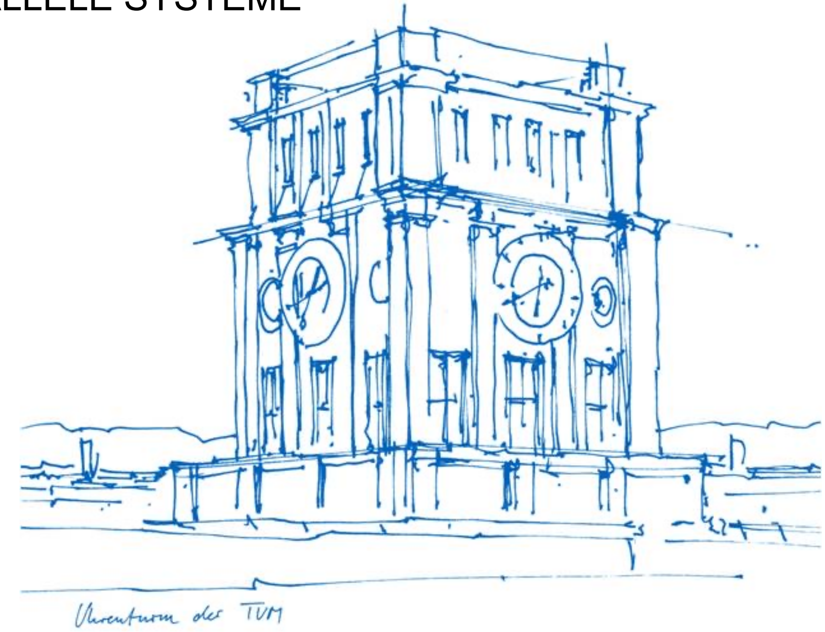
Ludwig

# Grundlagenpraktikum: Rechnerarchitektur

LEHRSTUHL FÜRRECHNERARCHITEKTUR UND PARALLELE SYSTEME



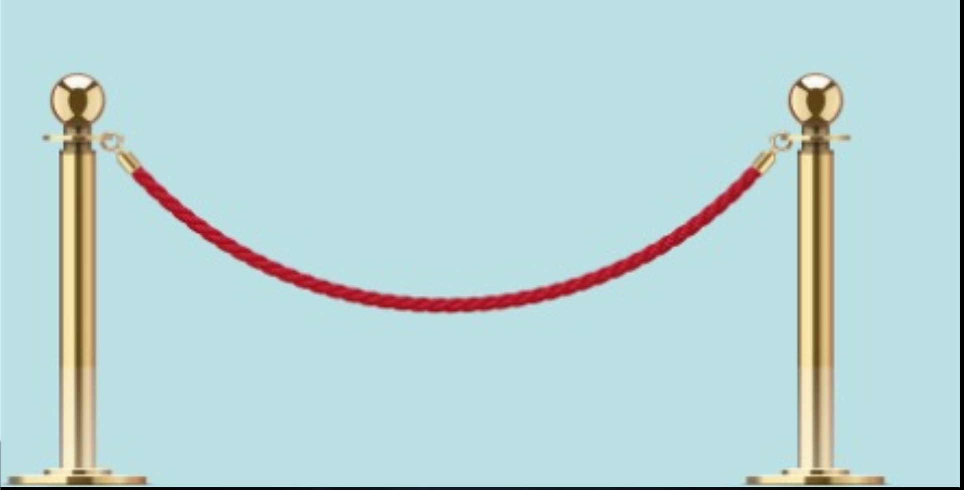

