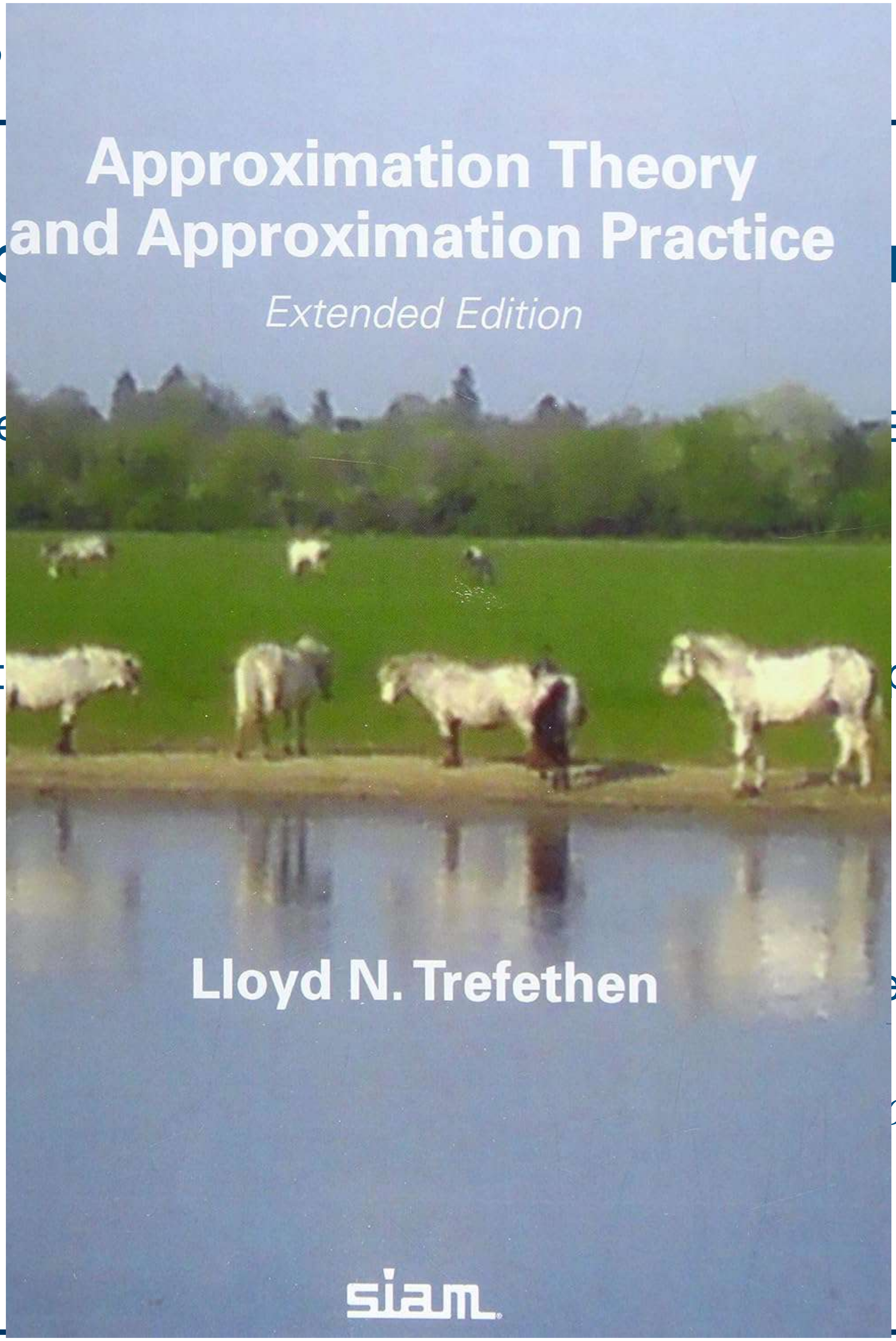
$$\|f^{(k)} - Q_n^{(k)}\|_\infty = \mathcal{O}_\varepsilon(\rho^{-n+1}) = \mathcal{O}((\rho-\varepsilon)^{-n+1}), \quad \rho > \varepsilon > 0.$$

Trefethen, L. N. (2019). Approximation theory and approximation practice, SIAM.

Polynomials are not feas

Geometric ap

nctions in 1D



Theorem: Let $f : [-1,1] \to \mathbb{R}$, p_n^* its be ∈ ℕ. Then

if and only if, f is the restriction of a funct pen *Bernstein ellipse* E_ρ.
Consequently,

1

applies for the resulting interpolant Q_n in en the k-th order deriavitves are approximated as

$$\rho - \varepsilon)^{-n+1}), \quad \rho > \varepsilon > 0.$$

Trefethen, L. N. (2019). Approximation theory and approximation practice, SIAM.

The framework in mD

Hypercube

$$\square_m = [-1,1]^m\,,\, m = \dim$$

Downward closed polynomial space

$\Pi_A = \mathrm{span}\{x^\alpha : \alpha \in A\}$, where A is downward closed, i.e, whenever $\beta_i \le \alpha_i\,, \forall i = 1,\ldots m\,, \alpha \in A \implies \beta \in A$

lp-degree spaces

$A_{m,n,p} = \{\alpha \in \mathbb{N}^m : \|\alpha\|_p \le n\}$, with *total* $(p=1)$, *Euclidean* $(p=2)$, and *maximum degree* $(p=\infty)$, being crucial.

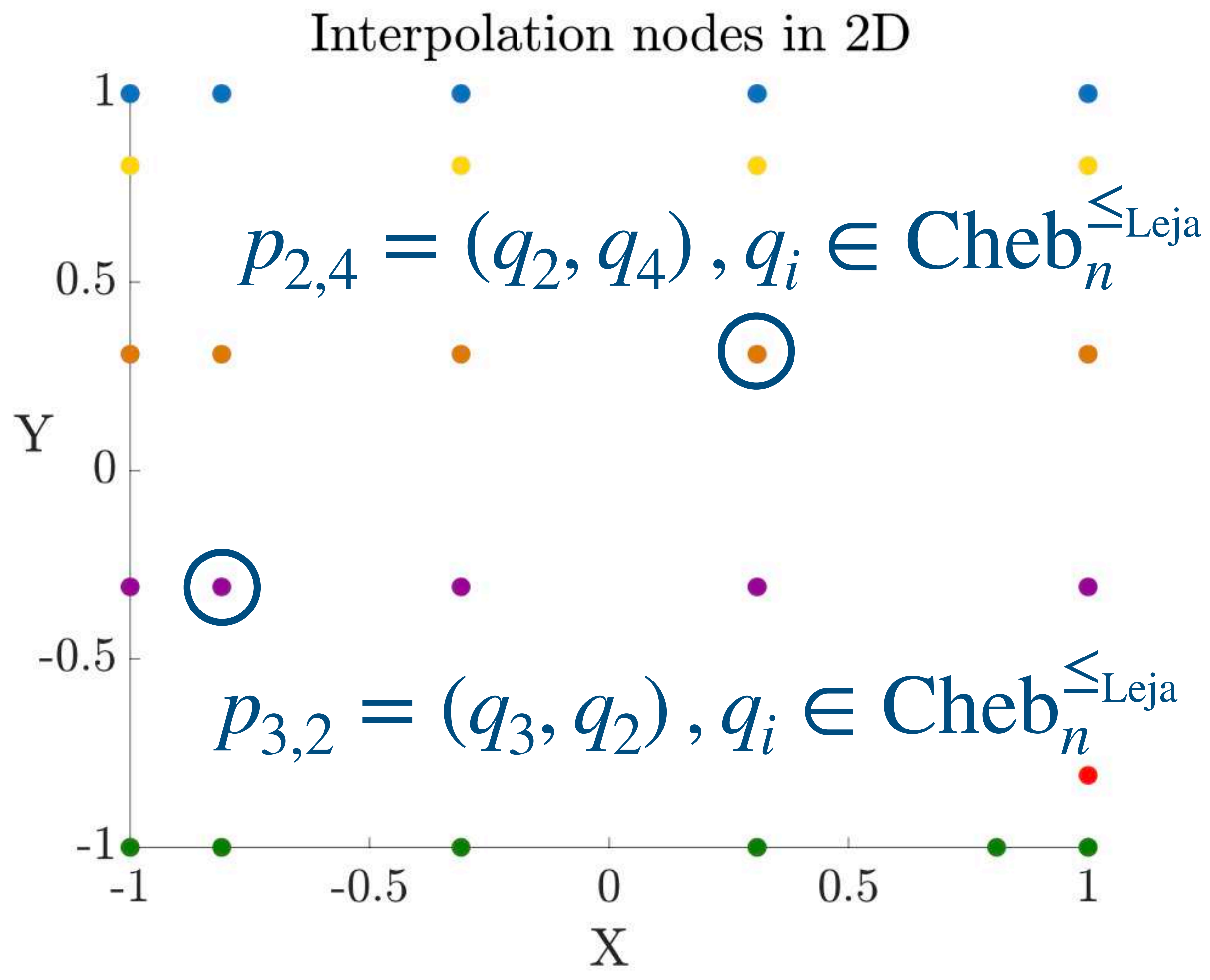
Unisolvent interpolation nodes

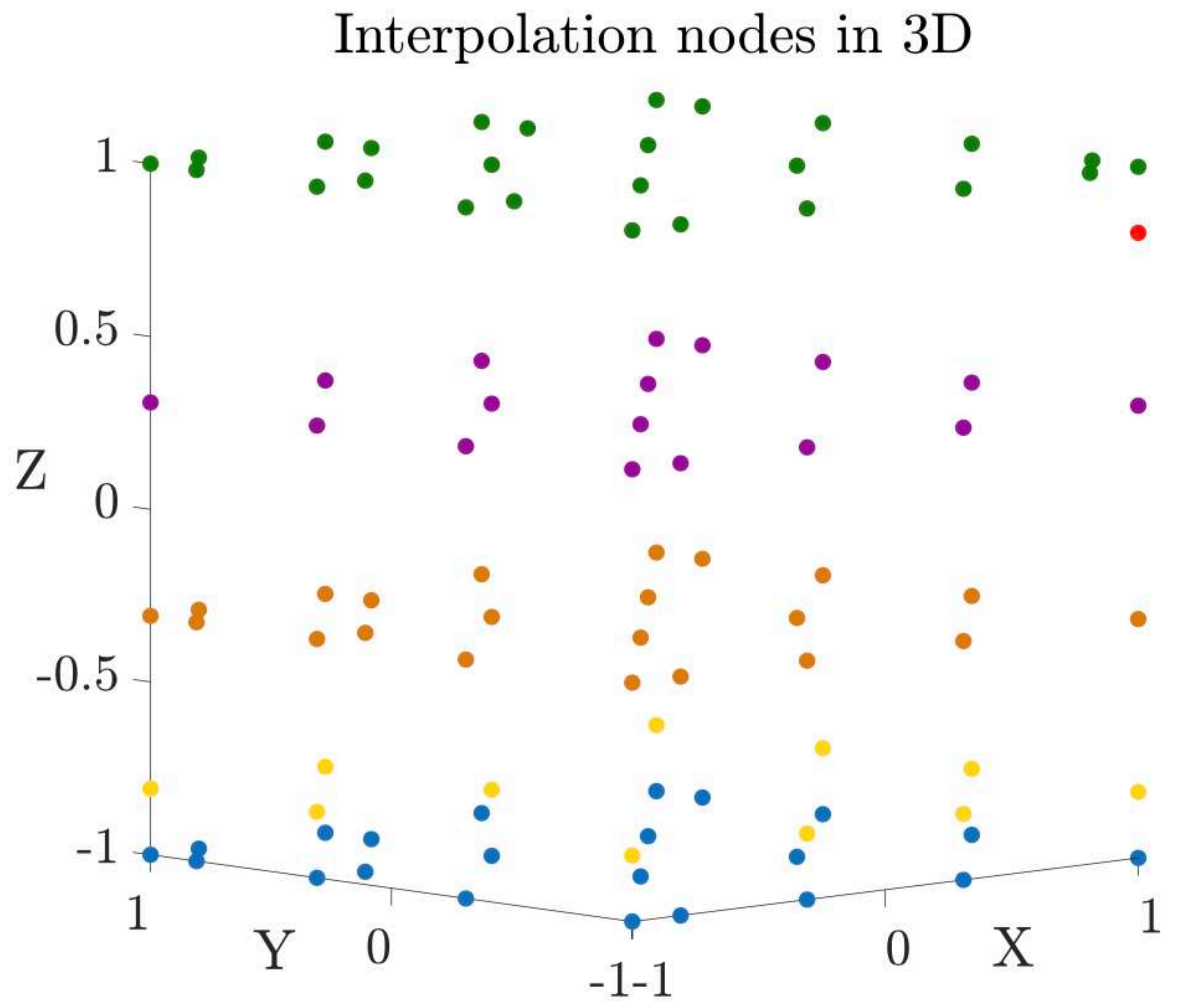
$P_A = \{p_\alpha = (p_{\alpha_1}, \ldots, p_{\alpha_m}) : \alpha \in A\,, p_{\alpha_i} \in P_i\}$, where $P_i = \{p_0, \ldots, p_n\} \subseteq [-1,1]\,, i = 1,\ldots,m.$

Leja-ordered Chebyshev-Lobatto (LCL) nodes

$$\mathrm{Cheb}_n^{\le_{\mathrm{Leja}}} = \left\{\cos\left(\frac{k\pi}{n}\right) : 0 \le k \le n\right\}^{\le_{\mathrm{Leja}}}, P_i = \mathrm{Cheb}_n^{\le_{\mathrm{Leja}}}\,, i = 1,\ldots,m.$$

Leja ordered Chebyshev—Lobatto (LCL) nodes

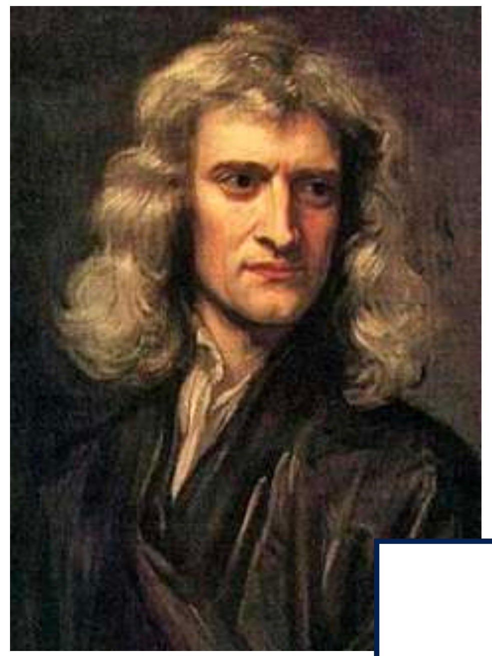




$$\boxed{P_A = \{p_\alpha : \alpha \in A = A_{m,n,p}\}\,, \quad p_\alpha = (p_{\alpha_1}, \ldots, p_{\alpha_m})\,, \quad p_{\alpha_i} \in \mathrm{Cheb_n}^{\leq_{\mathrm{Leja}}}\,, i = 1,\ldots,m\,.}$$

