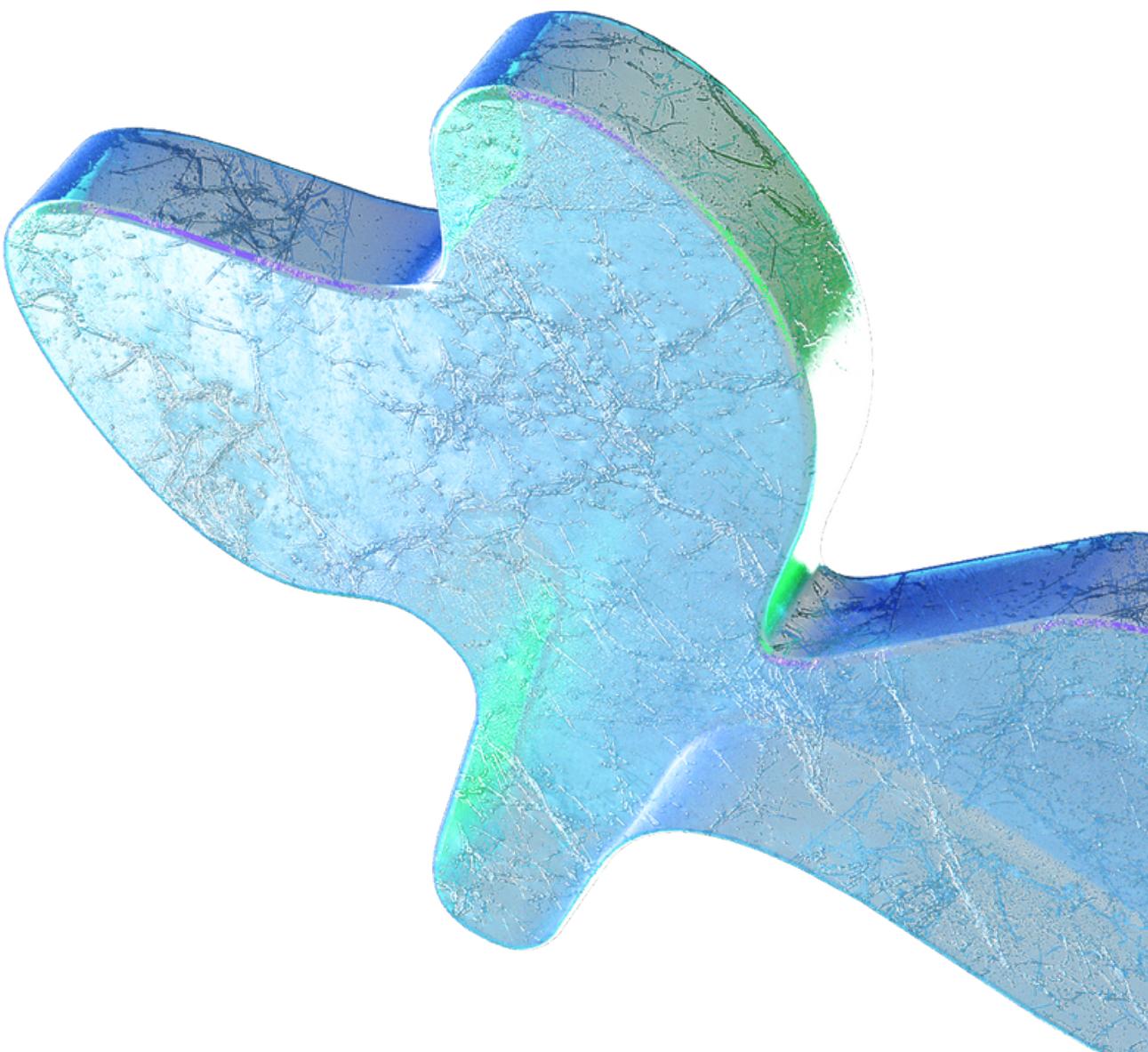




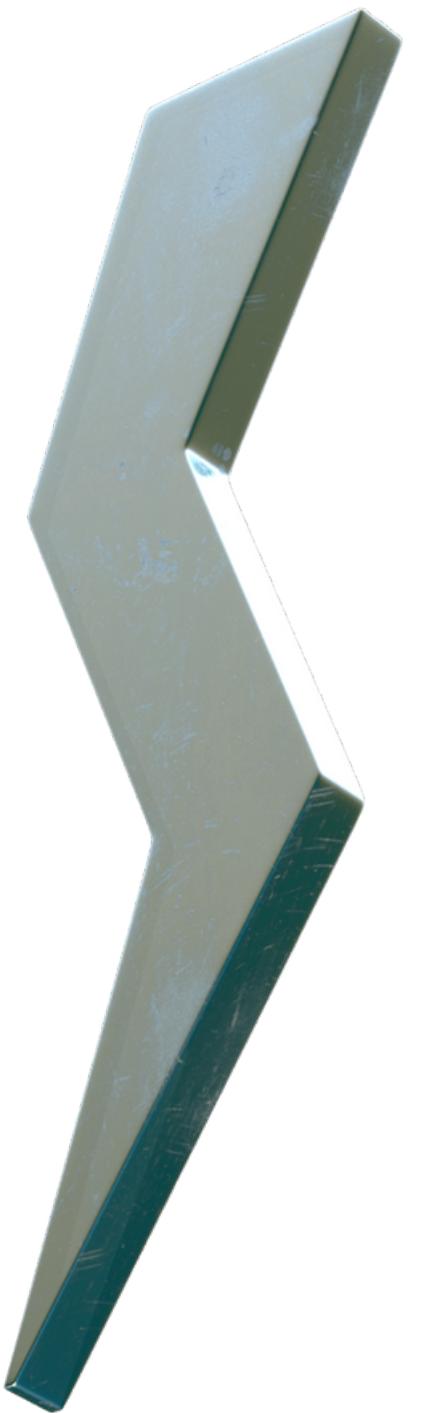
# Z - Kurven

Team 223 – SS23

Pascal Gaertner – Tushar Khandelwal – Shaurya Sharma



# Inhalt



Anwendungen

---

Z Kurve: Einführung

---

Optimierungen

---

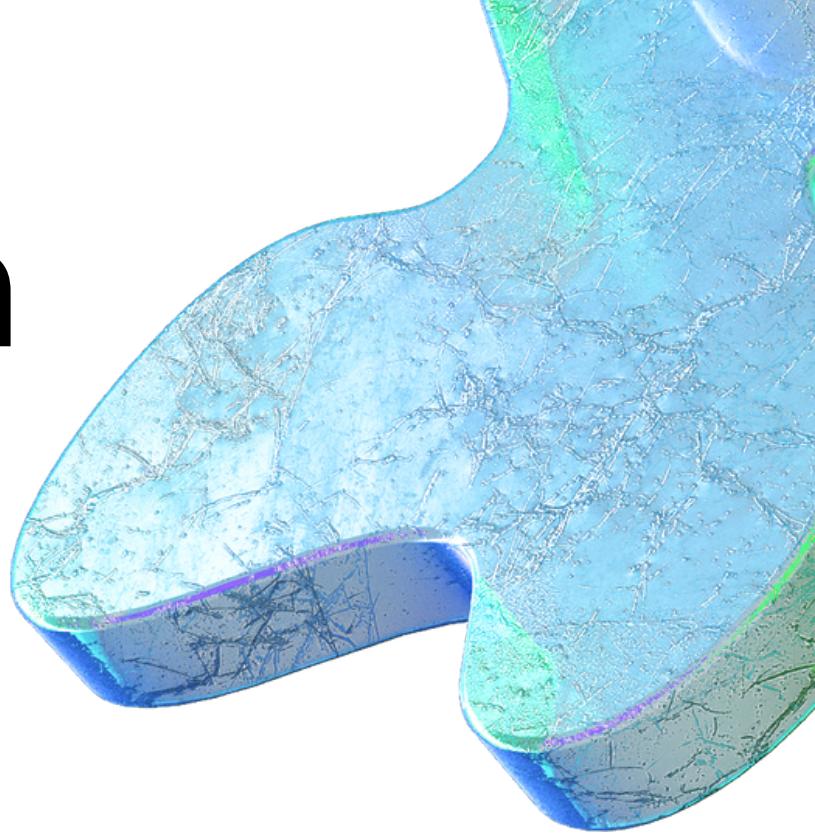
Performanzanalyse

---

Quellen

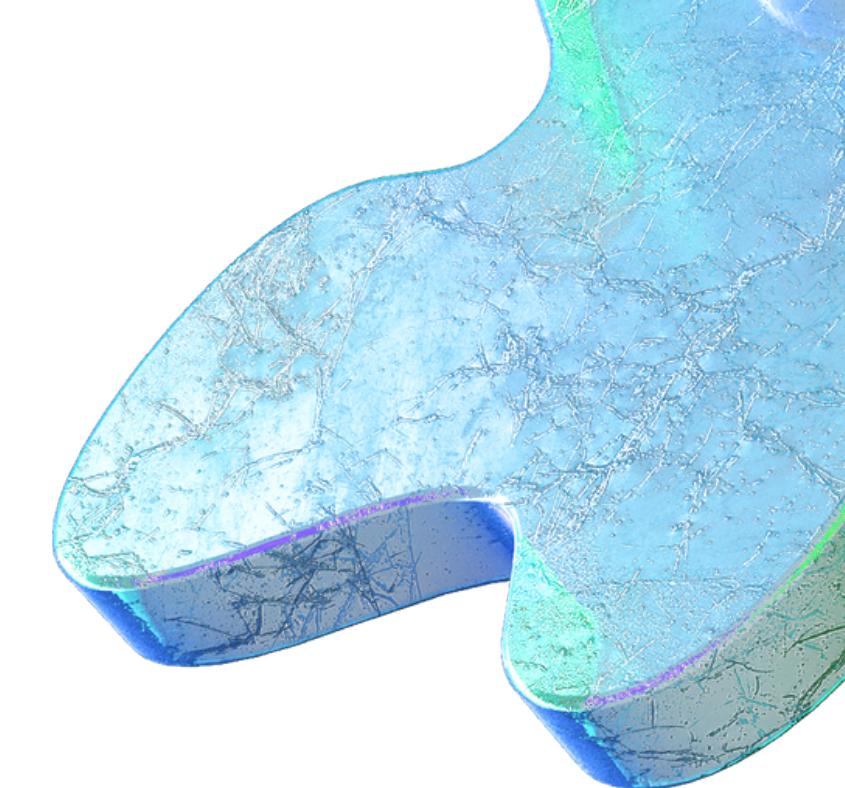
---

# Z-Kurve und ihre Anwendungen

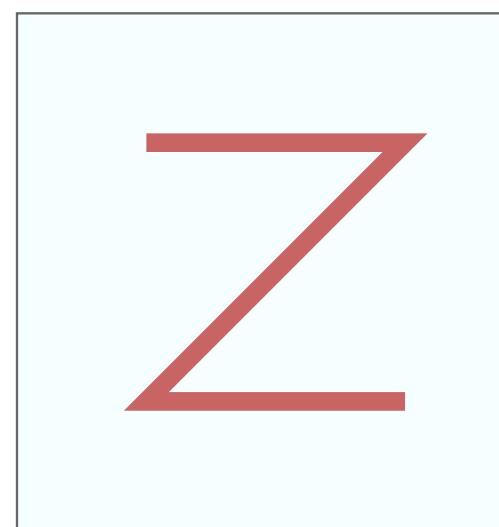


- Abbildung mehrdimensionaler Daten in eine eindimensionale Sequenz
- Anwendung in Computergrafik, Datenkompression, Datenbanken und räumlichen Indexstrukturen
- Datenvisualisierung mit der Z-Kurve
- Verbesserung der Effizienz räumlicher Abfragen in Datenbanken