Gruppe 233 – Vortrag zu Aufgabe A316

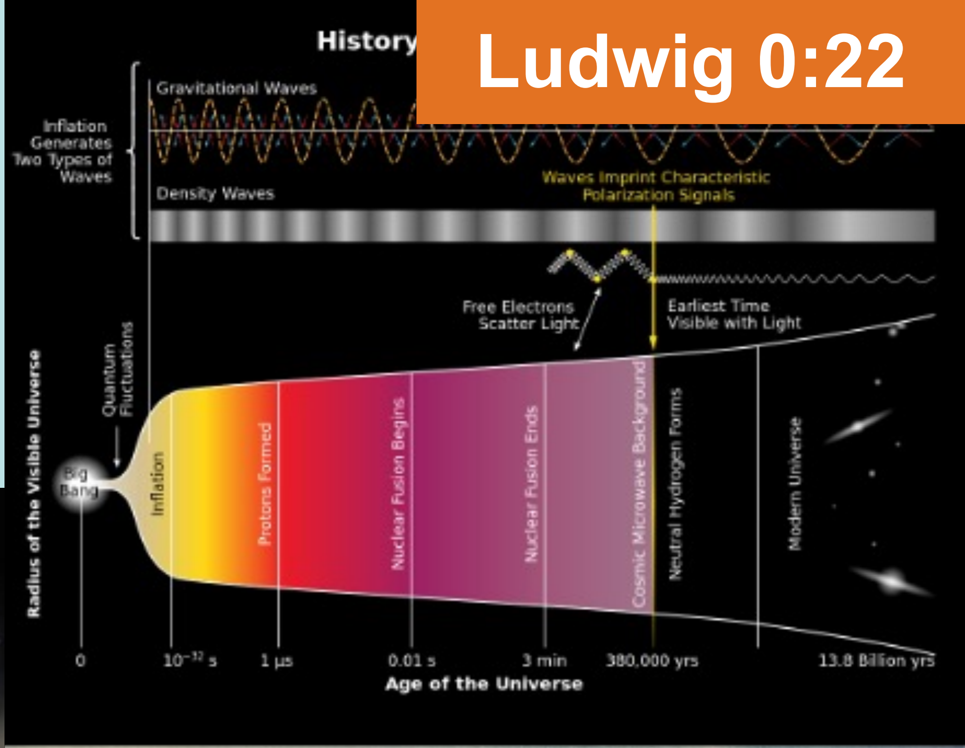
Sommersemester 2023

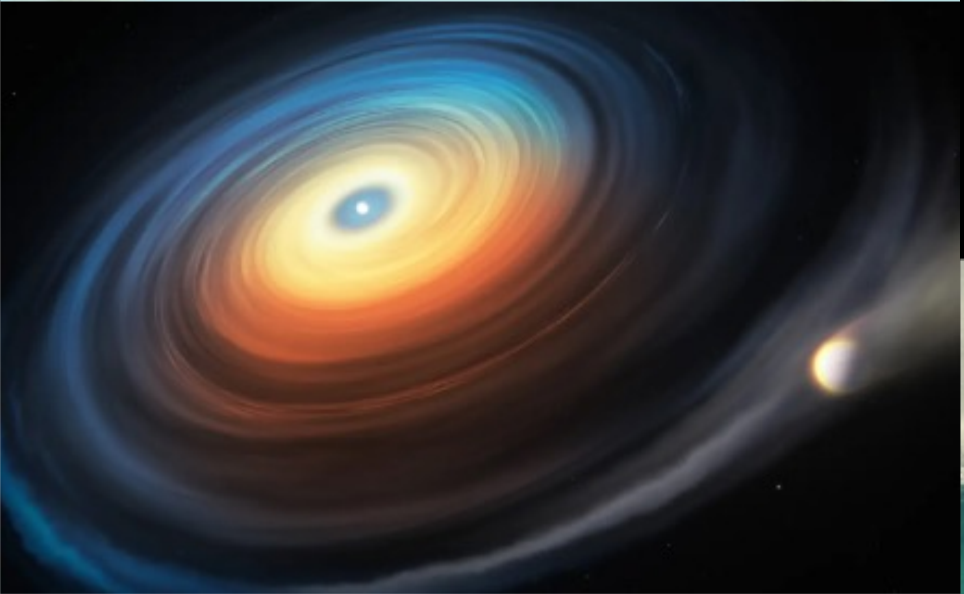
Ludwig Gröber, Julian Pins, Daniel Safyan