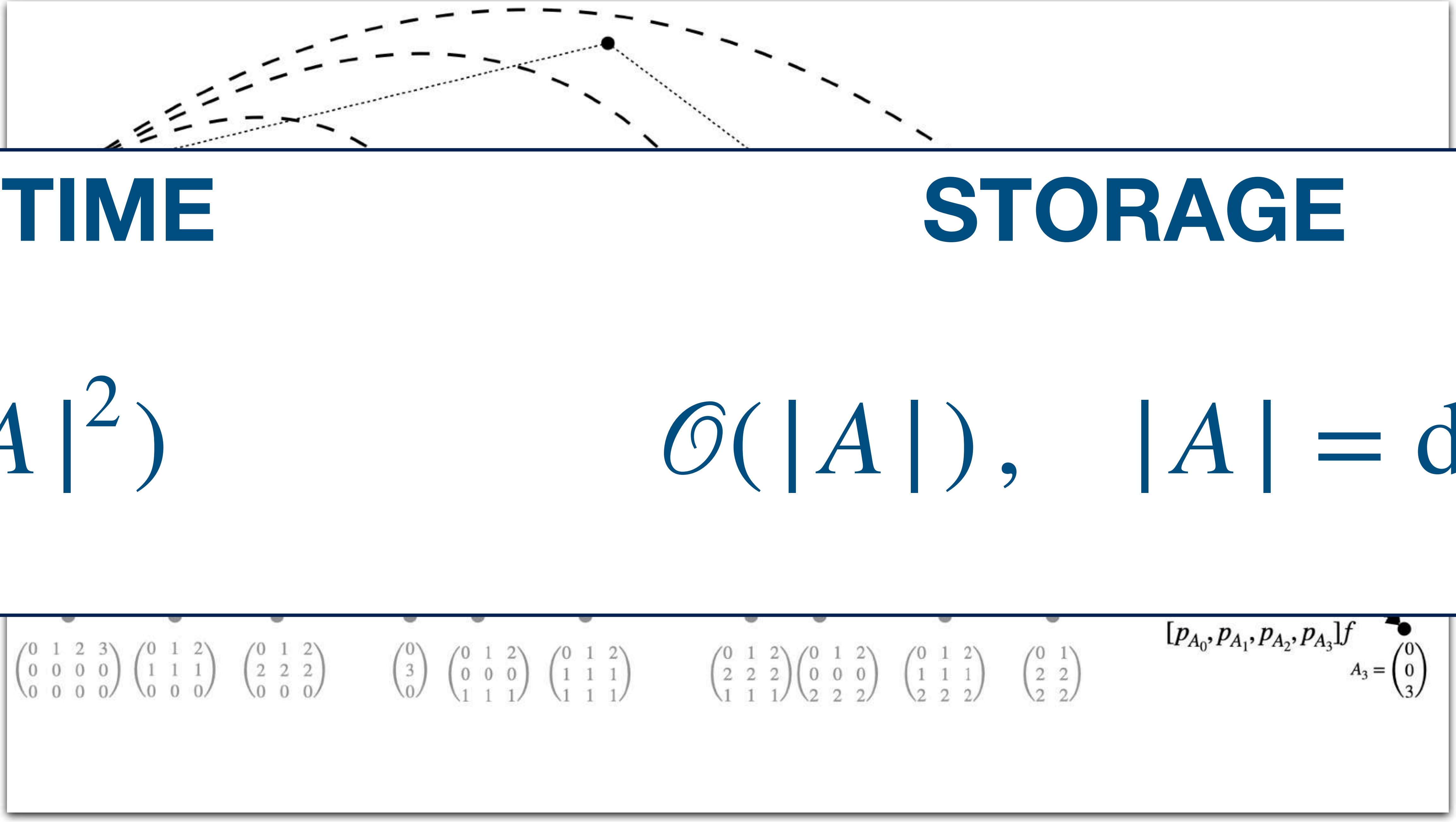
Kuntzmann J. (1960) Methodes numeriques interpolation derivees. Dunod Editeur, Paris.
Guenther, R. B., & Roetman, E. L. (1970). Some observations on interpolation in higher dimensions. *Mathematics of Computation*, *24*(111), 517-522.
Chung, K. C., & Yao, T. H. (1977). On lattices admitting unique Lagrange interpolations. *SIAM Journal on Numerical Analysis*, *14*(4), 735-743.
Chkifa, A., Cohen, A., & Schwab, C. (2014). High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Foundations of Computational Mathematics*, *14*, 601-633.

minterpy



Sir Isaac Newt
1643-1726

Based on Newton interpolation for **downward closed sets** due to a multivariate divided difference scheme (DDS)



Multivariate Divided Difference Scheme (DDS)

RUNTIME	STORAGE
$\mathcal{O}(\lvert A \rvert^2)$	$\mathcal{O}(\lvert A \rvert)\,, \quad \lvert A \rvert = \dim \Pi_A$

minterpy a

- **PDE solv** **differential geometry**,
- black box optimisation, auto encoder regularisation, model inference etc…

Acknowledgements

FLATIRON INSTITUTE
Center for Computational Mathematics

center for systems biology dresden

Ivo F. Sbalzarini

Michael Bussmann

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING
www.casus.science

MINTERPY

Uwe Hernandez Acosta **Sachin K.T. Veettil** **Damar Wicaksono** **Jannik Michelfeit** **Nico Hoffmann**

Tal-Ezer, H. (1988). High degree interpolation polynomial in Newton form (No. ICASE-88-39).

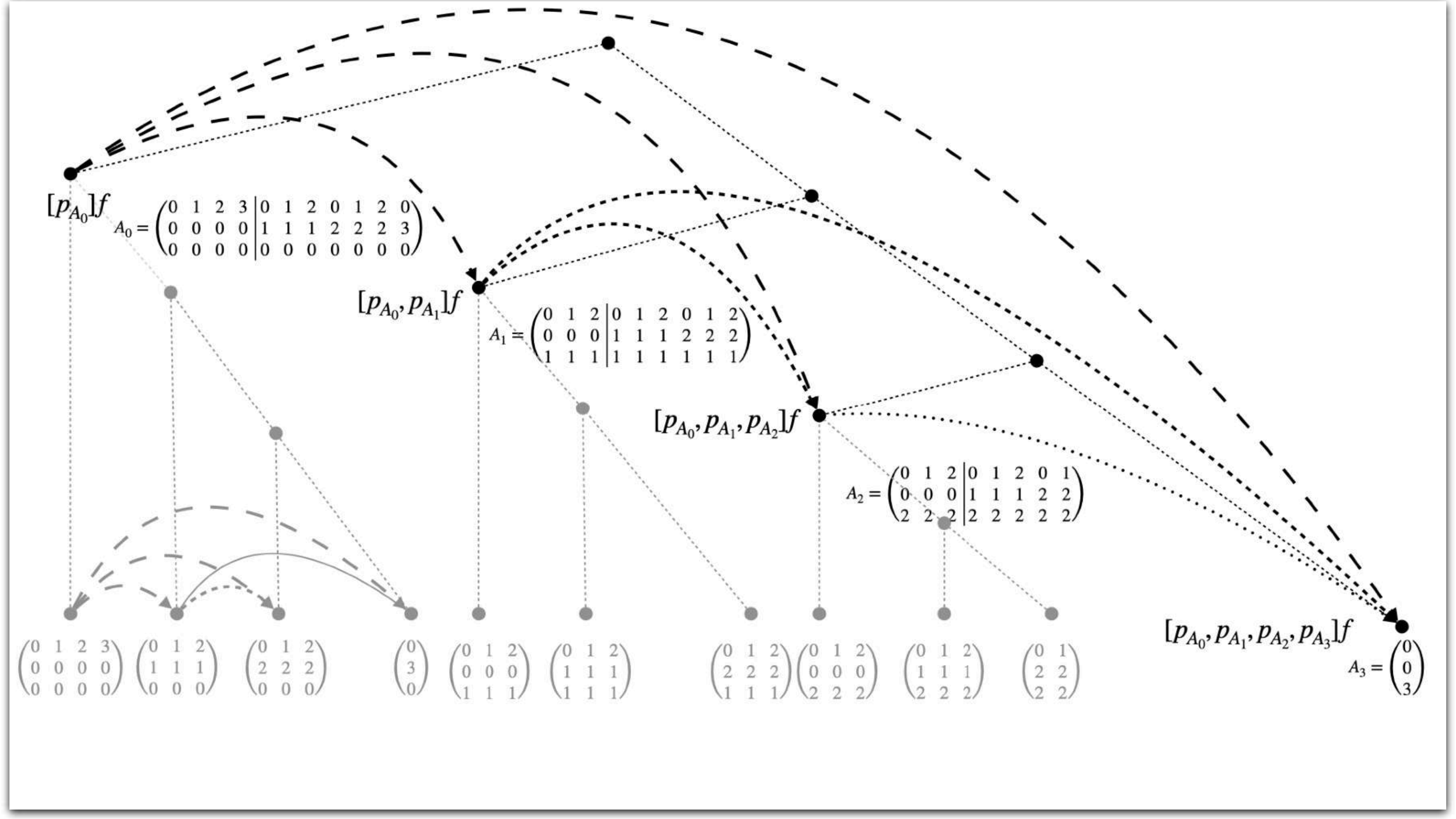
minterpy Multivariate interpolation, Python (2021) https://github.com/casus/minterpy

minterpy

Sir Isaac Newton
1643-1726

Joseph-Louis Lagrange
1736-1813

Based on Newton interpolation for **downward closed sets** due to a multivariate divided difference scheme (DDS)



Multivariate Divided Difference Scheme (DDS)

minterpy applications

- **PDE solvers & numerical differential geometry,**
- black box optimisation, auto encoder regularisation, model inference etc…

MINTERPY

Uwe Hernandez Acosta

Sachin K.T. Veettil

Damar Wicaksono

Jannik Michelfeit

Nico Hoffmann

Acknowledgements

Leslie Greengard

FLATIRON INSTITUTE
Center for Computational Mathematics

Ivo F. Sbalzarini

center for systems biology dresden

Michael Bussmann

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

www.casus.science

Tal-Ezer, H. (1988). High degree interpolation polynomial in Newton form (No. ICASE-88-39).

minterpy Multivariate interpolation, Python (2021) https://github.com/casus/minterpy

How powerful is lp-degree Chebyshev expansion ?

Trefethen's Theorem If $f : \square_m \to \mathbb{R}$ is analytic in the **Trefethen domain**

$$N_{m,\rho} = \left\{(z_1, \ldots, z_m) \in \mathbb{C}^m : (z_1^2 + \cdots + z_m^2) \in E^2_{m,h^2}\right\} \subseteq \mathbb{C}^m ,$$

where E^2_{m,h^2} is the **Newton ellipse** with foci 0 and m and leftmost point $-h^2$, $h \in [0,1]$. Then the truncation $\mathcal{T}_{A_{m,n,p}}(f)$ of the Chebyshev series $f(x) = \sum_{\alpha \in \mathbb{N}} c_\alpha T_\alpha(x)\,, c_\alpha \in \mathbb{R}\,,\quad T_\alpha(x) = \prod_{i=1}^m T_{\alpha_i}(x_i)\,,\quad \mathcal{T}_{A_{m,n,p}}(f) = \sum_{\alpha \in A_{m,n,p}} c_\alpha T_\alpha(x)\,,$ to $\Pi_{A_{m,n,p}}$ yields the errors

$$\|f - \mathcal{T}_{A_{m,n,p}}(f)\|_{C^0(\Omega)} = \begin{cases} \mathcal{O}_\varepsilon(\rho^{-n/\sqrt{m}}) & p = 1 \\ \mathcal{O}_\varepsilon(\rho^{-n}) & p = 2 \\ \mathcal{O}_\varepsilon(\rho^{-n}) & p = \infty \end{cases} , \qquad \rho = h + \sqrt{1 + h^2} > 1\,.$$

$$|A_{m,n,1}| = \binom{m+n}{m} \in \mathcal{O}(m^n) \qquad |A_{m,n,2}| \approx \frac{(n+1)^m}{\sqrt{\pi m}}\left(\frac{\pi e}{2m}\right)^{m/2} \in o(n^m) \qquad |A_{m,n,\infty}| = (n+1)^m \in \mathcal{O}(m^n)$$

total degree | **Euclidean degree** | **maximum degree**

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

Is the converse also true ?

Trefethen's Conjecture: If $f : \square_m \to \mathbb{R}$ possesses a polynomial approximation of geometric rate

$$\|f - p_n^*\|_\infty = \mathcal{O}(\rho^{-n}) \ .$$

Then f can be analytically extended to $N_{m,\rho}$.

Bos—Levenberg Theorem: Let $f : K \to \mathbb{C}$, $K \subseteq \mathbb{C}^m$, PL-regular $\Pi_n = \Pi(nP)$, P is a convex body. Then

$$\|f - p_n^*\|_{C^0(\Omega)} \lesssim \rho^{-n} , \quad \rho = \rho(P) , \quad \iff \quad f = F_{|K} \text{ with } F \text{ holomorphic in } \Omega_\rho ,$$

where the **Bos-Levenberg domain** $\Omega_\rho = \left\{ z \in \mathbb{C}^m : |Q(z)| < \log(\rho) , \text{ for all } Q \in \Pi(nP) , \|Q_{|K}\|_\infty \leq 1 \right\} \subset \mathbb{C}^m$ is precompact.

L. Bos & N. Levenberg Bernstein–Walsh theory associated to convex bodies and applications to multivariate approximation theory. *Computational Methods and Function Theory* (2018)

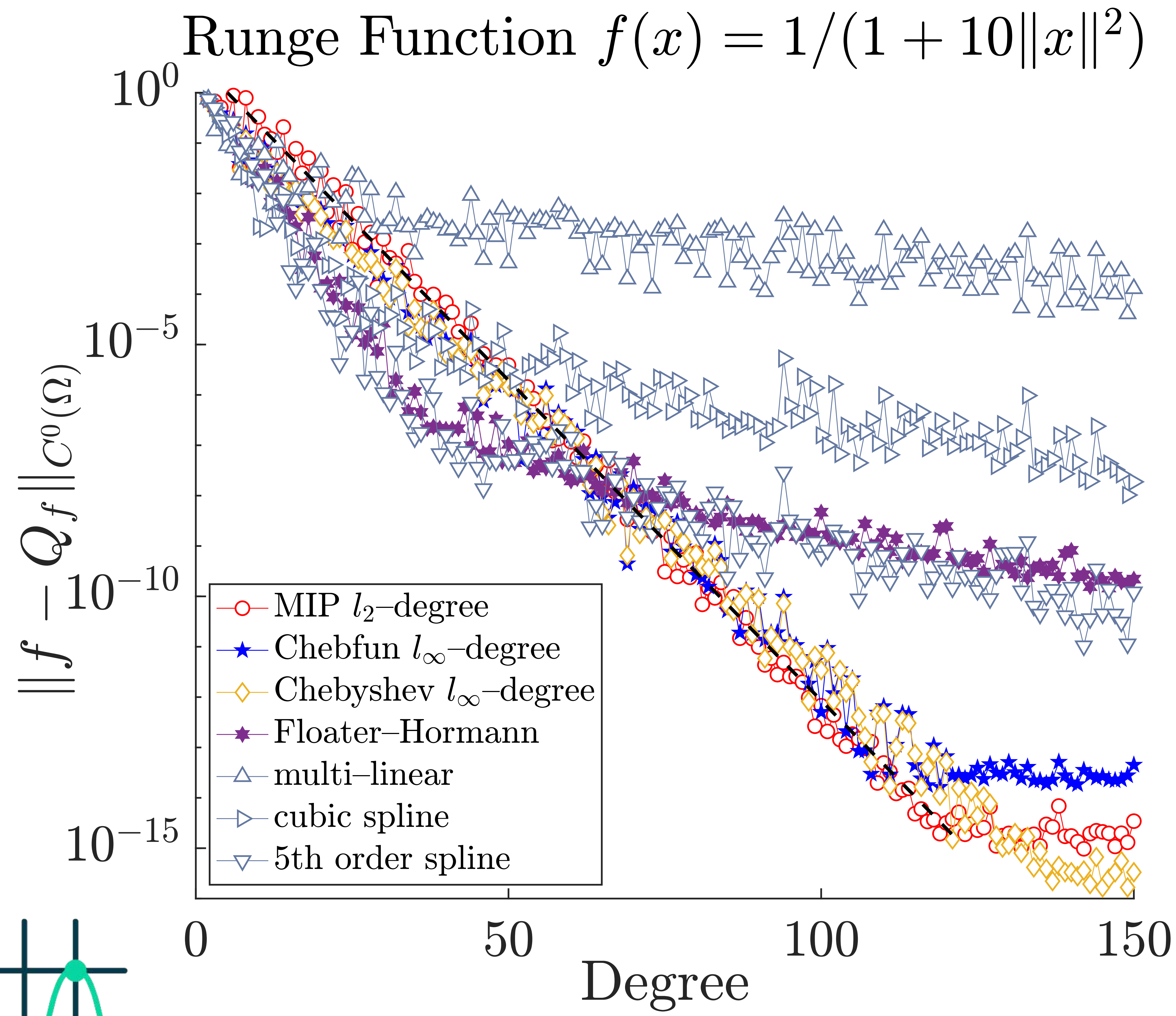
Theorem: Let $f : \square_m \to \mathbb{R}$, $f = F_{|\square_m}$, F holomorphic in Ω_ρ, $\Pi_n = \Pi_{m,n,p}$ $(P = A_{m,n,p})$ and $k \in \mathbb{N}$. Then