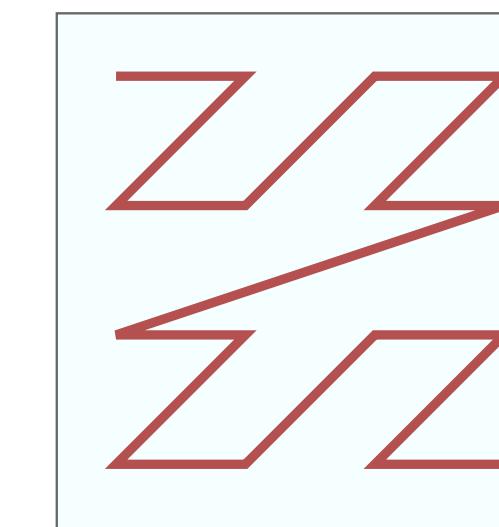
# Z-Kurve : Erklärung



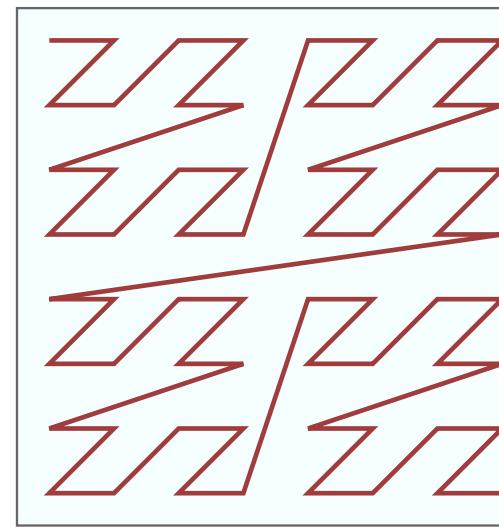
Grad 4 der Z-Kurve



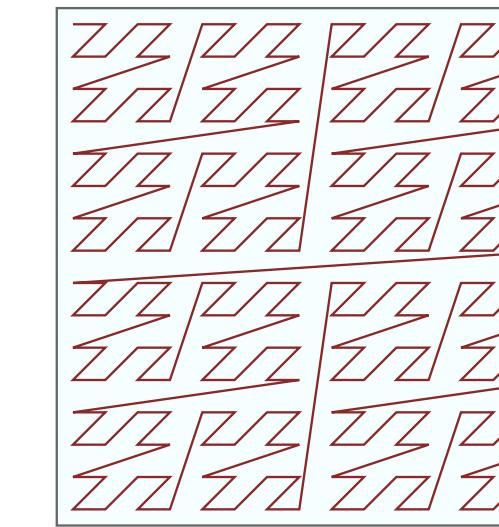
Grad 1



Grad 2



Grad 3

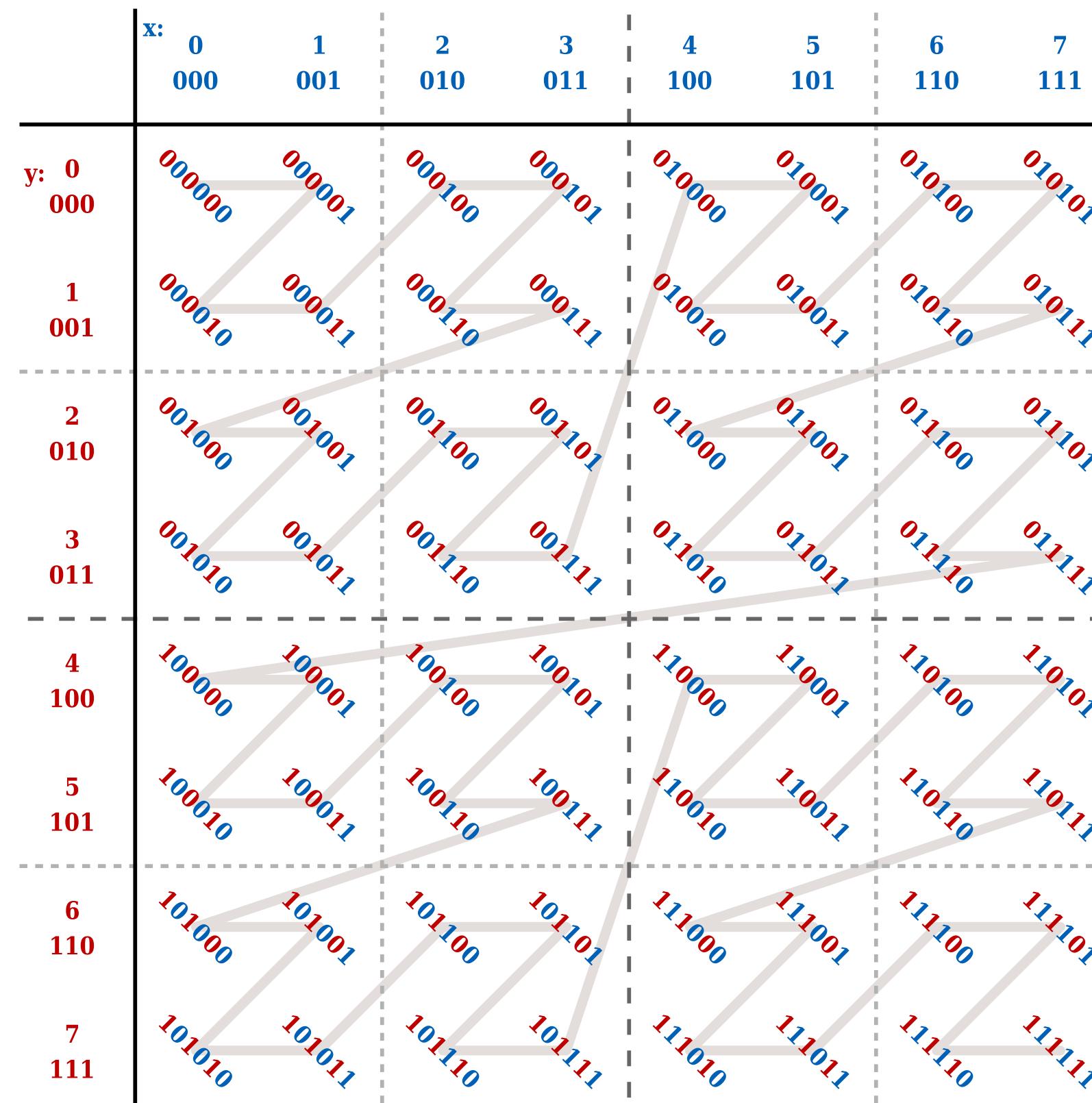


Grad 4

Z curve Matrix