München, 21. August 2023

# Ludwig 0:22

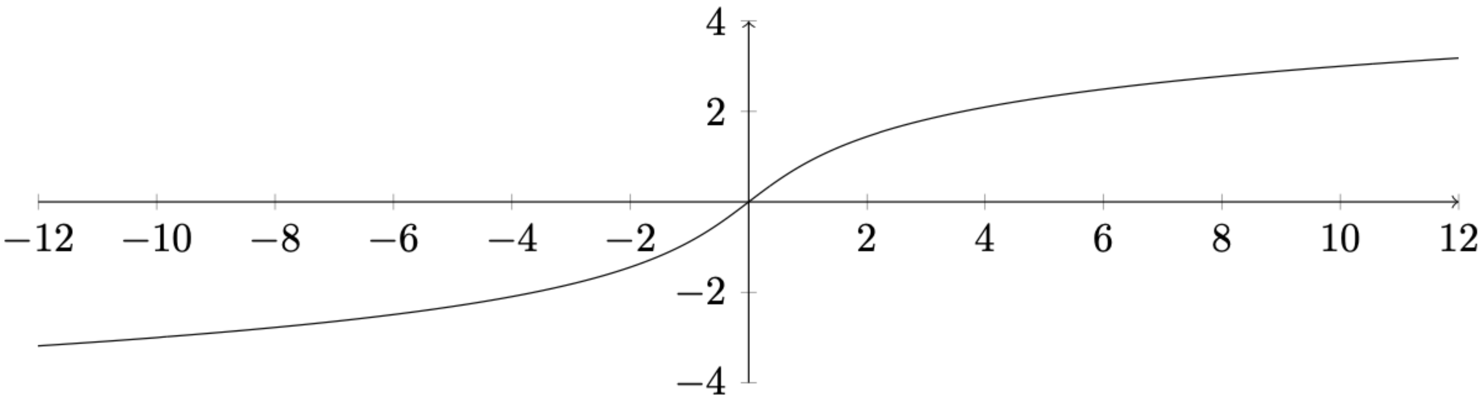



History
Inflation Generates Two Types of Waves
Gravitational Waves
Density Waves
Waves Imprint Characteristic Polarization Signals
Free Electrons Scatter Light
Earliest Time Visible with Light
Quantum Fluctuations
Big Bang
Inflation
Protons Formed
Nuclear Fusion Begins
Nuclear Fusion Ends
Cosmic Microwave Background
Neutral Hydrogen Forms
Modern Universe
Radius of the Visible Universe
0
10⁻³² s
1 µs
0.01 s
3 min
380,000 yrs
13.8 Billion yrs
Age of the Universe