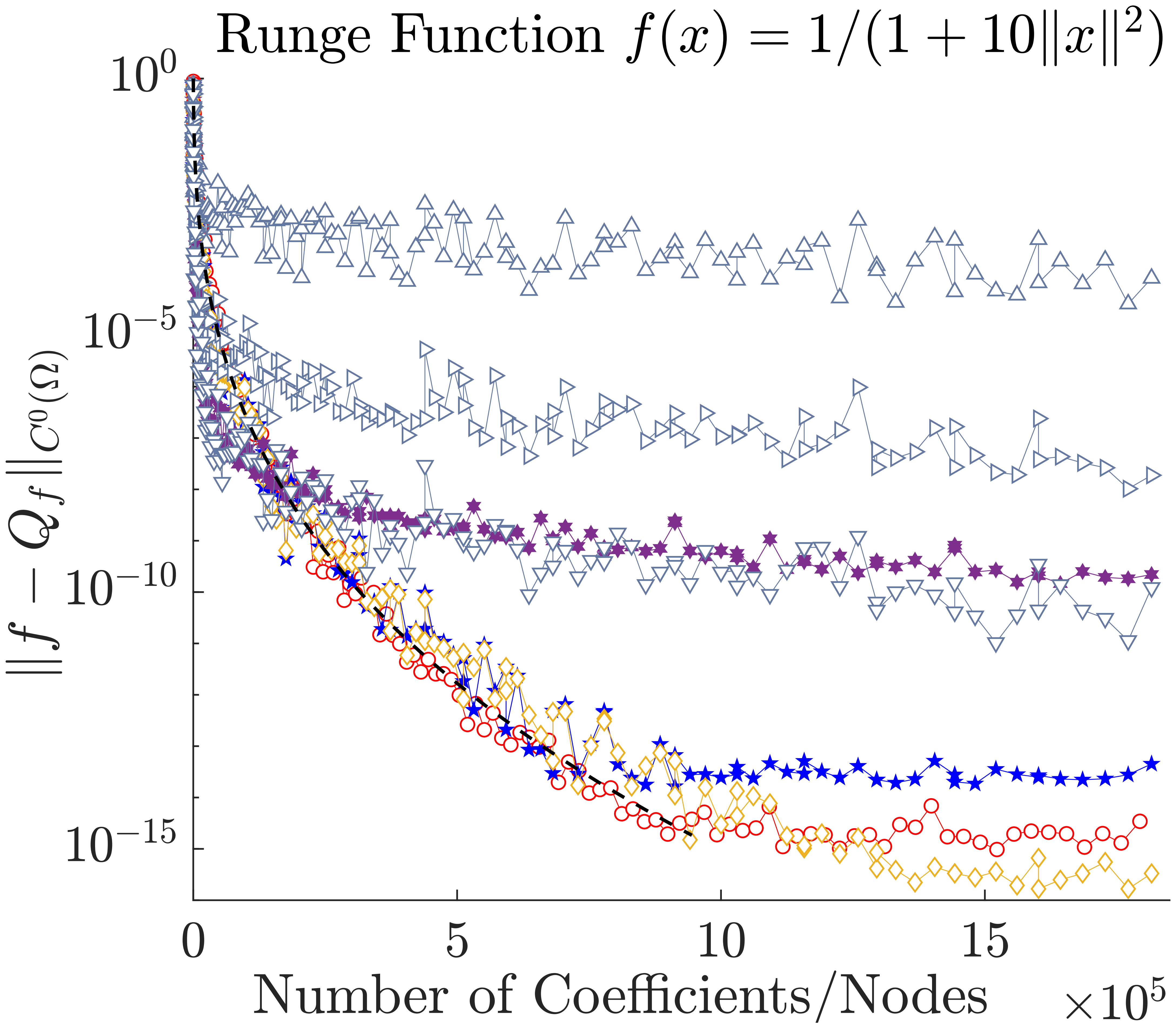
$$\|f - Q_n\|_{C^k(\square_m)} = \mathcal{O}_\varepsilon(\rho^{-n}) , \quad \rho = \rho_p ,$$

where $Q_n = Q_{f,P_{A_{m,n,p}}}$ is the interpolant of f in LCL-nodes.

Chkifa, A., Cohen, A., & Schwab, C. (2014). High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Foundations of Computational Mathematics*, *14*, 601-633.

Interpolating the Runge function in 3D

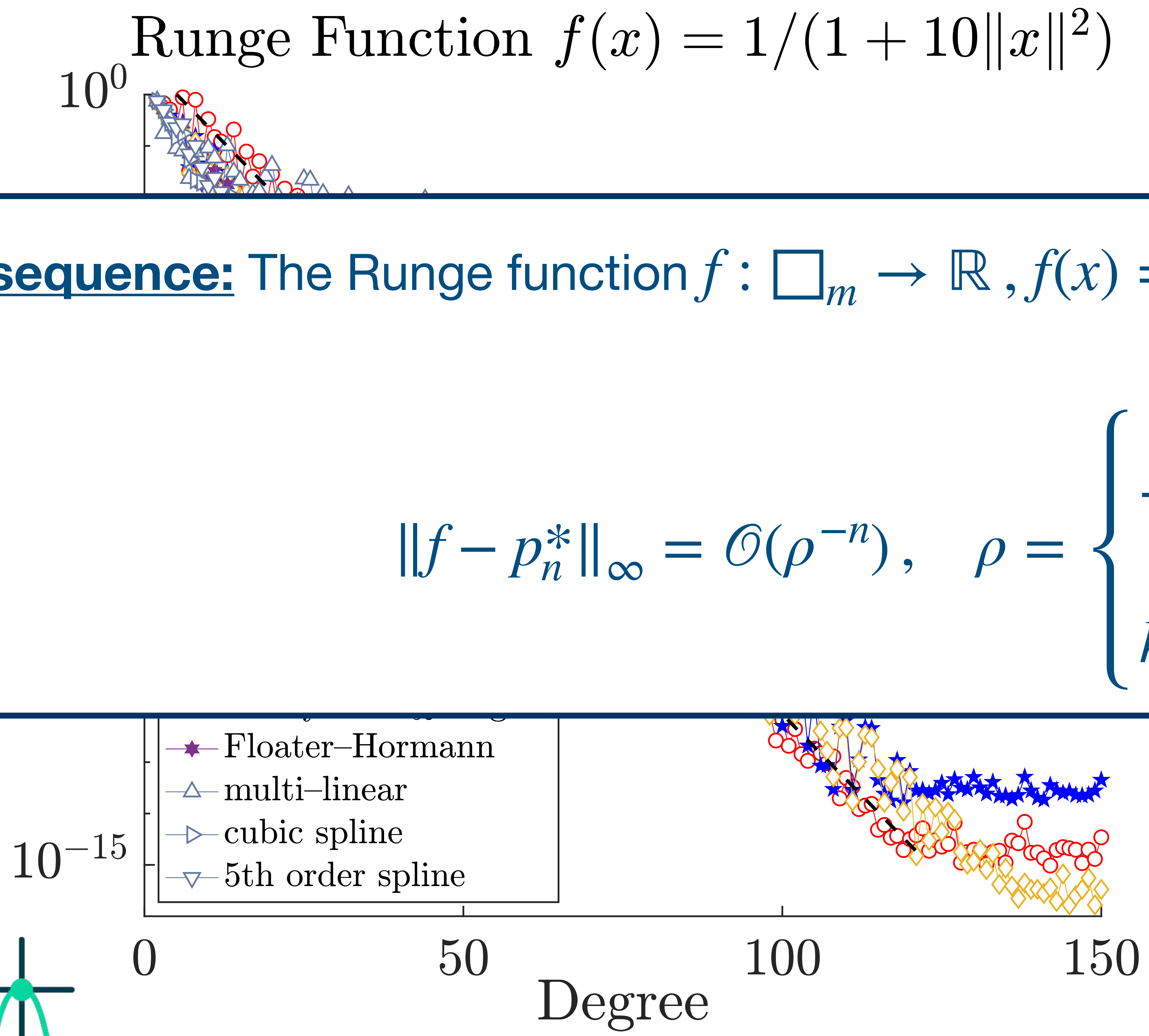


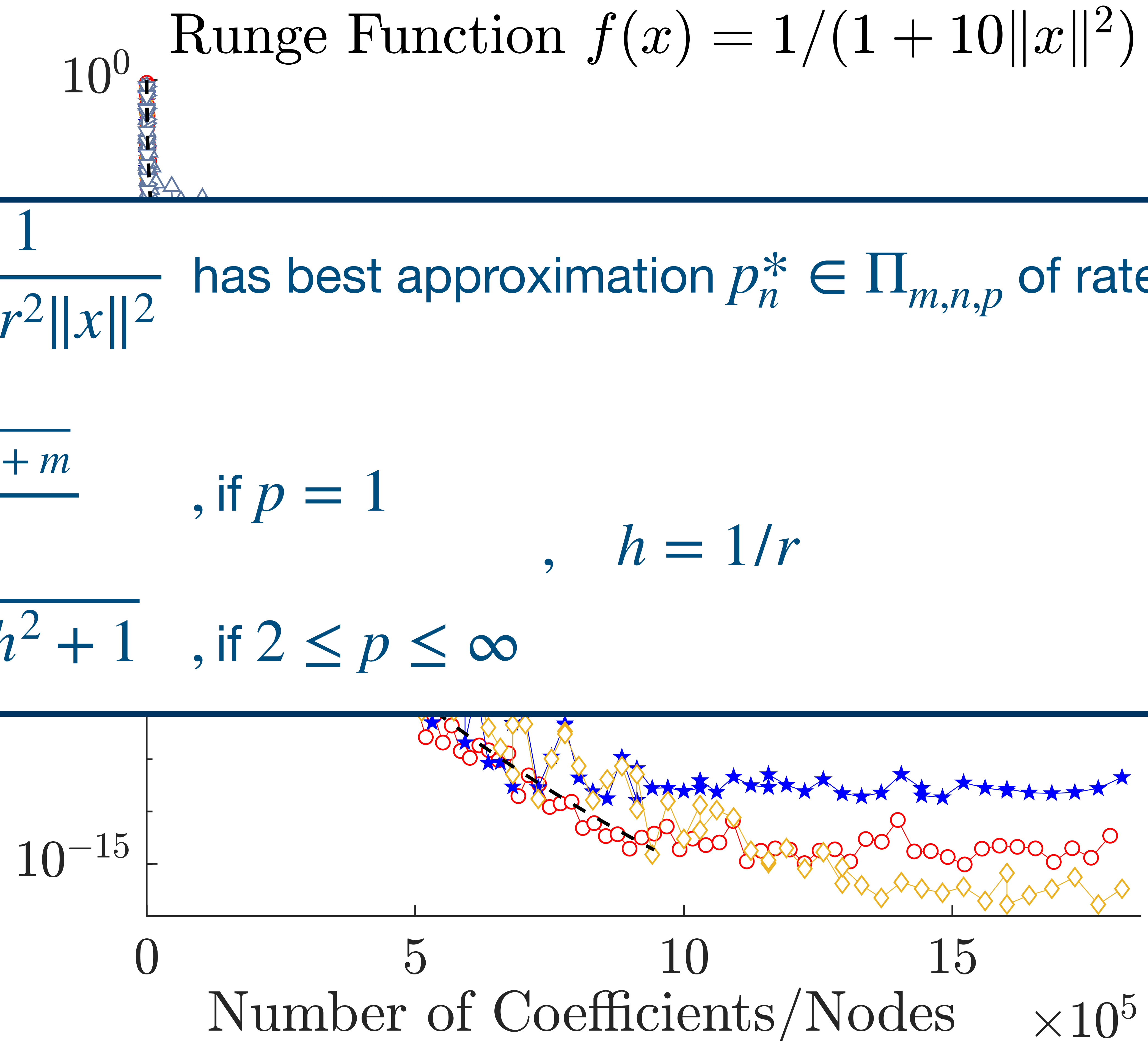


MINTERPY

minterpy Multivariate interpolation, Python (2021) https://github.com/casus/minterpy

Interpolating the Runge function in 3D

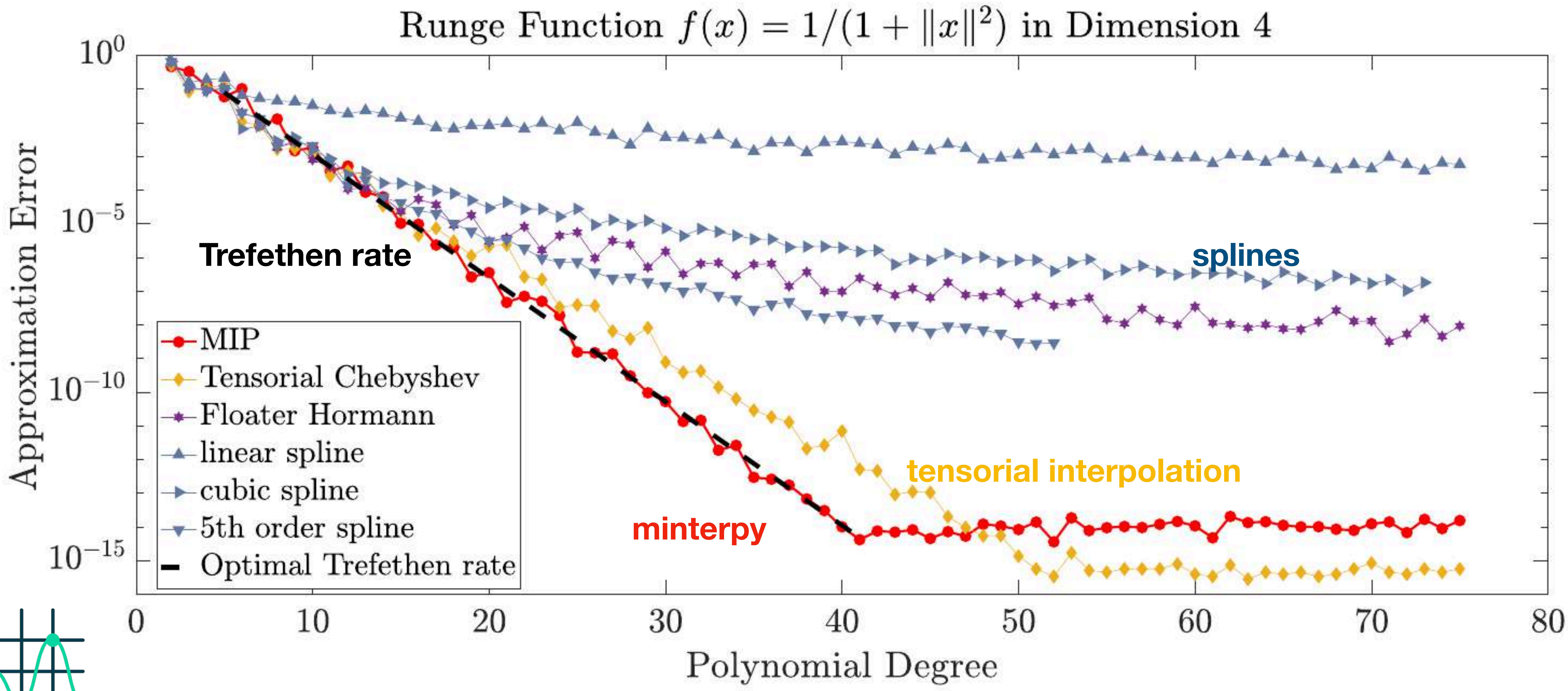




Consequence: The Runge function $f : \square_m \to \mathbb{R}\,,\ f(x) = \dfrac{1}{1 + r^2\|x\|^2}$ has best approximation $p_n^* \in \Pi_{m,n,p}$ of rate

$$\|f - p_n^*\|_\infty = \mathcal{O}(\rho^{-n})\,, \quad \rho = \begin{cases} \dfrac{h + \sqrt{h^2+m}}{\sqrt{m}} & \text{, if } p = 1 \\ h + \sqrt{h^2+1} & \text{, if } 2 \le p \le \infty \end{cases}, \quad h = 1/r$$