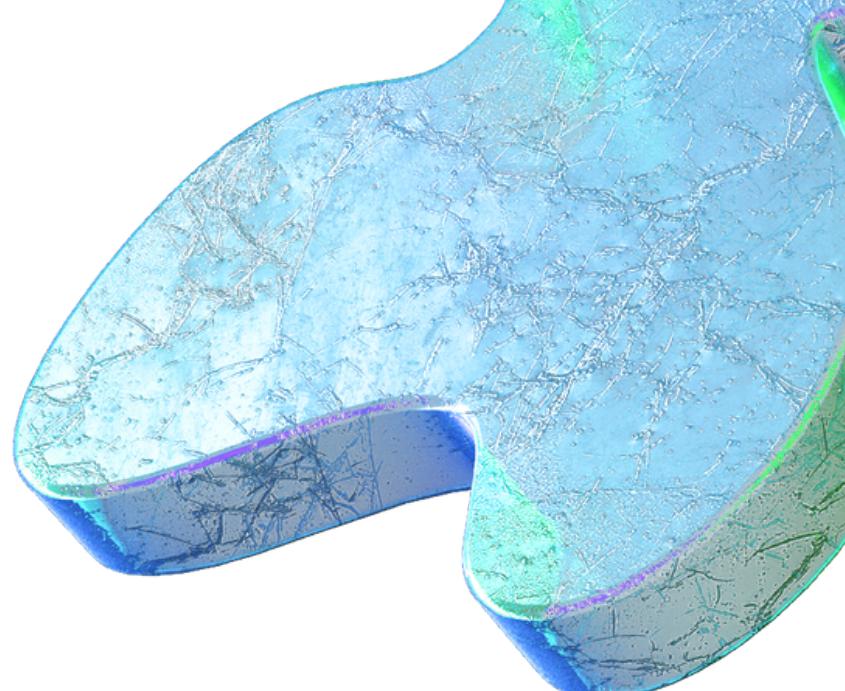
+	0	1	4	5	16	17	20	21
0	0	1	4	5	16	17	20	21
2	2	3	6	7	18	19	22	23
8	8	9	12	13	24	25	28	29
10	10	11	14	15	26	27	30	31
32	32	33	36	37	48	49	52	53
34	34	35	38	39	50	51	54	55
40	40	41	44	45	56	57	60	61
42	42	43	46	47	58	59	62	63