Ludwig Gröber, Julian Pins, Daniel Safyan

# Ludwig 0:49




$arsinh(x) = ln(x + \sqrt{x^2 + 1})$

(1) der Lösungsansatz

(2) die angewandten Optimierungen

(3) die Performanz

(4) die Genauigkeit

(5) ein Ausblick und die Einordnung

Daniel

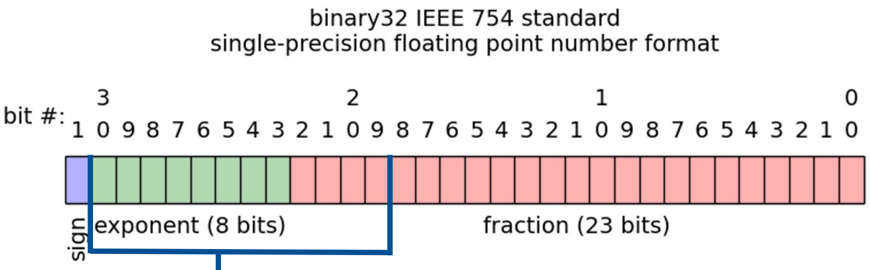
# (1) Lösungsansatz

Ludwig Gröber, Julian Pins, Daniel Safyan

München, 21. August 2023

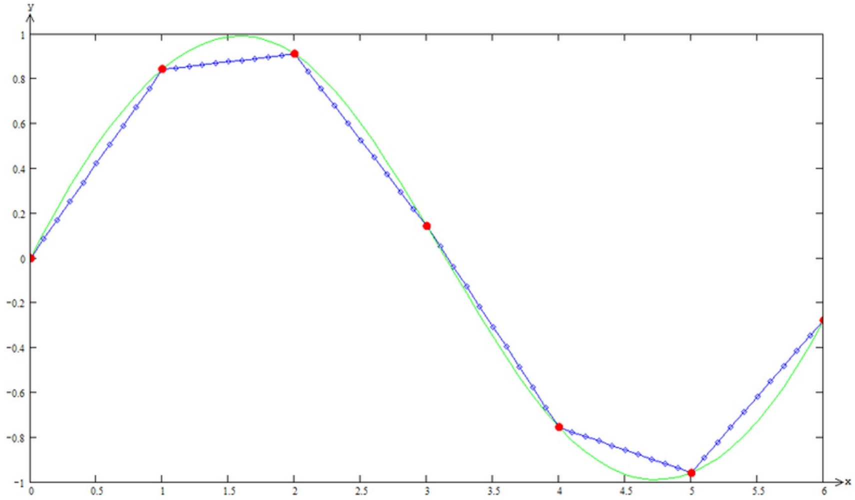
# Lookup-Tabelle






Tabelle

| | |
|---|---|
| 0 | 1.1125369292536007e-308 |
| 1 | 1.1820704873319507e-308 |
| 2 | 1.251604045410301e-308 |
| 3 | 1.321137603488651e-308 |
| 4 | 1.390671161567001e-308 |
| 5 | 1.460204719645351e-308 |
| 6 | 1.529738277723701e-308 |
| 7 | 1.599271835802051e-308 |
| ⋮ | ⋮ |



# Problematik einer Reihenentwicklung

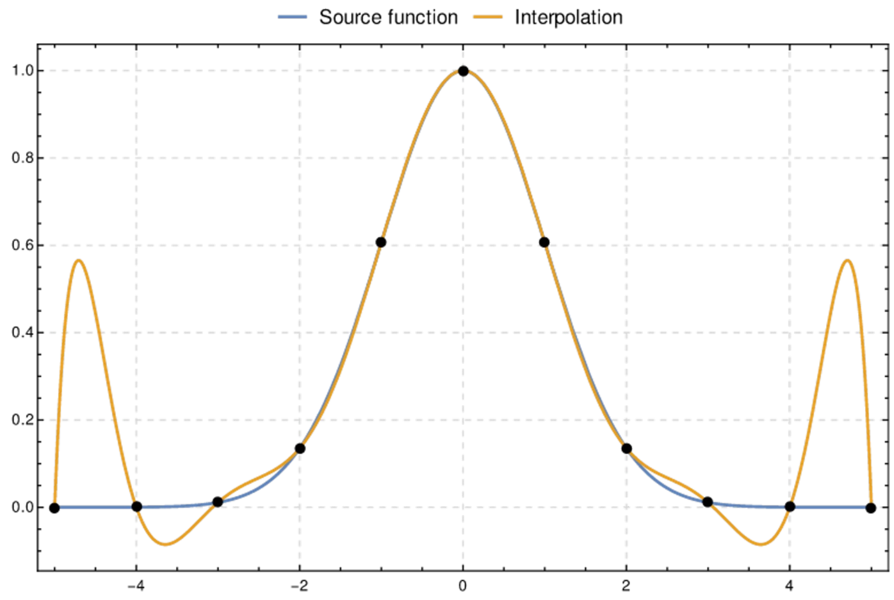
$$\sum_{k=0}^{\infty} a_k x^{\pm k}$$

Größtes erwartetes Ergebnis:

$$arsinh(DOUBLE\_MAX) \approx 710{,}41$$

Gemischte Reihe:

$$\operatorname{arsinh}(x) = \begin{cases} TaylorArsinh & \text{falls } |x| < 1 \\ \ln(2x) + error(x) & \text{falls } |x| > 1 \\ x & \text{falls } x \in \{\pm \inf, \pm Nan\} \end{cases}$$