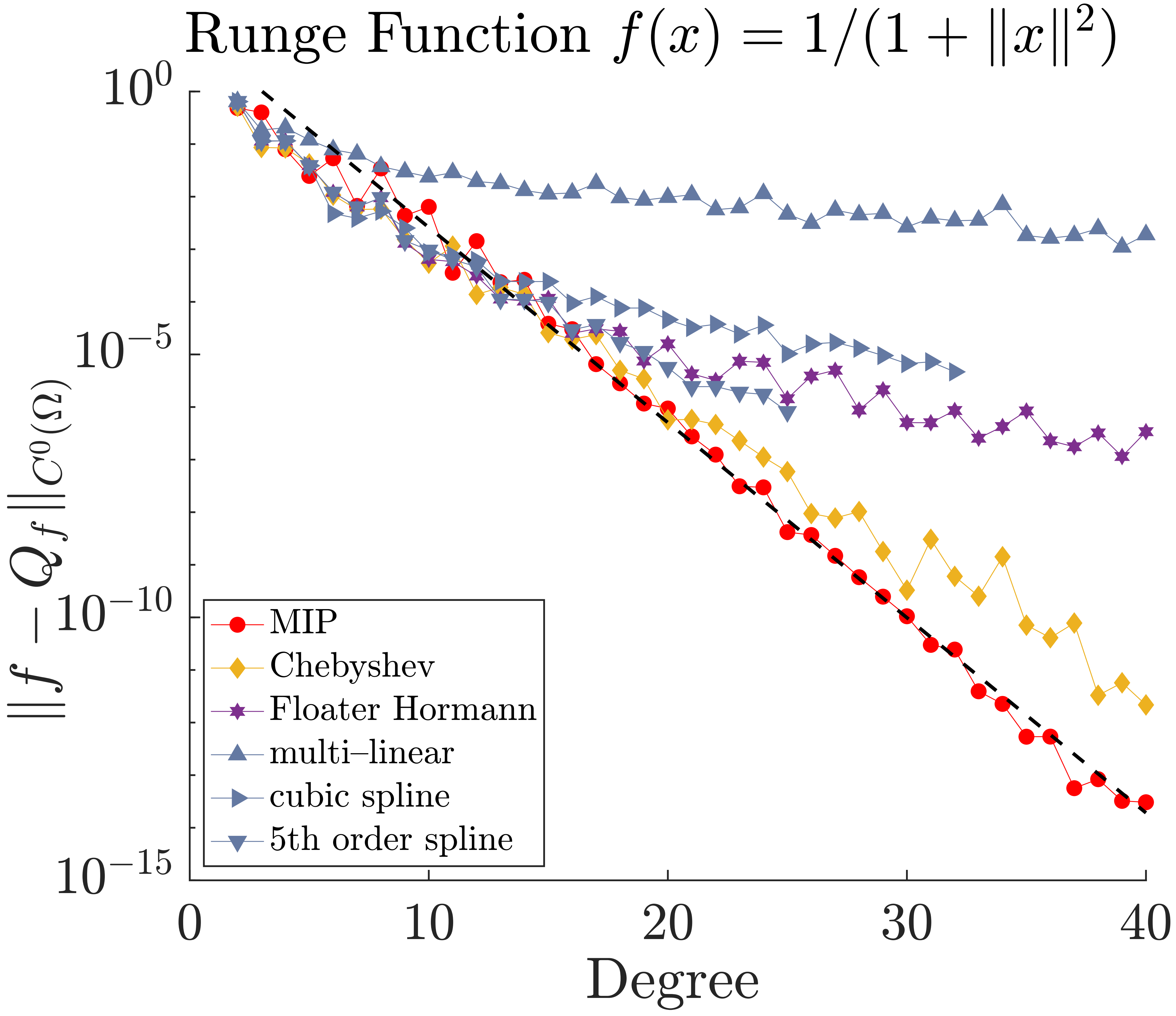
Interpolating the Runge function in 4D

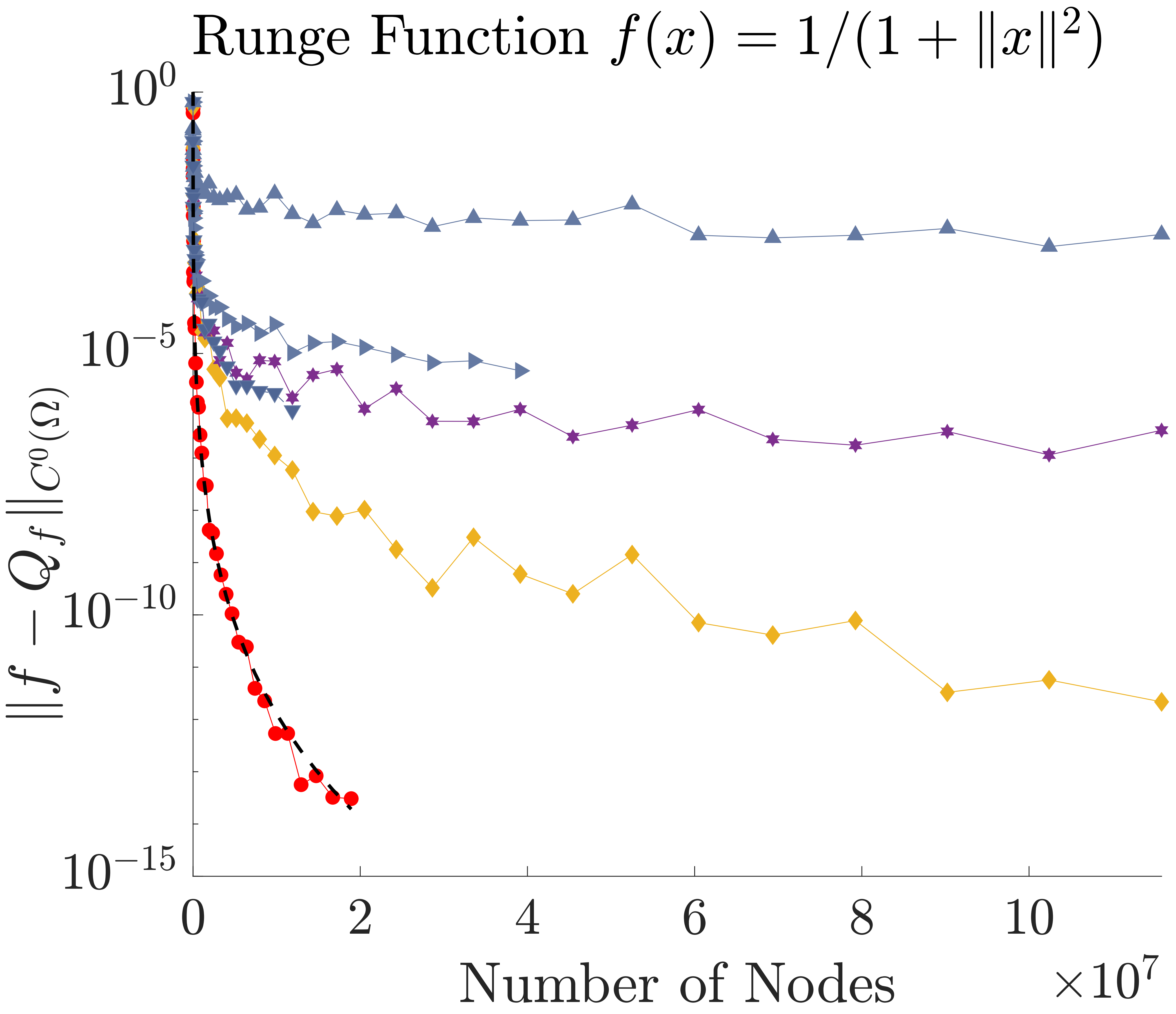


MINTERPY

minterpy Multivariate interpolation, Python (2021) https://github.com/casus/minterpy

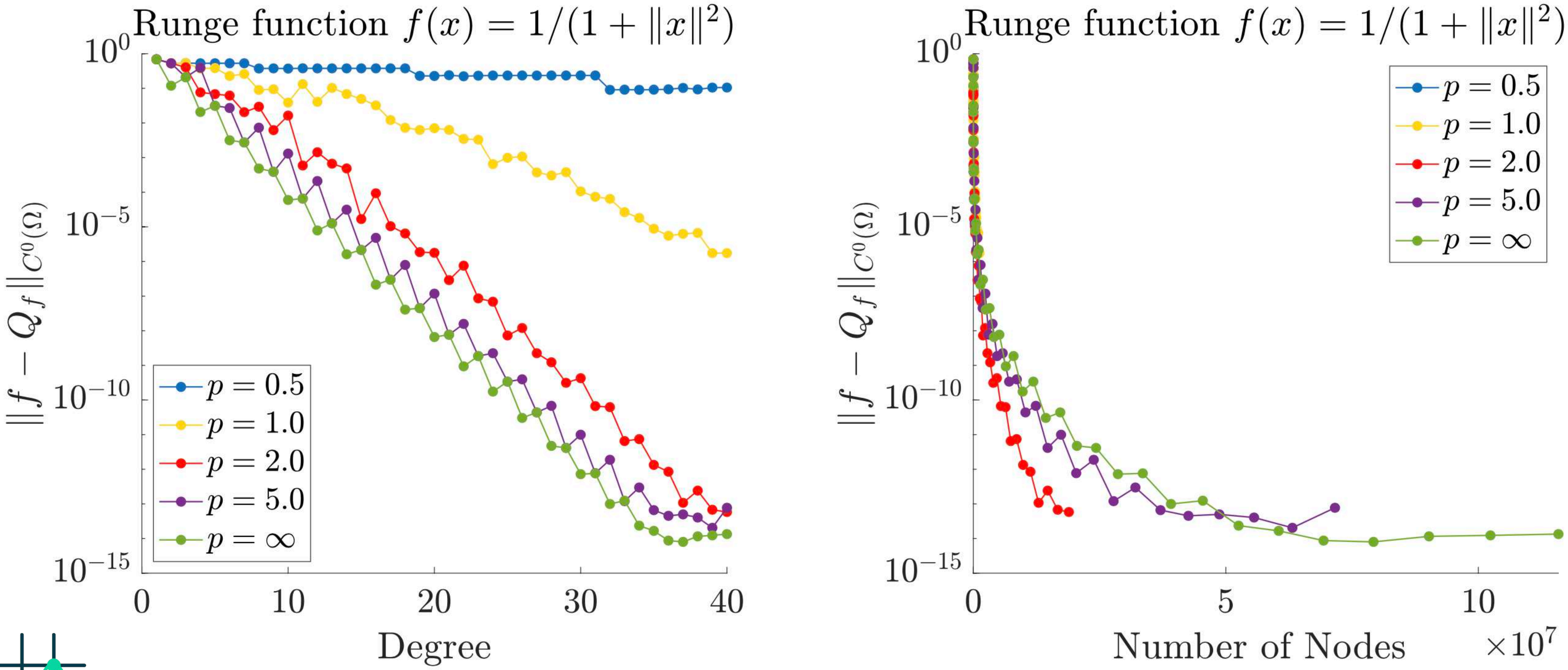
l_p−degree interpolation in dimension $m = 5$





function	dim	fit range	ρ_{MIP}	c	ρ
$f_R(x) = 1/(1+10\|x\|^2)$	2	$2 \sim 121$	1.35	4.30	1.365
$f_R(x) = 1/(1+10\|x\|^2)$	3	$2 \sim 121$	1.34	4.41	1.365
$f_R(x) = 1/(1+1\|x\|^2)$	4	$2 \sim 40$	2.33	5.40	2.41
$f_R(x) = 1/(1+1\|x\|^2)$	5	$2 \sim 40$	2.35	13.37	2.41

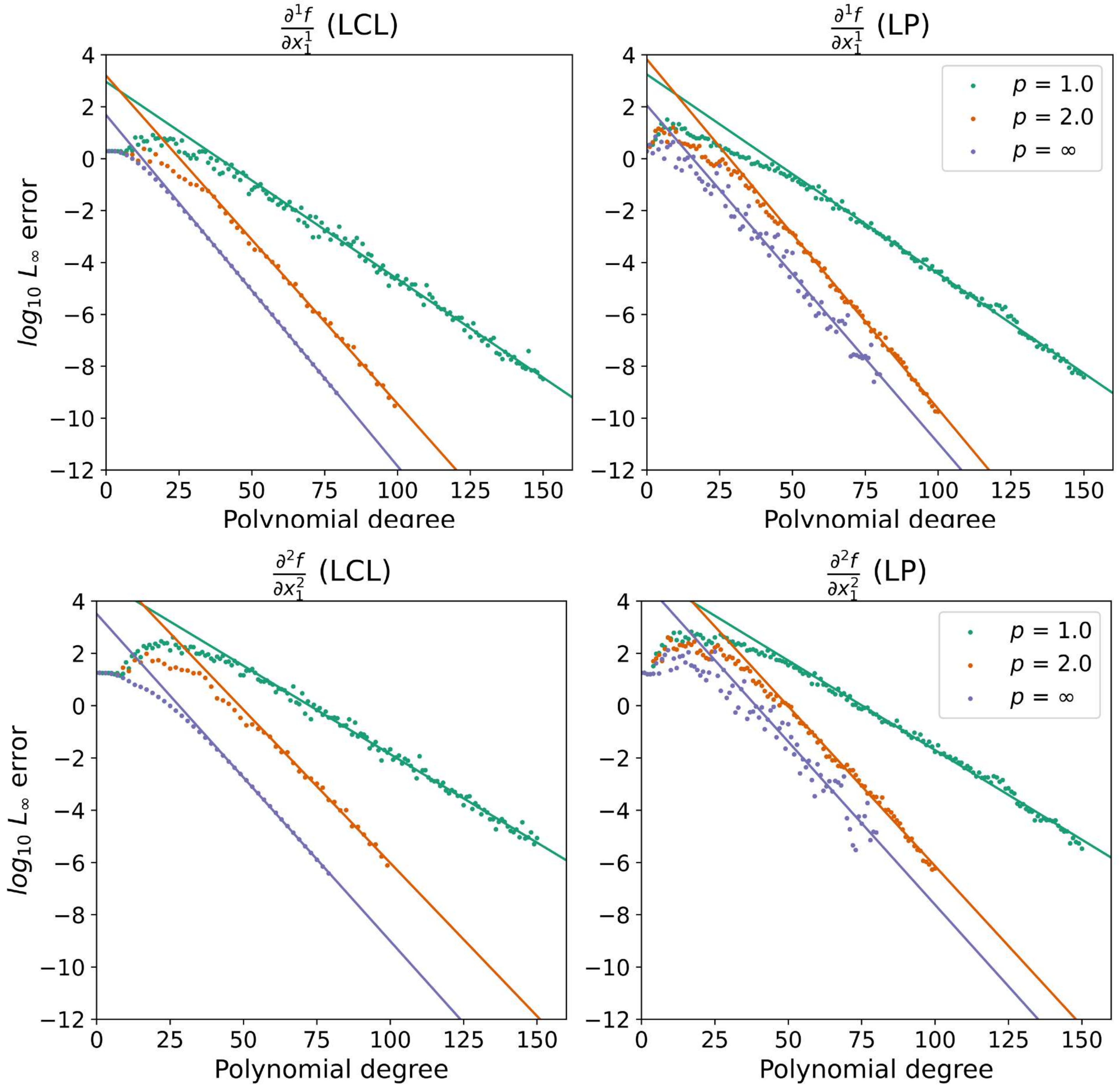
Interpolation in Dimension 5



MINTERPY

minterpy Multivariate interpolation, Python (2021) https://github.com/casus/minterpy