# Z-Kurve : Interleaving



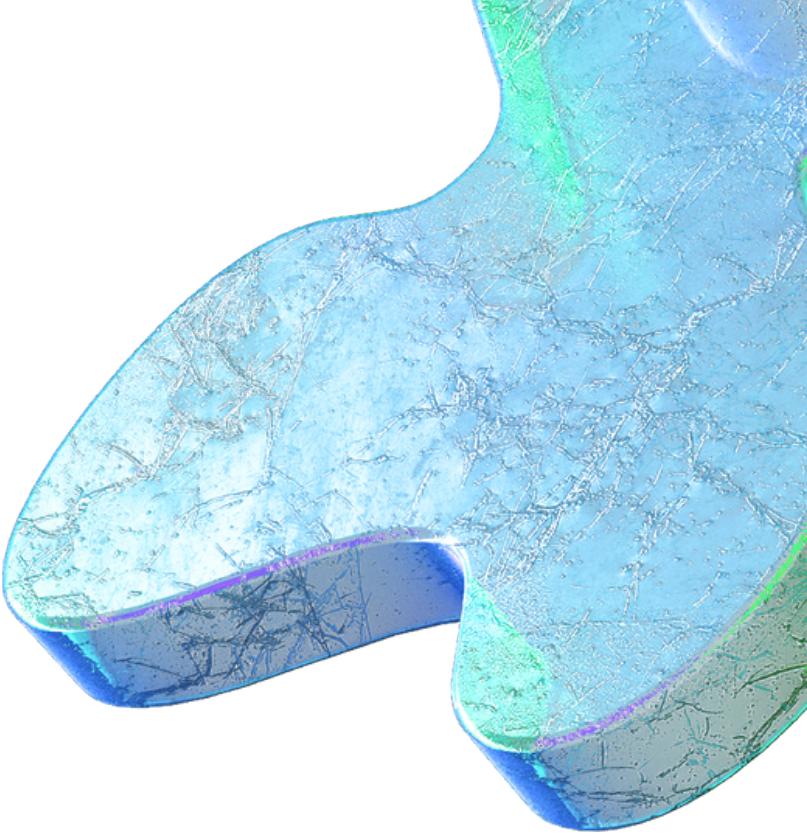
5

$$\begin{aligned} x &= 0b10011 = 19 \\ z &= 0b100110101111 = 47 \\ y &= 0b101111 = 2479 \end{aligned}$$



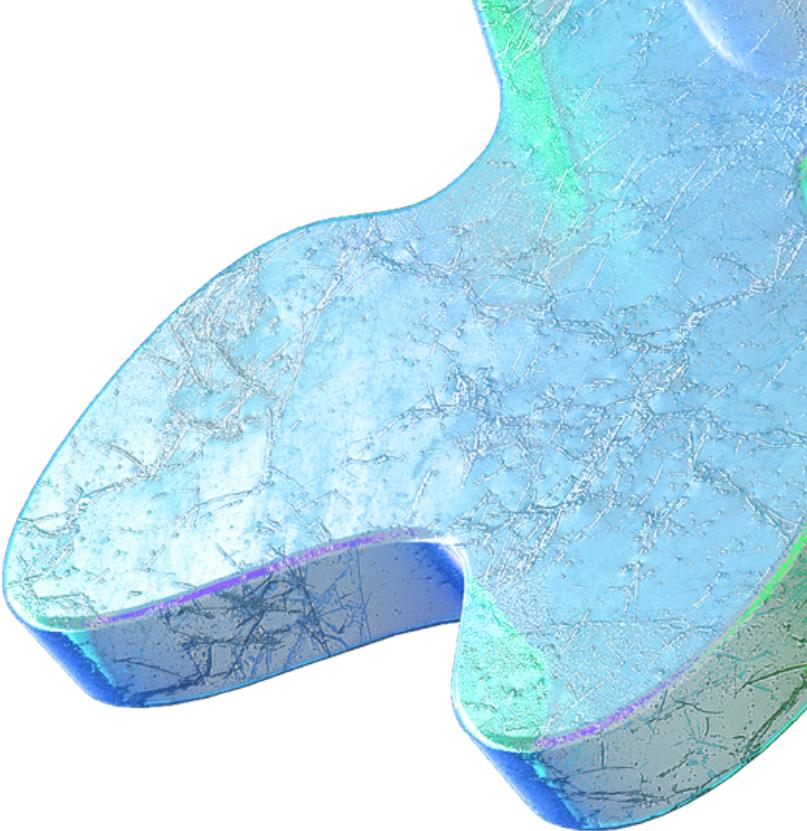
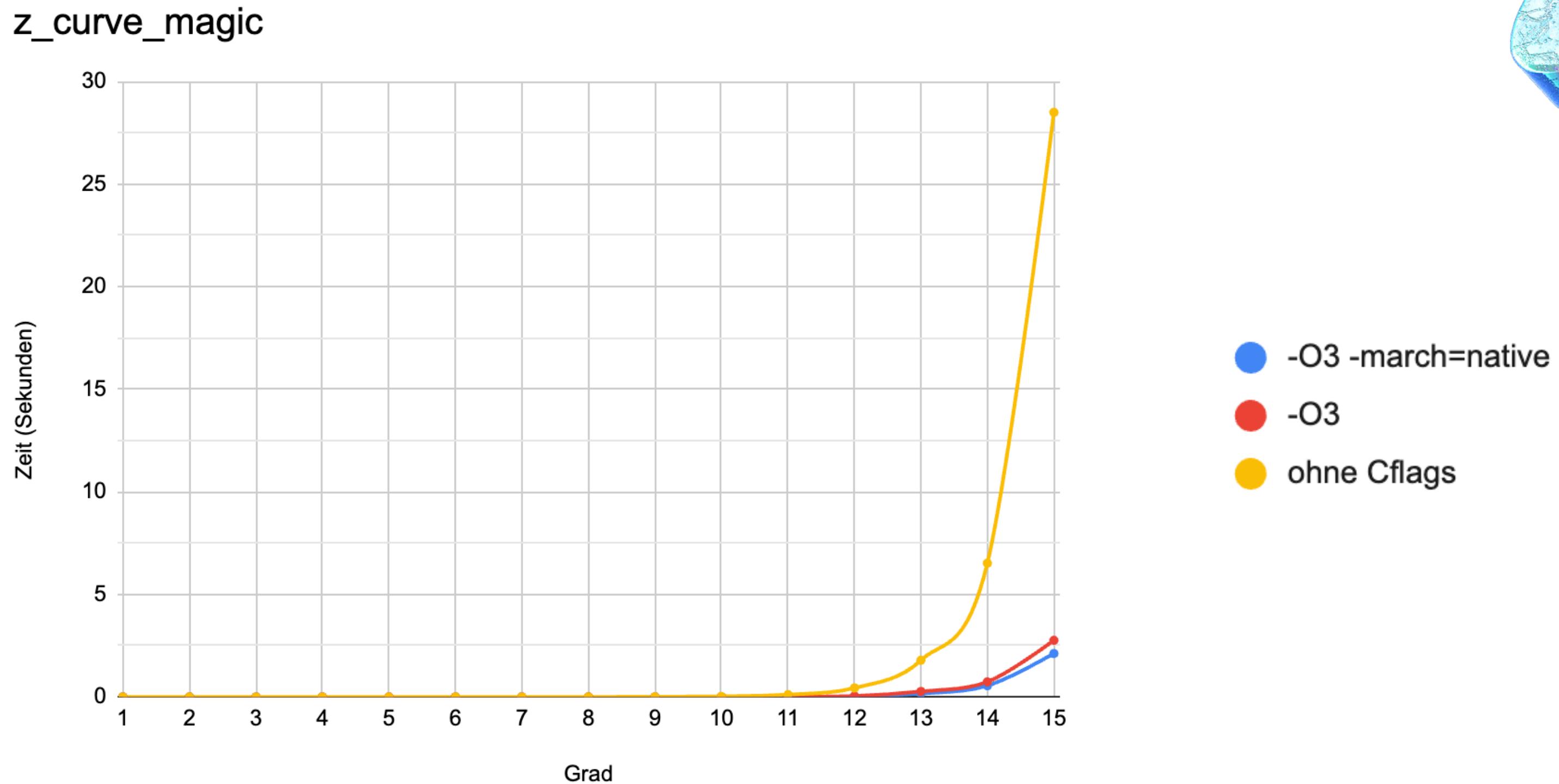
# OPTIMIERUNGEN