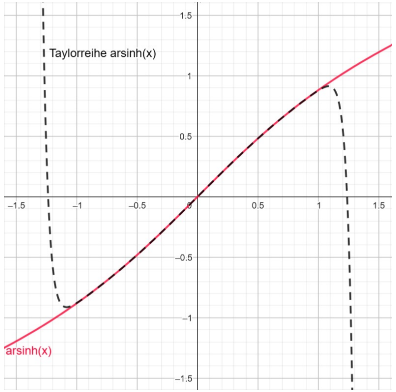
Daniel

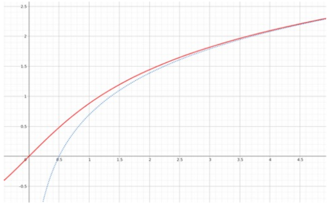
# Reihenentwicklung

$$|x| \leq 1$$

**Taylor Entwicklung um 0:**

$$arsinh(x) = \sum_{k=0}^{\infty} \frac{(2k-1)!!(-x^2)^k}{(2k)!!(2k+1)} = \sum_{k=0}^{\infty} \frac{(-1)^k (2k)! x^{2k+1}}{(2k+1)(2^k * k!)^2}$$



$$x > 1$$

**Approximation:**

$$arshinh(x) = \ln\left(x + \sqrt{x^2+1}\right) = \ln(2x) + error(x)$$

**Reine Reihe:**

$$\ln(2) - \left(\frac{1}{x}\right) + error(x)$$

**Gemischte Reihe:**

$$\ln(2x) = \ln(2) + \ln(x) = \ln(2) + \ln(M * 2E) = \ln(2) + E * \ln(2) + \ln(M)$$

Ludwig

# (2) Optimierungen

Ludwig Gröber,  Julian Pins,  Daniel Safyan

München, 21. August 2023

# Optimierungen

Ludwig 1:45

**Reihen: Tylor-Entwicklungen**

- Anzahl der Reihenglieder gewählt
- Koeffizienten der drei Reihen vorberechnet
- Horner-Schema angewandt

$$2x^4 - 4x^3 - 5x^2 + 7x + 11 =$$

$$\left(((2 * x - 4) * x - 5) * x + 7\right) * x + 11$$

- Absorption für kleine und große $x$ -> weniger Reihenglieder

**Tabellen-Lookup**

- Logarithmische Verteilung der Werte in der Lookup-Tabelle
- Exponent und erste 4 Bit der Mantisse -> Index des nächst kleineren Wertes der Tabelle
- Punktsymmetrie: $arsinh(-x) = -arsinh(x) \rightarrow$ nur betragsmäßige Betrachtung
- Index der Tabelle aus Exponent und höchstwertigen vier Bit der Mantisse errechnet
- Lineare Interpolation zwischen Werten