Differentiation of the Runge function



r	p	$m=2$ LCL	$m=2$ LP	$m=3$ LCL	$m=3$ LP	$m=4$ LCL	$m=4$ LP
1	1.0	1.911	1.896	1.700	1.703	1.585	1.584
	2.0	2.332	2.351	2.313	2.353	2.303	2.360
	∞	2.408	2.349	2.412	2.359	2.408	2.371
3	1.0	1.252	1.255	1.201	1.204	1.175	1.169
	2.0	1.360	1.373	1.370	1.375	1.293	1.303
	∞	**1.387**	1.372	**1.387**	1.367	1.367	1.346
5	1.0	1.145	1.147	1.116	1.115	0.000	0.000
	2.0	1.206	1.212	1.208	1.209	0.000	0.000
	∞	**1.219**	1.209	**1.219**	1.209	0.000	0.000

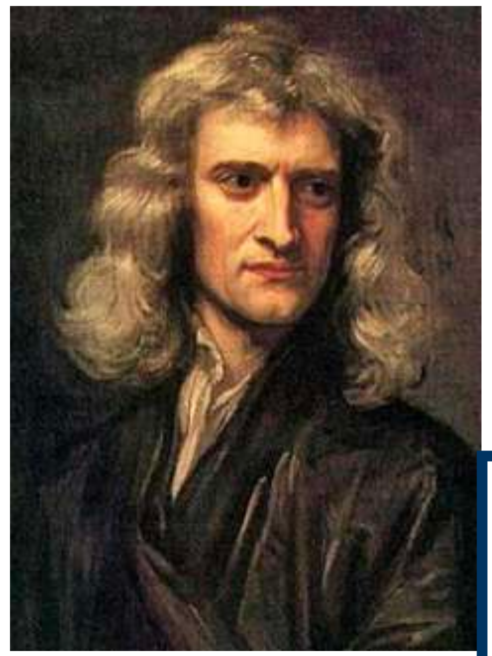
Table 6: Approximation rates of LCL-node and LP-node interpolants of the Runge function **F2**). Optimal rates are marked bold.

p	$\frac{\partial f}{\partial x_1}$ LCL	$\frac{\partial f}{\partial x_1}$ LP	$\frac{\partial^2 f}{\partial x_1^2}$ LCL	$\frac{\partial^2 f}{\partial x_1^2}$ LP
1.0	1.191	1.193	1.169	1.171
2.0	1.338	1.364	1.310	1.325
∞	1.365	1.349	1.334	1.333

Table 7: Approximation rates of derivatives of LCL-node and LP-node interpolants of the Runge function **F2**) in dimension $m = 3$, with $r = 3$.

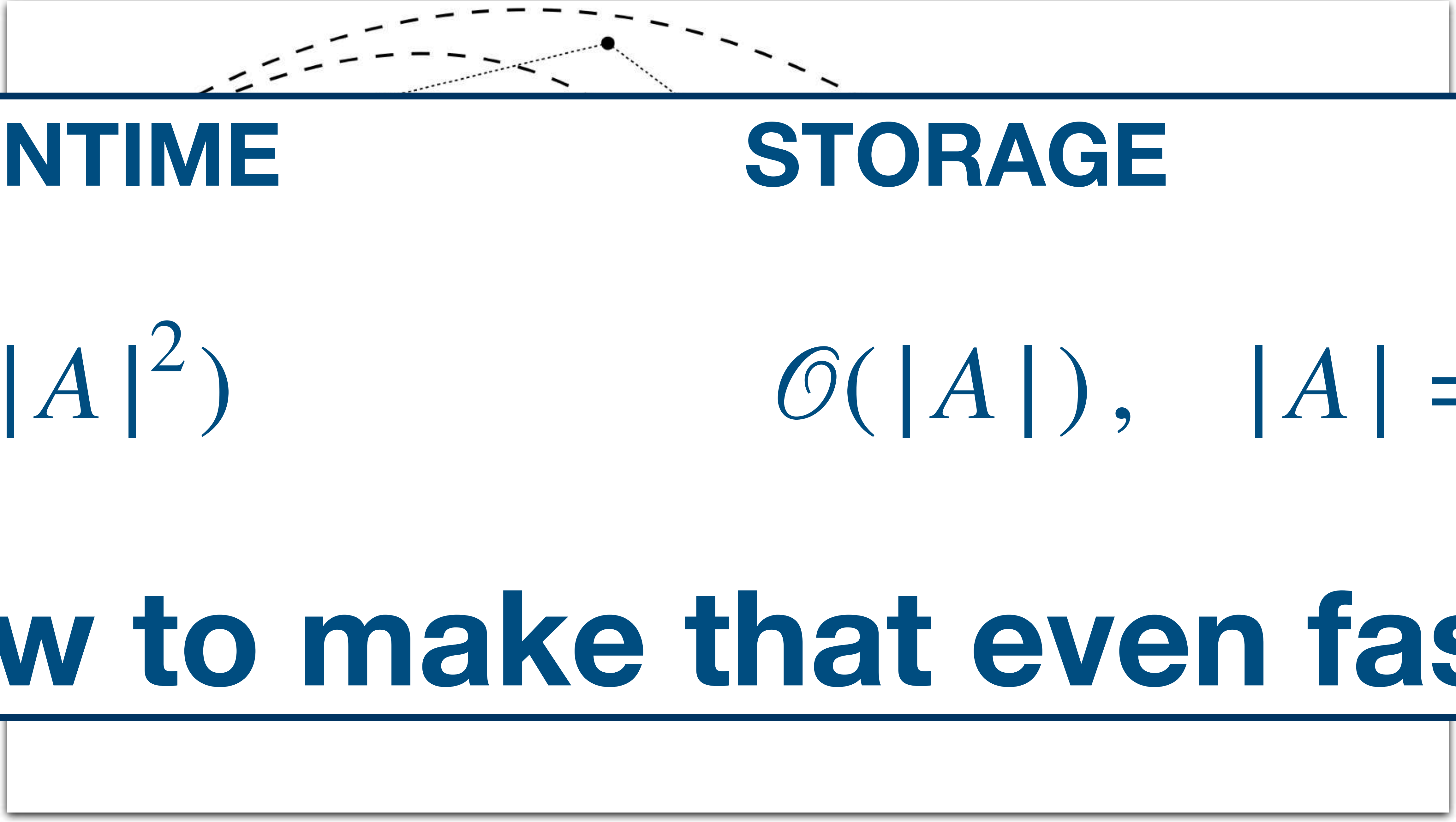
MINTERPY

minterpy



Sir Isaac Newton
1643-1726

Based on Newton interpolation for **downward closed sets** due to a multivariate divided difference scheme (DDS)



Multivariate Divided Difference Scheme (DDS)

RUNTIME

$$\mathcal{O}(|A|^2)$$

STORAGE

$$\mathcal{O}(|A|)\,, \quad |A| = \dim \Pi_A$$

How to make that even fast ?

minterpy app

- **PDE solvers differential**
- black box optimisation, auto encoder regularisation, model inference etc…

Acknowledgements

FLATIRON INSTITUTE
Center for Computational Mathematics

center for systems biology dresden

Michael Bussmann

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING
www.casus.science

MINTERPY

Uwe Hernandez Acosta

Sachin K.T. Veettil

Damar Wicaksono

Jannik Michelfeit

Nico Hoffmann

minterpy Multivariate interpolation, Python (2021) https://github.com/casus/minterpy

Relevant publications and contributions

Hecht, M., and Sbalzarini, I.F. Fast interpolation and Fourier transform in high-dimensional spaces. In Intelligent Computing. Proc. 2018 IEEE Computing Conf., Vol.2, (London, UK), K. Arai, S. Kapoor, and R. Bhatia, Eds., vol. 857 of Advances in Intelligent Systems and Computing, Springer Nature, pp. 53–75, 2018.

preprints:

Hecht, M., Gonciarz, K., Michelfeit, J., Sivkin, V., and Sbalzarini, I.F. Multivariate Interpolation in Unisolvent Nodes–Lifting the Curse of Dimensionality, arXiv:2010.10824 , 2020.

Hecht, M., Hoffmann, K.B., Cheeseman, B.L., and Sbalzarini, I.F. Multivariate Newton interpolation. arXiv:1812.04256, 2018.

Hecht, M., Hoffmann, K.B., Cheeseman, B.L., and Sbalzarini, I.F. A Quadratic-Time Algorithm for General Multivariate Polynomial Interpolation. arXiv:1710.10846, 2017.

in preparation:

Hecht, M., Wicaksono, D., Gonciarz, K., Michelfeit, J., Sivkin, V., and Sbalzarini, I.F. Multivariate Newton Interpolation Reaches the Optimal Approximation Rates for Bos–Levenberg– Trefethen Functions, submission planned to IMA Journal of Numerical Analysis, Oxford Academic

Hofmann, P.A., Wicaksono, D., and Hecht, M. The Fast Newton Transform: Interpolation in Downward Closed Spaces

Relevant publications and contributions

software releases:

Zavalani, G., and Hecht, M. Surfgeopy: A Python3 library for numerical differential geoemetry on regular surfaces, 2024, https://codebase.helmholtz.cloud/interpol/surfgeopy

Thekke Veettil, S.K., Zavalani, G., Hernandez Acosta, U., Sbalzarini, I.F., and Hecht, M. Global polynomial level sets for numerical differential geometry of smooth closed surfaces. Python library, https://github.com/minterpy-project/minterpy-levelsets,2023

Wicaksono, D., and Hecht, M. UQTestFuns: A Python3 library of uncertainty quantification (UQ) test functions, 2023. https://github.com/casus/uqtestfuns

Hernandez Acosta,U., Thekke Veettil, S. K., Wicaksono, D., and Hecht, M. minterpy – multivariate interpolation in python, 2022, https://github.com/casus/minterpy/

Fast Multivariate Interpolation in Downward–Closed Spaces

Phil-Alexander Hofmann, Damar Wicaksono, and Michael Hecht

CASUS – Center for Advanced Systems Understanding
HZDR – Helmholtz-Zentrum Dresden-Rossendorf
University Wrocław

October 31, 2024

SIGMA-2024

Outline

Overview

Introduction

m	...	spatial dimension
n	...	polynomial degree
$A \subset \mathbb{N}_0^m$	...	downward closed multi–index set (non–empty finite)
Π	...	downward-closed polynomial space

Introduction

m	...	spatial dimension
n	...	polynomial degree
$A \subset \mathbb{N}_0^m$	...	downward closed multi–index set (non–empty finite)
Π	...	downward-closed polynomial space

1. Provide a *framework* for multivariate interpolation in Π for any pair of

$$P' \ \dots \ \text{one-dimensional nodes}, \ \ Q \ \dots \ \text{polynomial basis}.$$

Introduction

m	...	spatial dimension
n	...	polynomial degree
$A \subset \mathbb{N}_0^m$	...	downward closed multi–index set (non–empty finite)
Π	...	downward-closed polynomial space

1. Provide a *framework* for multivariate interpolation in Π for any pair of

$$P' \;\ldots\; \text{one-dimensional nodes}, \quad Q \;\ldots\; \text{polynomial basis}.$$

2. Interpolation is realised on a *non-tensorial grid* P, where all nodes are grid points, but not vice versa.

Introduction

m	...	spatial dimension
n	...	polynomial degree
$A \subset \mathbb{N}_0^m$	...	downward closed multi–index set (non–empty finite)
Π	...	downward-closed polynomial space

1. Provide a *framework* for multivariate interpolation in Π for any pair of
$$P' \ \ldots \text{ one-dimensional nodes}, \ Q \ \ldots \text{ polynomial basis.}$$
2. Interpolation is realised on a *non-tensorial grid* P, where all nodes are grid points, but not vice versa.
3. Present an *algorithm* for Π that applies the backward and forward transform in
$$\mathcal{O}(N \cdot m \cdot n \cdot \kappa), \quad N := \dim(\Pi), \quad 1 \leq \kappa \leq m.$$

Introduction

m	...	spatial dimension
n	...	polynomial degree
$A \subset \mathbb{N}_0^m$	...	downward closed multi–index set (non–empty finite)
Π	...	downward-closed polynomial space

1. Provide a *framework* for multivariate interpolation in Π for any pair of
$$P' \ \ldots \text{ one-dimensional nodes}, \ Q \ \ldots \text{ polynomial basis.}$$
2. Interpolation is realised on a *non-tensorial grid* P, where all nodes are grid points, but not vice versa.
3. Present an *algorithm* for Π that applies the backward and forward transform in
$$\mathcal{O}(N \cdot m \cdot n \cdot \kappa), \quad N := \dim(\Pi), \quad 1 \leq \kappa \leq m.$$
4. The algorithm is designed for **any** Π, with a detailed analysis conducted on ℓ^p *degree polynomial spaces*, including Euclidean degree polynomials.

Trefethen, 2017

Chkifa, Cohen, Schwab, 2014

Overview

Framework

Downward Closed Polynomial Space

Definition (downward closed multi-index set)

(Finite) $A \subset \mathbb{N}_0^m$ downward closed, if $\forall \beta \in A \ : \ \beta \in A \ \Rightarrow \{\alpha \in \mathbb{N}^m \mid \alpha \leq \beta\} \subset A$.

Framework

Downward Closed Polynomial Space

Definition (downward closed multi-index set)

(Finite) $A \subset \mathbb{N}_0^m$ downward closed, if $\forall \beta \in A \; : \; \beta \in A \; \Rightarrow \{\alpha \in \mathbb{N}^m \mid \alpha \leq \beta\} \subset A$.

1. The maximum degree of each x_i, $1 \leq i \leq m$, and the overall maximum degree:

$$n_{A,i} := \max_{\alpha \in A} \alpha_i, \quad n_A := \max_{i=1,\dots,m} n_{A,i}.$$

2. The smallest hyper-rectangle containing A:

$$A^{\square} := \{0, \dots, n_{A,1}\} \times \dots \times \{0, \dots, n_{A,m}\}.$$

Framework

Downward Closed Polynomial Space

Definition (downward closed multi-index set)

(Finite) $A \subset \mathbb{N}_0^m$ downward closed, if $\forall \beta \in A \;:\; \beta \in A \;\Rightarrow \{\alpha \in \mathbb{N}^m \mid \alpha \leq \beta\} \subset A$.