# **Z-Kurve Methode und Implementierungen**



- `z_curve_magic`
- `z_curve_magic_simd`
- `z_curve_recursive`
- `z_curve_iterative_simd`
- `z_curve_iterative`

# Optimierung mit Hilfe von Cflags

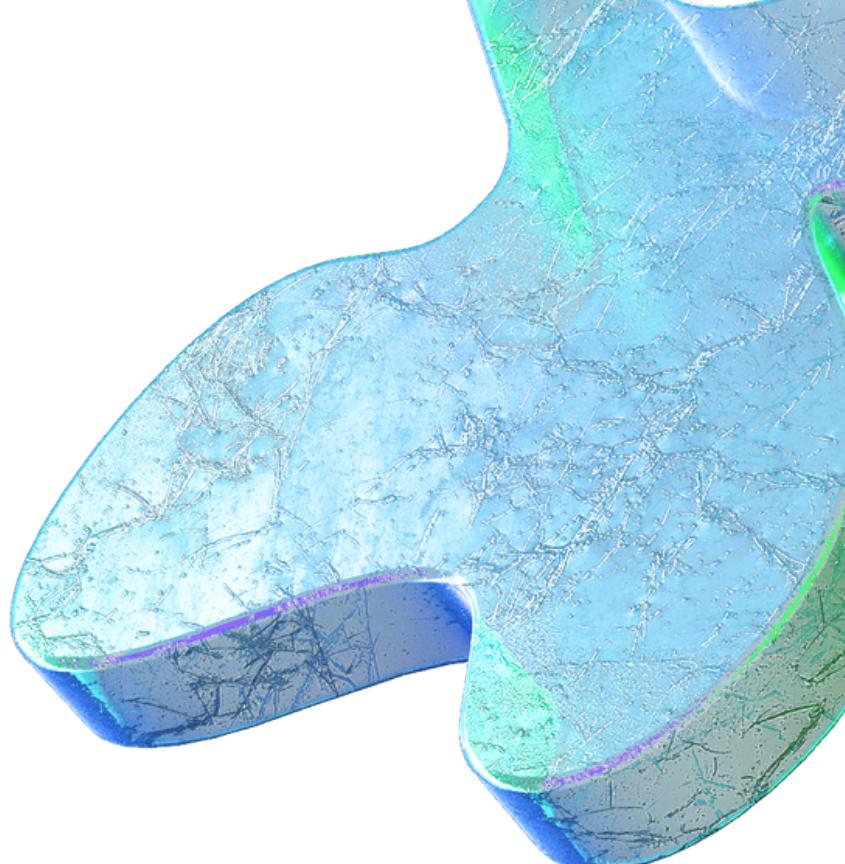


# PERFORMANZANALYSE

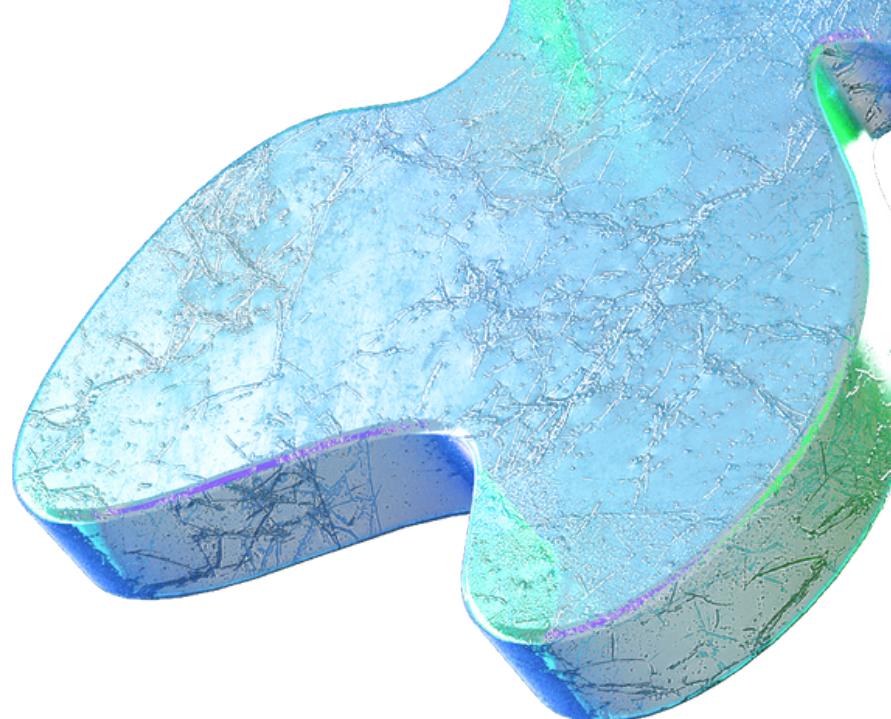