1. The maximum degree of each x_i, $1 \leq i \leq m$, and the overall maximum degree:

$$n_{A,i} := \max_{\alpha \in A} \alpha_i, \quad n_A := \max_{i=1,\dots,m} n_{A,i}.$$

2. The smallest hyper-rectangle containing A:

$$A^{\square} := \{0,\dots,n_{A,1}\} \times \dots \times \{0,\dots,n_{A,m}\}.$$

Definition (downward closed polynomial space)

Let $m \in \mathbb{N}$ and a (finite) downward closed multi-index set $A \subset \mathbb{N}^m$. Then

$$\Pi_A := \operatorname{span}_{\mathbb{R}} \{x^\alpha \mid \alpha \in A\}, \quad N_A := \dim(\Pi_A) = |A|.$$

Framework

Downward Closed Polynomial Space

Definition (downward closed multi-index set)

(Finite) $A \subset \mathbb{N}_0^m$ downward closed, if $\forall \beta \in A \ : \ \beta \in A \ \Rightarrow \{\alpha \in \mathbb{N}^m \mid \alpha \leq \beta\} \subset A$.

1. The maximum degree of each x_i, $1 \leq i \leq m$, and the overall maximum degree:

$$n_{A,i} := \max_{\alpha \in A} \alpha_i, \quad n_A := \max_{i=1,\dots,m} n_{A,i}.$$

2. The smallest hyper-rectangle containing A:

$$A^{\square} := \{0, \dots, n_{A,1}\} \times \dots \times \{0, \dots, n_{A,m}\}.$$

Definition (downward closed polynomial space)

Let $m \in \mathbb{N}$ and a (finite) downward closed multi-index set $A \subset \mathbb{N}^m$. Then

$$\Pi_A := \mathrm{span}_{\mathbb{R}} \{x^\alpha \mid \alpha \in A\}, \quad N_A := \dim(\Pi_A) = |A|.$$

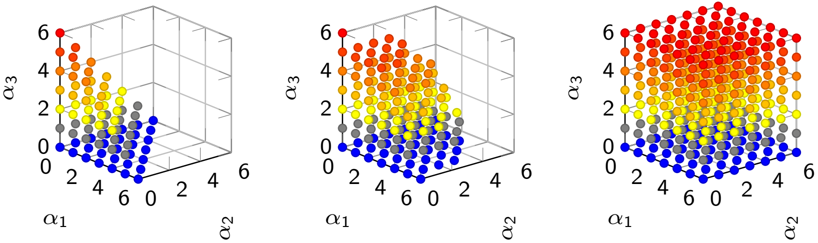
Definition (ℓ^p degree multi-index set)

$$A_{m,n,p} := \{\alpha \in \mathbb{N}_0^m \ : \ \|\alpha\|_p \leq n\}, \quad \Pi_{m,n,p} := \mathrm{span}_{\mathbb{R}}\{x^\alpha \mid \alpha \in A_{m,n,p}\}$$

Framework

ℓ^p Degree Polynomial Spaces (2)

Figure: Absolute, Euclidean and maximal degree multi-index-sets in 3d

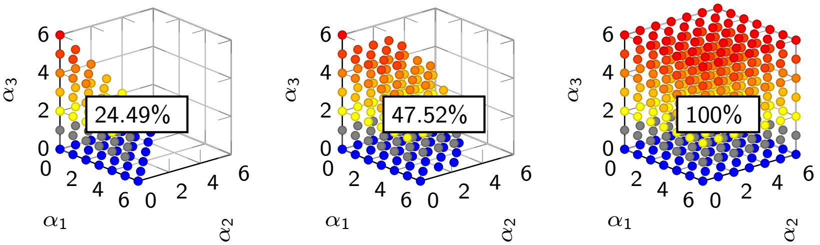


(a) $A_{3,6,1}$ (b) $A_{3,6,2}$ (c) $A_{3,6,\infty}$

Framework

ℓ^p Degree Polynomial Spaces (2)

Figure: Absolute, Euclidean and maximal degree multi-index-sets in 3d

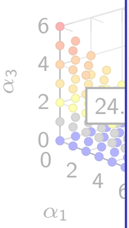


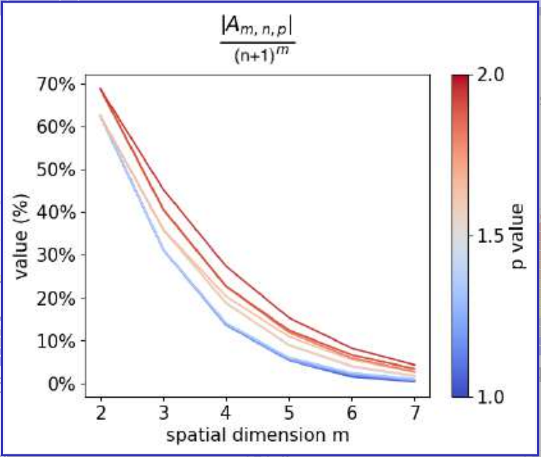
(a) $A_{3,6,1}$ (b) $A_{3,6,2}$ (c) $A_{3,6,\infty}$

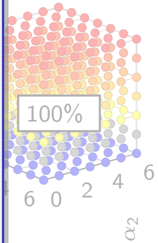
Framework

ℓ^p Degree Polynomial Spaces (2)

Figure: … s in 3d







(a) $A_{3,6,1}$ (b) $A_{3,6,2}$ (c) $A_{3,6,\infty}$

Framework

Computational Complexity

Basis	Interpolation	Differentiation
Fourier[1]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n)$
Chebyshev[2]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n^2/4)$
Newton[3]	$\mathcal{O}(n^2/2)$	$\mathcal{O}(n^2/2)$
Fast Downward–Closed Transform	$\|A\| \cdot m^2 \cdot \mathcal{O}(-"-/n_A)$	$\|A\| \cdot \mathcal{O}(-"-/n_A)$
Fast ℓ^p Transform	$\|A_{m,n,p}\| \cdot m \cdot \mathcal{O}(-"-/n)$	$\|A_{m,n,p}\| \cdot \mathcal{O}(-"-/n)$

[1] James W. Cooley and John W. Tukey, 1965

[2] Ahmed and Fisher, 1968

[3] Newton, 1736

Framework

Computational Complexity

Basis	Interpolation	Differentiation
Fourier[1]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n)$
Chebyshev[2]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n^2/4)$
Newton[3]	$\mathcal{O}(n^2/2)$	$\mathcal{O}(n^2/2)$
Fast Downward–Closed Transform	$\lvert A\rvert \cdot m^2 \cdot \mathcal{O}(\log_2(n_A))$	$\lvert A\rvert \cdot \mathcal{O}(1)$
Fast ℓ^p Transform	$\lvert A_{m,n,p}\rvert \cdot m \cdot \mathcal{O}(\log_2(n))$	$\lvert A_{m,n,p}\rvert \cdot \mathcal{O}(1)$

1 James W. Cooley and John W. Tukey, 1965

2 Ahmed and Fisher, 1968

3 Newton, 1736

Framework

Computational Complexity

Basis	Interpolation	Differentiation
Fourier[1]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n)$
Chebyshev[2]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n^2/4)$
Newton[3]	$\mathcal{O}(n^2/2)$	$\mathcal{O}(n^2/2)$
Fast Downward–Closed Transform	$\lvert A\rvert \cdot m^2 \cdot \mathcal{O}(\log_2(n_A))$	$\lvert A\rvert \cdot \mathcal{O}(n_A/4)$
Fast ℓ^p Transform	$\lvert A_{m,n,p}\rvert \cdot m \cdot \mathcal{O}(\log_2(n))$	$\lvert A_{m,n,p}\rvert \cdot \mathcal{O}(n/4)$

[1] James W. Cooley and John W. Tukey, 1965

[2] Ahmed and Fisher, 1968

[3] Newton, 1736

Framework

Computational Complexity

Basis	Interpolation	Differentiation
Fourier[1]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n)$
Chebyshev[2]	$\mathcal{O}(n \cdot \log_2(n))$	$\mathcal{O}(n^2/4)$
Newton[3]	$\mathcal{O}(n^2/2)$	$\mathcal{O}(n^2/2)$
Fast Downward–Closed Transform	$\lvert A\rvert \cdot m^2 \cdot \mathcal{O}(n_A/2)$	$\lvert A\rvert \cdot \mathcal{O}(n_A/2)$
Fast ℓ^p Transform	$\lvert A_{m,n,p}\rvert \cdot m \cdot \mathcal{O}(n/2)$	$\lvert A_{m,n,p}\rvert \cdot \mathcal{O}(n/2)$

[1] James W. Cooley and John W. Tukey, 1965

[2] Ahmed and Fisher, 1968

[3] Newton, 1736

Framework

Flattening

1. Arranges sequences of objects based on co-lexicographical rules.

Framework

Flattening

1. Arranges sequences of objects based on co-lexicographical rules.
2. Tensors are flattened into matrices according to co-lexicographic order.

Framework

Flattening

1. Arranges sequences of objects based on co-lexicographical rules.
2. Tensors are flattened into matrices according to co-lexicographic order.
3. Whenever an index $\alpha \in A$ appears as a subscript, it is understood to follow the co-lexicographic order.

Framework

Flattening

1. Arranges sequences of objects based on co-lexicographical rules.
2. Tensors are flattened into matrices according to co-lexicographic order.
3. Whenever an index $\alpha \in A$ appears as a subscript, it is understood to follow the co-lexicographic order.

Definition (Co-Lexicographic Order)

Let $m \in \mathbb{N}$, $A \subset \mathbb{N}^m$ finite and $\alpha, \beta \in \mathbb{N}^m$ we define

$$\alpha \leq_{\text{colex}} \beta \iff \alpha = \beta \text{ or } \alpha_i < \beta_i, \alpha_{i+1} = \beta_{i+1}, \ldots, \alpha_m = \beta_m \text{ for } 1 \leq i \leq m.$$

Framework

Flattening

1. Arranges sequences of objects based on co-lexicographical rules.
2. Tensors are flattened into matrices according to co-lexicographic order.
3. Whenever an index $\alpha \in A$ appears as a subscript, it is understood to follow the co-lexicographic order.

Definition (Co-Lexicographic Order)

Let $m \in \mathbb{N}$, $A \subset \mathbb{N}^m$ finite and $\alpha, \beta \in \mathbb{N}^m$ we define

$$\alpha \leq_{\text{colex}} \beta \iff \alpha = \beta \text{ or } \alpha_i < \beta_i, \alpha_{i+1} = \beta_{i+1}, \ldots, \alpha_m = \beta_m \text{ for } 1 \leq i \leq m.$$

Definition (Ordinal Position)

Let $m \in \mathbb{N}$, $A \subset \mathbb{N}^m$ finite and $\alpha \in \mathbb{N}^m$. The ordinal position of α in denoted by

$$\text{colex}_A : A \ni \alpha \mapsto |\{\beta \in A \mid \beta \leq_{\text{colex}} \alpha\}| \in \mathbb{N}.$$

Hence,

$$\text{colex}_A^{-1}(k), \quad 1 \leq k \leq N_A,$$

denotes the multi-index $\alpha \in A$ belonging to the k-th element of A w.r.t. $\leq_{\text{colex}}$.

Overview

Fast Full-Tensor Transform

Definition

Let $\{Q_\beta\}_{\beta\in A}$ be a basis of Π_A. The *backward* transform $\mathbf{B}_A$, the *forward* transform $\mathbf{F}_A$ and the *differentiation* matrix $\mathbf{D}_A$ are straightforwardly expressed through

$$\mathbf{F}_A := \mathbf{B}_A^{-1},\ \mathbf{B}_A := \big(Q_\beta(p_\alpha)\big)_{\alpha,\beta\in A},\ \mathbf{D}_{A,i} := \big({}^{\partial}/_{\partial x_i} Q_\beta(p_\alpha)\big)_{\alpha,\beta\in A} \in \mathbb{R}^{N_A\times N_A}.$$

Fast Full-Tensor Transform

Definition

Let $\{Q_\beta\}_{\beta\in A}$ be a basis of Π_A. The *backward* transform $\mathbf{B}_A$, the *forward* transform $\mathbf{F}_A$ and the *differentiation* matrix $\mathbf{D}_A$ are straightforwardly expressed through

$$\mathbf{F}_A := \mathbf{B}_A^{-1}, \; \mathbf{B}_A := \big(Q_\beta(p_\alpha)\big)_{\alpha,\beta\in A}, \; \mathbf{D}_{A,i} := \big({}^{\partial}\!/\!{}_{\partial x_i} Q_\beta(p_\alpha)\big)_{\alpha,\beta\in A} \in \mathbb{R}^{N_A\times N_A}.$$

Proposition

If $A = A^{\square}$, we can express $\mathbf{B}_A$, $\mathbf{F}_A$, and $\mathbf{D}_{A,1}$ as Kronecker products

$$\mathbf{F}_A = \bigotimes_{i=1}^m \mathbf{F}_{\{0,\dots,n_{i,A}\}}, \mathbf{B}_A = \bigotimes_{i=1}^m \mathbf{B}_{\{0,\dots,n_{i,A}\}}, \mathbf{D}_{A,1} = \mathbf{D}_{\{0,\dots,n_{1,A}\}} \bigotimes_{i=2}^m \mathbf{I}_{n_{i,A}+1} \in \mathbb{R}^{N_A\times N_A}$$

where $I_n \in \mathbb{R}^n$ denotes the identity matrix.

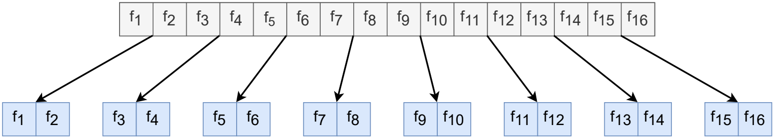
Fast Full-Tensor Transform

$m = 4, n = 1, p = \infty$

f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}

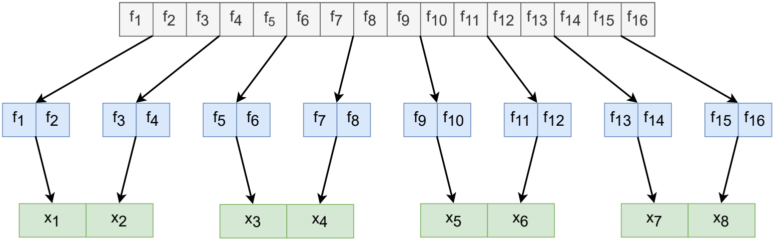
Fast Full-Tensor Transform

$m = 4, n = 1, p = \infty$



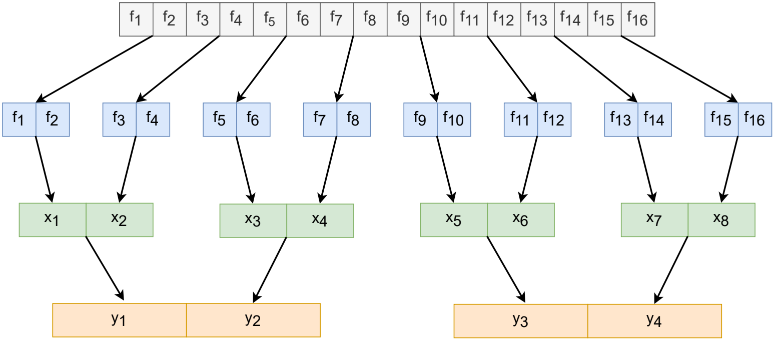
Fast Full-Tensor Transform

$m = 4, n = 1, p = \infty$



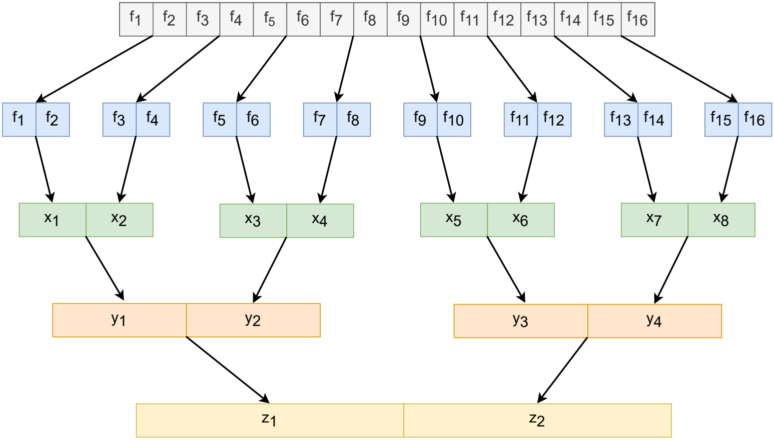
Fast Full-Tensor Transform

$m = 4, n = 1, p = \infty$



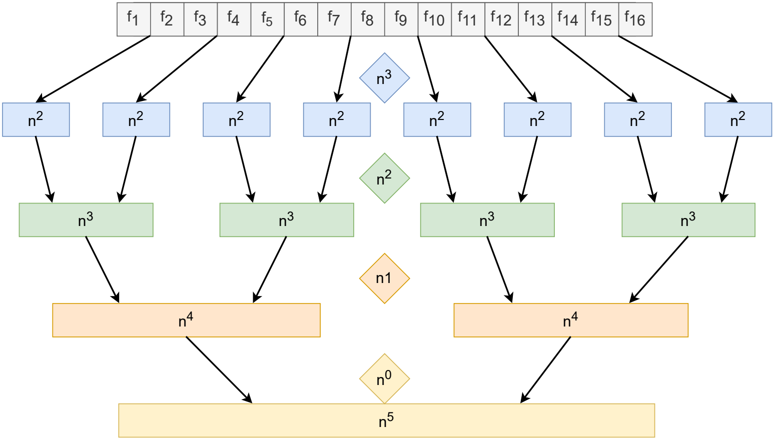
Fast Full-Tensor Transform

$m = 4, n = 1, p = \infty$



Fast Full-Tensor Transform

$m = 4, n = 1, p = \infty$



Fast Full-Tensor Transform

Theorem (Fast Full-Tensor Transform)

Given the square matrices $\mathbf{B}_i \in \mathbb{R}^{(n_{i,A}+1)\times(n_{i,A}+1)}$, $1 \leq i \leq m$, *and a vector* $\mathbf{v} = (v_1, v_2, \dots, v_{N_A})^\top \in \mathbb{R}^{N_A}$, *the matrix vector product*

$$(\mathbf{B}_1 \otimes \dots \otimes \mathbf{B}_m) \cdot \mathbf{v}$$

can be computed in $\mathcal{O}(N_A \cdot n_A \cdot m)$.

Fast Full-Tensor Transform

Theorem (Fast Full-Tensor Transform)

Given the square matrices $\mathbf{B}_i \in \mathbb{R}^{(n_{i,A}+1)\times(n_{i,A}+1)}$, $1 \le i \le m$, *and a vector* $\mathbf{v} = (v_1, v_2, \dots, v_{N_A})^\top \in \mathbb{R}^{N_A}$, *the matrix vector product*

$$(\mathbf{B}_1 \otimes \dots \otimes \mathbf{B}_m) \cdot \mathbf{v}$$

can be computed in $\mathcal{O}(N_A \cdot n_A \cdot m)$.

Definition (Generalized Matrix-Vector Product)

$\mathbf{B} = (\mathbf{B}_{i,j})_{i,j=1}^n \in \mathbb{R}^{(n+1)\times(n+1)}$, $\mathbf{w} = (w_i)_{i=1}^{N'} \in \mathbb{R}^{N'}$ and $N' \equiv 0 \mod (n+1)$. Next, partition $\mathbf{w}$ into $n+1$ chunks of size $s := N'/(n+1)$, compute

$$\mathbf{u}_i := \sum_{j=1}^{n+1} \mathbf{B}_{i,j}\mathbf{w}_{(j-1)\cdot s+1\,:\,j\cdot s} \in \mathbb{R}^s, \quad 1 \le i \le n+1.$$

Consequently, we define $\#$ by flattening

$$\mathbf{B}\#\mathbf{v} := (\mathbf{u}_1^\top, \dots, \mathbf{u}_{n+1}^\top)^\top \in \mathbb{R}^{N'}.$$

Note that $\# \in \Theta(n^2 \cdot s)$.

Fast Full-Tensor Transform

1. $H_{m,i} := \mathbb{N}_0^i \times \{0\}^{m-i},\ H_{m,i}^{\perp} = \{0\}^i \times \mathbb{N}_0^{m-i}$
2. $N_{A,i} := |A \cap H_{m,i}|,\ N_{A,i}^{\perp} := |A \cap H_{m,i}^{\perp}|$

Fast Full-Tensor Transform

1. $H_{m,i} := \mathbb{N}_0^i \times \{0\}^{m-i},\ H_{m,i}^{\perp} = \{0\}^i \times \mathbb{N}_0^{m-i}$
2. $N_{A,i} := |A \cap H_{m,i}|,\ N_{A,i}^{\perp} := |A \cap H_{m,i}^{\perp}|$

Scheme

We further define a double sequence of vectors $\{\mathbf{w}_i^j\}_{i,j}$, which allows to compute the matrix-vector product in (7) as

$$\mathbf{w} = \mathbf{B}_m \# \mathbf{w}_1^m.$$

Hereby, the double sequence $\{\mathbf{w}_i^j\}_{i,j}$, is computed by the iterative scheme:

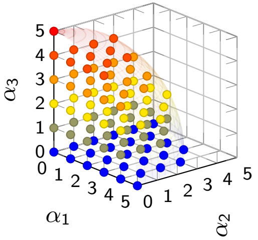
$$\begin{aligned}
\mathbf{w}_i^1 &:= v_{(i-1)\cdot(n_{A,1}+1)+1\ :\ i\cdot(n_{A,1}+1)}, 1 \leq i \leq N_{A,1}^{\perp}, \\
\mathbf{w}_i^{j+1} &:= \left(\mathbf{B}_j \# \mathbf{w}^j_{(i-1)\cdot(n_{A,j+1}+1)+k}\right)^{\top}_{1\leq k\leq n_{A,j+1}+1}, 1 \leq i \leq N_{A,j+1}^{\perp}, 1 \leq j \leq m-1
\end{aligned}$$

Overview

Fast Downward–Closed Transform

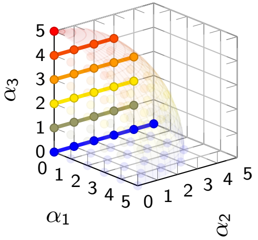
Tube Projections

Figure: Illustration of the second tube projection of $A_{3,5,2}$



(a) $A_{3,5,2}$

$$\stackrel{\mathcal{T}_{3,2}}{\longmapsto} \begin{pmatrix} 6 \\ 5 \\ 5 \\ 5 \\ 4 \\ 1 \end{pmatrix} \widehat{=}$$



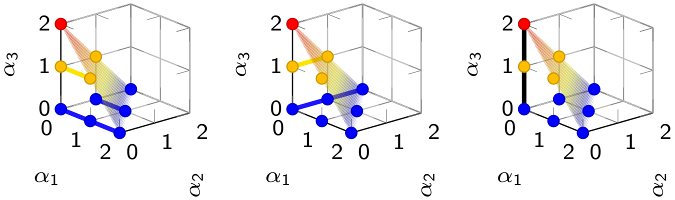
(b) $\mathcal{T}_{3,2}(A_{3,5,2})$

Fast Downward–Closed Transform

Tube Projections

$$\blacksquare = (0,0,0)+H_{3,2},\ \blacksquare = (0,0,1)+H_{3,2},\ \blacksquare = (0,0,2)+H_{3,2},\ \blacksquare = (0,0,0)+H_{3,3}.$$

Figure: Illustration of tube projections of $A_{3,2,1}$



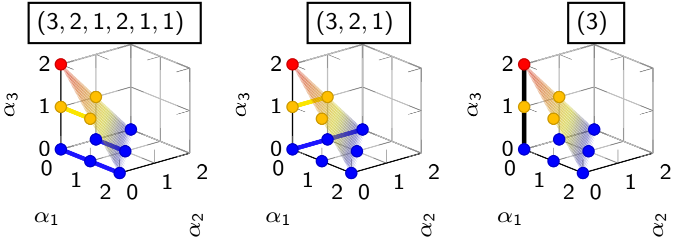
(a) $\mathcal{T}_{3,1}(A_{3,2,1})$ (b) $\mathcal{T}_{3,2}(A_{3,2,1})$ (c) $\mathcal{T}_{3,3}(A_{3,2,1})$

Fast Downward–Closed Transform

Tube Projections

$$\color{blue}\blacksquare \color{black} = (0,0,0)+H_{3,2},\ \color{yellow}\blacksquare \color{black} = (0,0,1)+H_{3,2},\ \color{red}\blacksquare \color{black} = (0,0,2)+H_{3,2},\ \blacksquare = (0,0,0)+H_{3,3}.$$

Figure: Illustration of tube projections of $A_{3,2,1}$



(a) $\mathcal{T}_{3,1}(A_{3,2,1})$ (b) $\mathcal{T}_{3,2}(A_{3,2,1})$ (c) $\mathcal{T}_{3,3}(A_{3,2,1})$

Fast Downward–Closed Transform

Proposition($A_{m,n,p}$–Construction)

Let $m, n \in \mathbb{N}$ and $p \in [0, \infty]$. To construct $A_{m,n,p}$, it takes

$$(1 + \kappa_{m,n,p}) \cdot |A_{m,n,p}| := |A_{m,n,p}| + |A_{m-1,n,p}| + \ldots + |A_{1,n,p}|.$$

Especially

$$\kappa_{m,n,0} \in \Theta(m), \quad \kappa_{m,n,1} \in \Theta(n/m), \quad \kappa_{m,n,\infty} \in \Theta(1).$$

Fast Downward–Closed Transform

Proposition($A_{m,n,p}$–Construction)

Let $m, n \in \mathbb{N}$ and $p \in [0, \infty]$. To construct $A_{m,n,p}$, it takes

$$(1 + \kappa_{m,n,p}) \cdot |A_{m,n,p}| := |A_{m,n,p}| + |A_{m-1,n,p}| + \ldots + |A_{1,n,p}|.$$

Especially

$$\kappa_{m,n,0} \in \Theta(m), \quad \kappa_{m,n,1} \in \Theta(n/m), \quad \kappa_{m,n,\infty} \in \Theta(1).$$

Proposition(A–Construction)

We define the *carry–overhead–factor* as

$$\kappa_A := N_A^{-1} \sum_{i=2}^{m} \left|\mathcal{T}_{m,i}(A)\right| \in [0, m-1].$$

To construct $\mathcal{T}_m(A)$, it requires $(1 + \kappa_A) \cdot N_A$ steps.

Fast Downward–Closed Transform

Proposition($A_{m,n,p}$–Construction)

Let $m, n \in \mathbb{N}$ and $p \in [0, \infty]$. To construct $A_{m,n,p}$, it takes

$$(1 + \kappa_{m,n,p}) \cdot |A_{m,n,p}| := |A_{m,n,p}| + |A_{m-1,n,p}| + \ldots + |A_{1,n,p}|.$$

Especially

$$\kappa_{m,n,0} \in \Theta(m), \quad \kappa_{m,n,1} \in \Theta(n/m), \quad \kappa_{m,n,\infty} \in \Theta(1).$$

Proposition(A–Construction)

We define the *carry–overhead–factor* as

$$\kappa_A := N_A^{-1} \sum_{i=2}^{m} \left|\mathcal{T}_{m,i}(A)\right| \in [0, m-1].$$

To construct $\mathcal{T}_m(A)$, it requires $(1 + \kappa_A) \cdot N_A$ steps.

Proposition($\kappa_{m,n,p}$–Bound)

For $m \in \mathbb{N}$, $n > 4 \cdot (m+1)$ and $p \in (1, \infty)$

$$\kappa_{m,n,p} \leq \sqrt{e} \approx 1.65.$$

Fast Downward–Closed Transform

Proposition

For any non-empty finite downward–closed sets $A \subset A' \subset \mathbb{N}_0^m$, the map Φ_m is well-defined through

$$(T, T') := (\mathcal{T}_m(A), \mathcal{T}_m(A')) \overset{\Phi_m}{\mapsto} \varphi_{A,A'} := \mathrm{colex}_{A'} \circ \mathrm{colex}_A^{-1}.$$

Further, values of the map Φ_m can be computed in $\mathcal{O}(N_A \cdot (1 + \kappa_A))$.

Fast Downward–Closed Transform

Proposition

For any non-empty finite downward–closed sets $A \subset A' \subset \mathbb{N}_0^m$, the map Φ_m is well-defined through

$$(T, T') := (\mathcal{T}_m(A), \mathcal{T}_m(A')) \overset{\Phi_m}{\mapsto} \varphi_{A,A'} := \mathrm{colex}_{A'} \circ \mathrm{colex}_A^{-1}.$$

Further, values of the map Φ_m can be computed in $\mathcal{O}(N_A \cdot (1 + \kappa_A))$.

Example

For instance $\varphi_{A,A'}$ with $A = A_{3,2,1}$ and $A' = A_{3,2,\infty}$ is given by:

$$\begin{gathered}(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}) \\ \downarrow \\ (c_1, c_2, c_3, c_4, c_5, 0, c_6, 0, 0, c_7, c_8, 0, c_9, 0, 0, 0, 0, 0, c_{10}, 0, 0, 0, 0, 0, 0, 0, 0)\end{gathered}$$

Fast Downward–Closed Transform

Proposition

For any non-empty finite downward–closed sets $A \subset A' \subset \mathbb{N}_0^m$, the map Φ_m is well-defined through

$$(T, T') := (\mathcal{T}_m(A), \mathcal{T}_m(A')) \overset{\Phi_m}{\mapsto} \varphi_{A,A'} := \mathrm{colex}_{A'} \circ \mathrm{colex}_A^{-1}.$$

Further, values of the map Φ_m can be computed in $\mathcal{O}(N_A \cdot (1 + \kappa_A))$.

Example

For instance $\varphi_{A,A'}$ with $A = A_{3,2,1}$ and $A' = A_{3,2,\infty}$ is given by:

$$(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10})$$
$$\downarrow$$
$$(c_1, c_2, c_3, c_4, c_5, 0, c_6, 0, 0, c_7, c_8, 0, c_9, 0, 0, 0, 0, 0, c_{10}, 0, 0, 0, 0, 0, 0, 0, 0)$$

Proposition

The backward transformation matrix $\mathbf{B}_A$, its inverse $\mathbf{F}_A$, and the differentiation matrix $\mathbf{D}_A$ can be expressed as

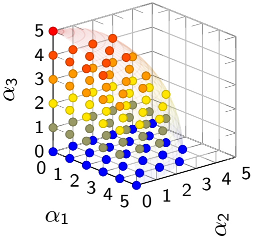
$$\mathbf{F}_A = \varphi_{A,A^\square}\ \mathbf{F}_{A^\square}, \quad \mathbf{B}_A = \varphi_{A,A^\square}\ \mathbf{B}_{A^\square}, \quad \mathbf{D}_{A,i} = \varphi_{A,A^\square}\ \mathbf{D}_{A^\square,i},$$

where $\varphi_{A,A^\square}$ is interpreted as a matrix.