# Testplattform : Rechnerhalle

## Hardware:

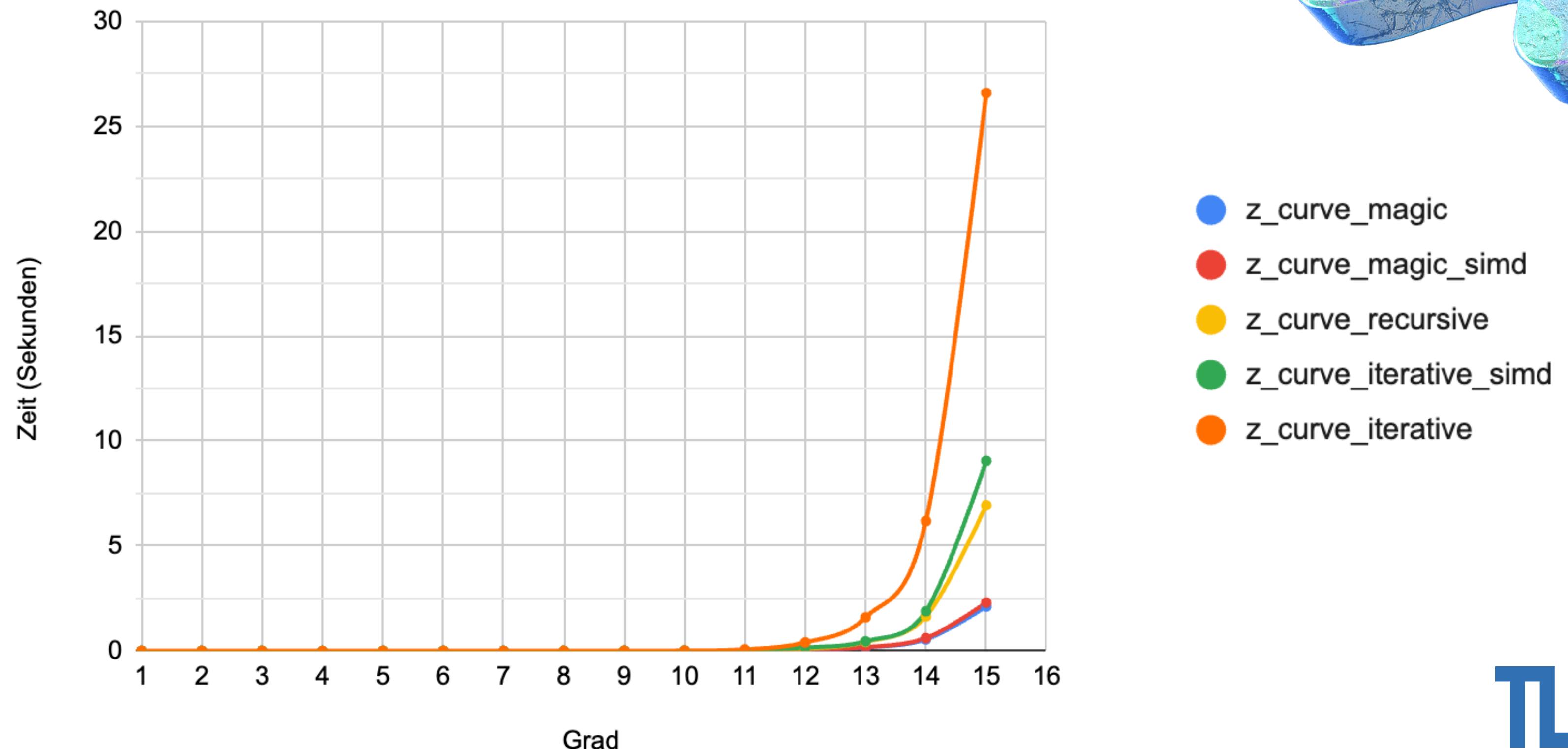
- Architektur: **x86\_64**
- CPU: **Intel(R) Xeon(R) CPU E5-2687W v3 @ 3.10GHz**
- Betriebssystem: **Ubuntu 22.04.1 LTS**
- CFLAGS: **-O3 -march=native**



# Grafik der Laufzeit verschiedener Methoden.

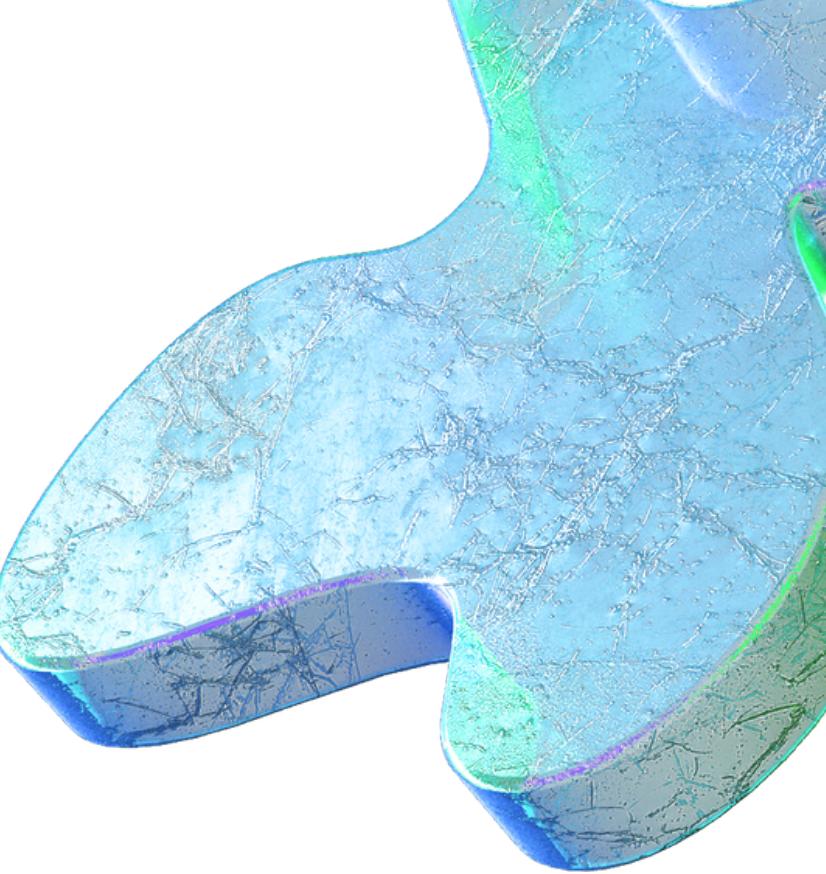
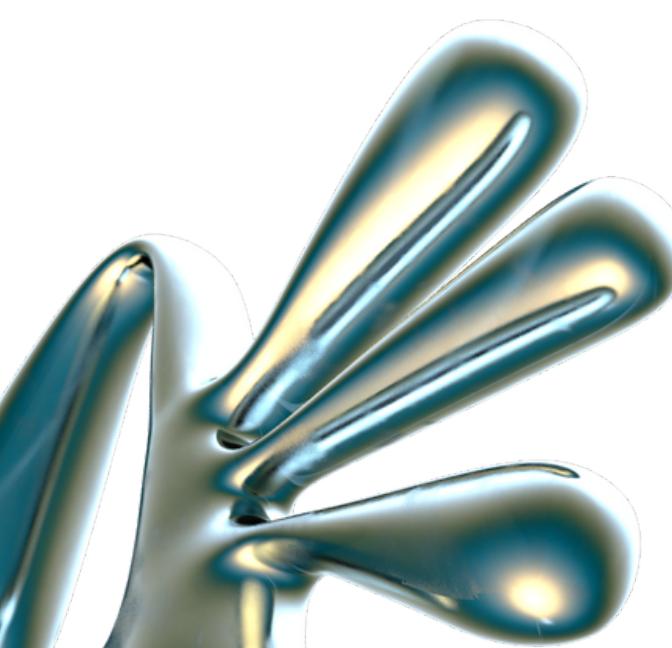


Z-Kurve Graph



# QUELEN

- [Slide 4,5] [https://en.wikipedia.org/wiki/Z-order\\_curve](https://en.wikipedia.org/wiki/Z-order_curve)



**Danke für Ihre  
Aufmerksamkeit**