Fast Downward–Closed Transform

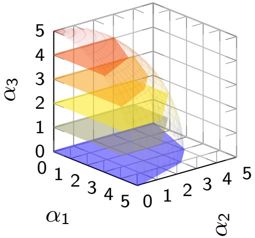
Volume Projections

Figure: Illustration of the second volume projection of $A_{3,5,2}$



(a) $A_{3,5,2}$

$$\xrightarrow{\mathcal{V}_{3,2}} \begin{pmatrix} 26 \\ 22 \\ 22 \\ 17 \\ 11 \\ 1 \end{pmatrix} \;\widehat{=}$$



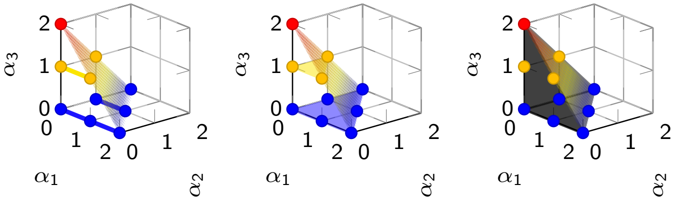
(b) $\mathcal{T}_{3,2}(A_{3,5,2})$

Fast Downward–Closed Transform

Volume Projections

■ $= (0,0,0) + H_{3,2}$, ■ $= (0,0,1) + H_{3,2}$, ■ $= (0,0,2) + H_{3,2}$, ■ $= (0,0,0) + H_{3,3}$.

Figure: Illustration of volume projections of $A_{3,2,1}$



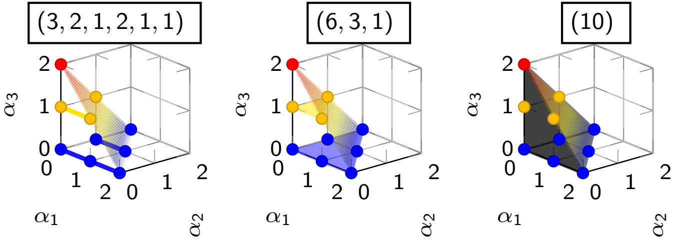
(a) $\mathcal{V}_{3,1}(A_{3,2,1})$ (b) $\mathcal{V}_{3,2}(A_{3,2,1})$ (c) $\mathcal{V}_{3,3}(A_{3,2,1})$

Fast Downward–Closed Transform

Volume Projections

■ $= (0,0,0)+H_{3,2}$, ■ $= (0,0,1)+H_{3,2}$, ■ $= (0,0,2)+H_{3,2}$, ■ $= (0,0,0)+H_{3,3}$.

Figure: Illustration of volume projections of $A_{3,2,1}$



(a) $\mathcal{V}_{3,1}(A_{3,2,1})$ (b) $\mathcal{V}_{3,2}(A_{3,2,1})$ (c) $\mathcal{V}_{3,3}(A_{3,2,1})$

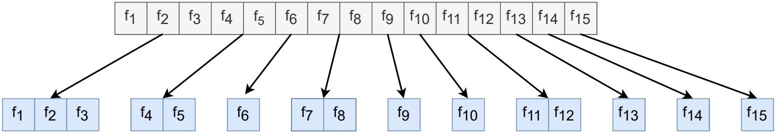
Fast Downward–Closed Transform

$m = 4, n = 2, p = 1$

f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}

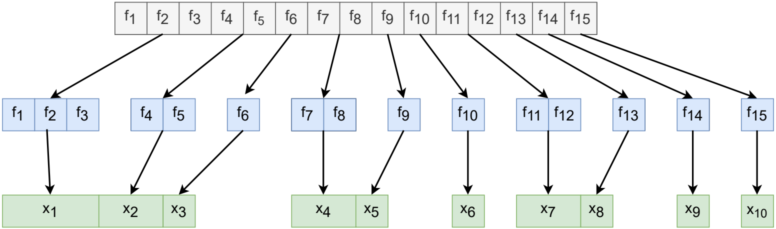
Fast Downward–Closed Transform

$m = 4$, $n = 2$, $p = 1$



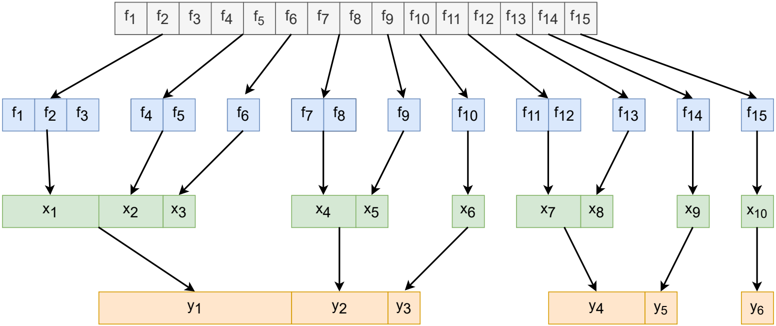
Fast Downward–Closed Transform

$m = 4, n = 2, p = 1$



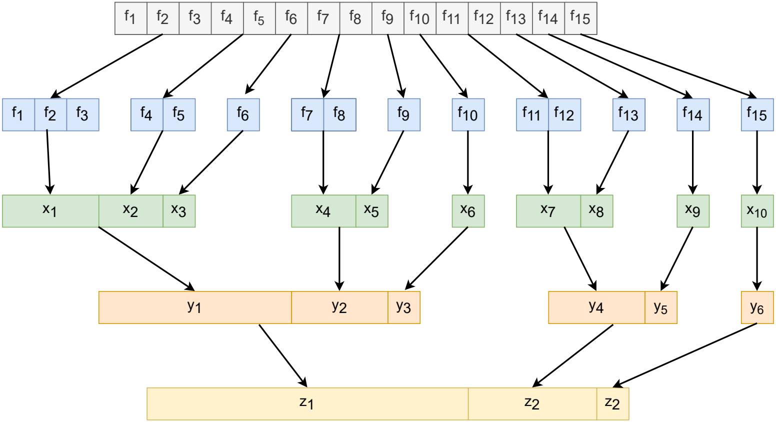
Fast Downward–Closed Transform

$m = 4, n = 2, p = 1$



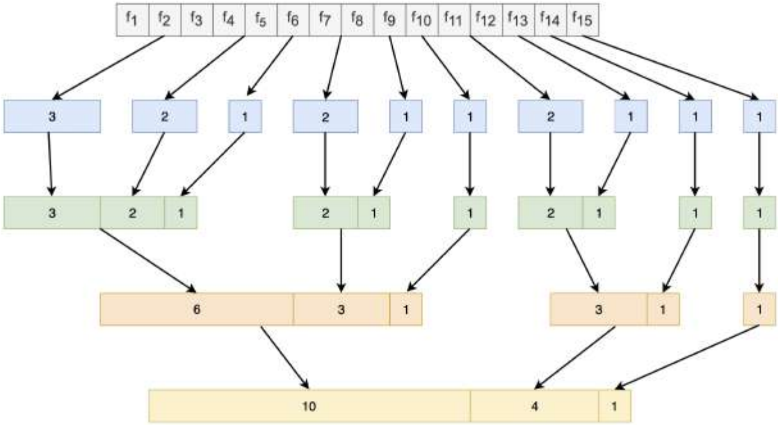
Fast Downward–Closed Transform

$m = 4, n = 2, p = 1$



Fast Downward–Closed Transform

$m = 4, n = 2, p = 1$



Fast Downward–Closed Transform

$m = 4, n = 2, p = 1$

Fast ℓ^p Transformation

Given $\{(p_\beta, f_\beta)\}_{\beta \in A_{m,n,p}} \subset \mathbb{R}^{m+1}$, the parameters $\{c_\alpha\}_{\alpha \in A_{m,n,p}} \subset \mathbb{R}$ such that

$$\forall \beta \in A_{m,n,p} \ : \sum_{\alpha \in A_{m,n,p}} c_\alpha \cdot Q_\alpha(p_\beta) = f_\beta,$$

can be computed in

$$\mathcal{O}(|A_{m,n,p}| \cdot m \cdot n \cdot \kappa_{m,n,p}) \subset \mathcal{O}(|A_{m,n,p}| \cdot m \cdot n),$$

for $n > 4 \cdot (m+1)$.

Overview

Numerical Experiments

Interpolation of Runges Function

Definition (modified Runge function)

$$f : [-1,1]^m \to, \; x \mapsto \frac{1}{1 + 9 \cdot \|x\|^2}$$

Numerical Experiments

Interpolation of Runges Function

Definition (modified Runge function)

$$f : [-1, 1]^m \to, \; x \mapsto \frac{1}{1 + 9 \cdot \|x\|^2}$$

We benchmark against

- ApproxFun[1] in 2D
- ChebFun[2] in 3D

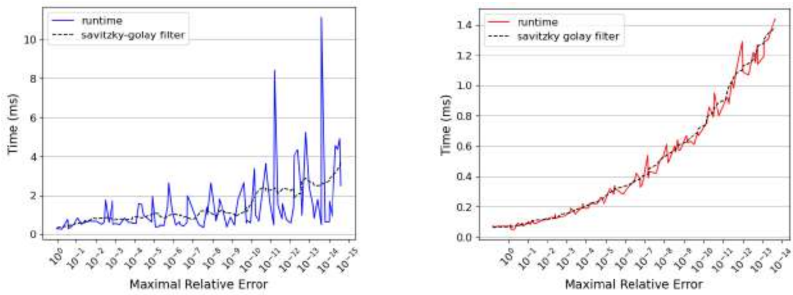
[1] Sheehan Olver and Alex Townsend, 2014

[2] Trefethen, Lloyd N. and others, 2023

Numerical Experiments

Interpolation of Runges Function

Figure: Benchmark Runge function 2d

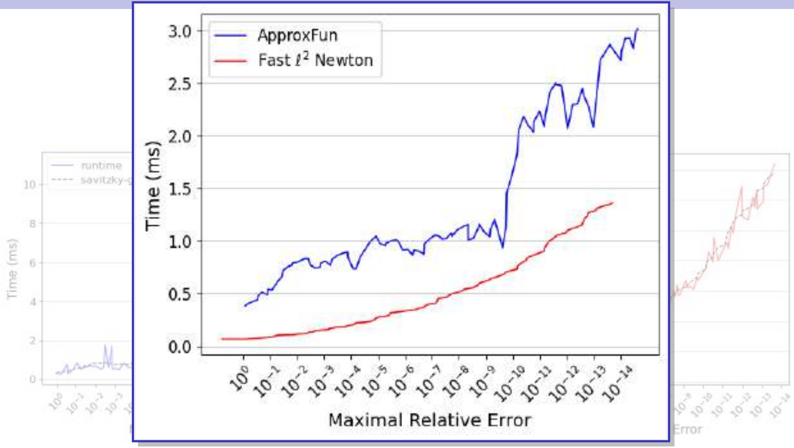


(a) ApproxFun package

(b) Fast ℓ^2 Newton transformation

Numerical Experiments

Interpolation of Runges Function

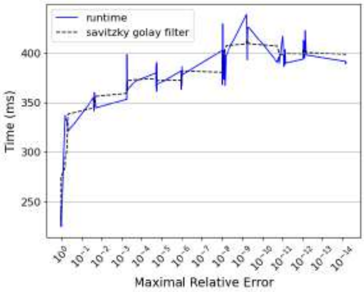


(a) ApproxFun package (b) Fast ℓ^2 Newton transformation

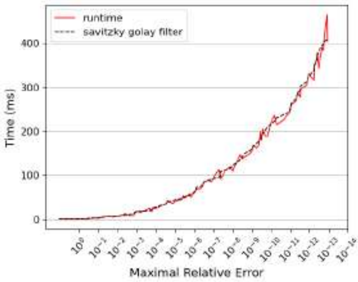
Numerical Experiments

Interpolation of Runges Function

Figure: Benchmark Runge function 3d



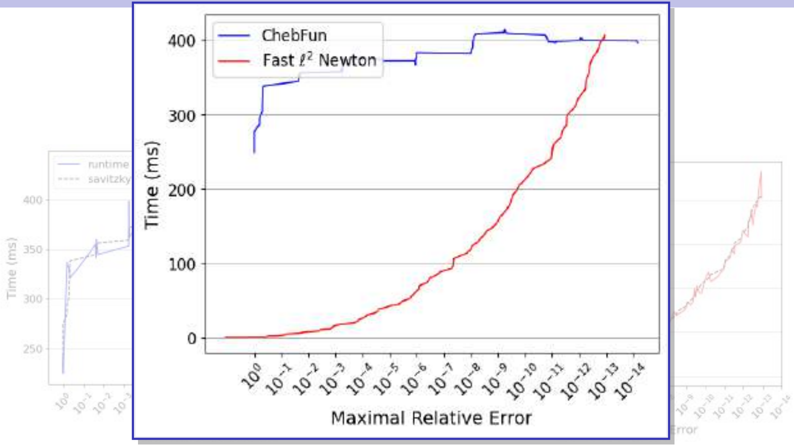
(a) ChebFun package



(b) Fast ℓ^2 Newton transformation

Numerical Experiments

Interpolation of Runges Function



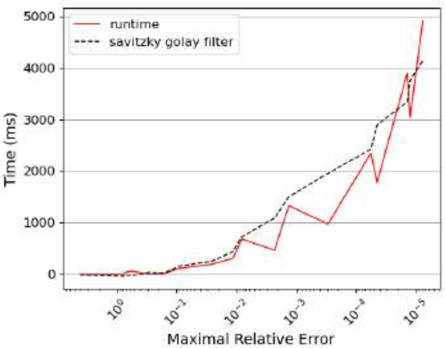
(a) ChebFun package

(b) Fast ℓ^2 Newton transformation

Numerical Experiments

Interpolation of Runges Function

Figure: Benchmark Runge function 4d



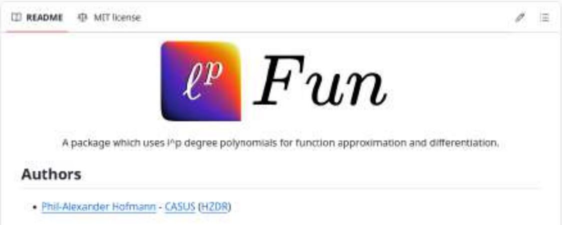
(a) Fast ℓ^2 Newton transformation

What is next?

- Soon: PyPI Release of *lpFun* providing Fast ℓ^p Transform for *Newton*, *Chebyshev* and *Fourier* basis including spectral ℓ^p differentiation

What is next?

- Soon: PyPI Release of *lpFun* providing Fast ℓ^p Transform for *Newton*, *Chebyshev* and *Fourier* basis including spectral ℓ^p differentiation



- Replace/Extent Numba by C++ (2x - 10x Speedup expected)

What is next?

- Soon: PyPI Release of *lpFun* providing Fast ℓ^p Transform for *Newton*, *Chebyshev* and *Fourier* basis including spectral ℓ^p differentiation

- Replace/Extent Numba by C++ (2x - 10x Speedup expected)
- Differential geometry : Spectral ℓ^p methods based on the hierarchical Poincaré-Steklov method – Gentian Zavalani (CASUS) & Dan Fortunato (CCM, Flatiron Institute, NYC)

What is next?

- Soon: PyPI Release of *lpFun* providing Fast ℓ^p Transform for *Newton*, *Chebyshev* and *Fourier* basis including spectral ℓ^p differentiation

- Replace/Extent Numba by C++ (2x - 10x Speedup expected)
- Differential geometry : Spectral ℓ^p methods based on the hierarchical Poincaré-Steklov method – Gentian Zavalani (CASUS) & Dan Fortunato (CCM, Flatiron Institute, NYC)
- Gierer-Meinhardt Reaction Diffusion Model – Prof. Grzegorz Plebanek

What is next?

- Soon: PyPI Release of *lpFun* providing Fast ℓ^p Transform for *Newton*, *Chebyshev* and *Fourier* basis including spectral ℓ^p differentiation

- Replace/Extent Numba by C++ (2x - 10x Speedup expected)
- Differential geometry : Spectral ℓ^p methods based on the hierarchical Poincaré-Steklov method – Gentian Zavalani (CASUS) & Dan Fortunato (CCM, Flatiron Institute, NYC)
- Gierer-Meinhardt Reaction Diffusion Model – Prof. Grzegorz Plebanek
- 6-7d Fokker-Planck / Kohn-Sham equation – Dr. Petr Cagaš (CASUS)

Thank you for your attention!