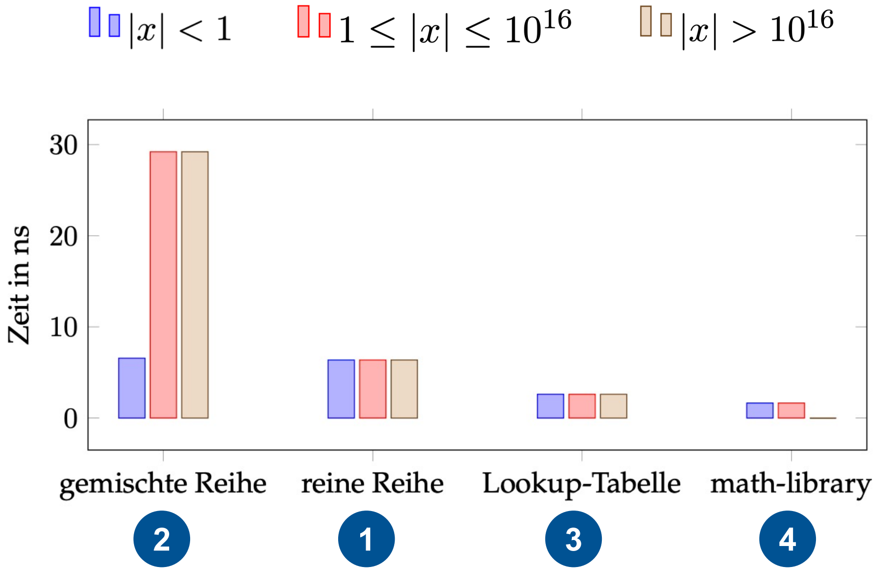
# (3) Performanz

Ludwig Gröber, Julian Pins, Daniel Safyan

München, 21. August 2023

# Performanz

Ludwig 0:50




1. reine Reihe ist rechen-intensiv, da für alle $x$ 13 Reihenglieder berechnet werden
2. gemischte Reihe für $|x| \geq 1$ am langsamsten, da 40 Reihenglieder berechnet werden
2. gemischte Reihe für $|x| < 1$ fast gleich schnell wie die reine Reihe, da gleiche Reihe verwendet
3. Lookup-Table verbessert die Laufzeit auf Kosten des Speicherverbrauchs von etwa 255KB
4. <math.h> für große $x$ auch mit mehr als $10^{11}$ Wiederholungen Laufzeit 0ns

# (4) Genauigkeit

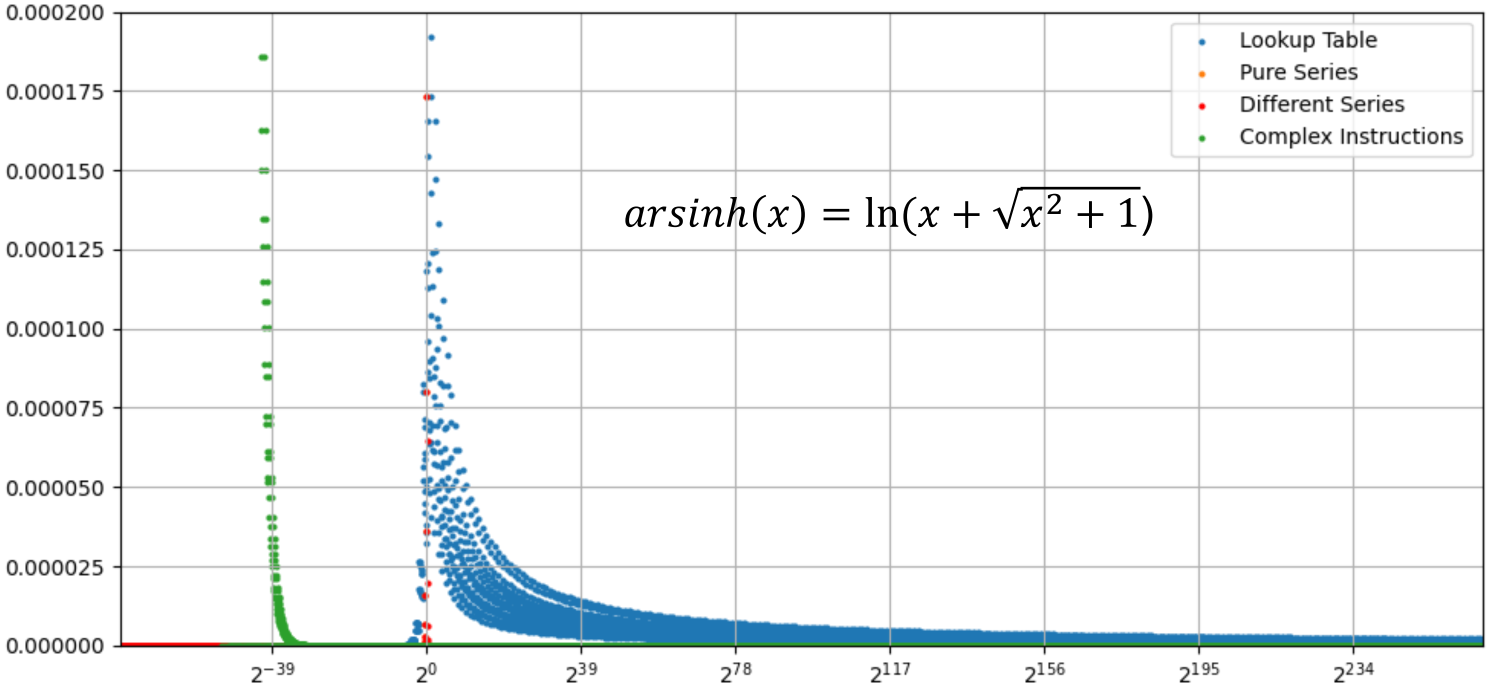
Ludwig Gröber, Julian Pins, Daniel Safyan

München, 21. August 2023

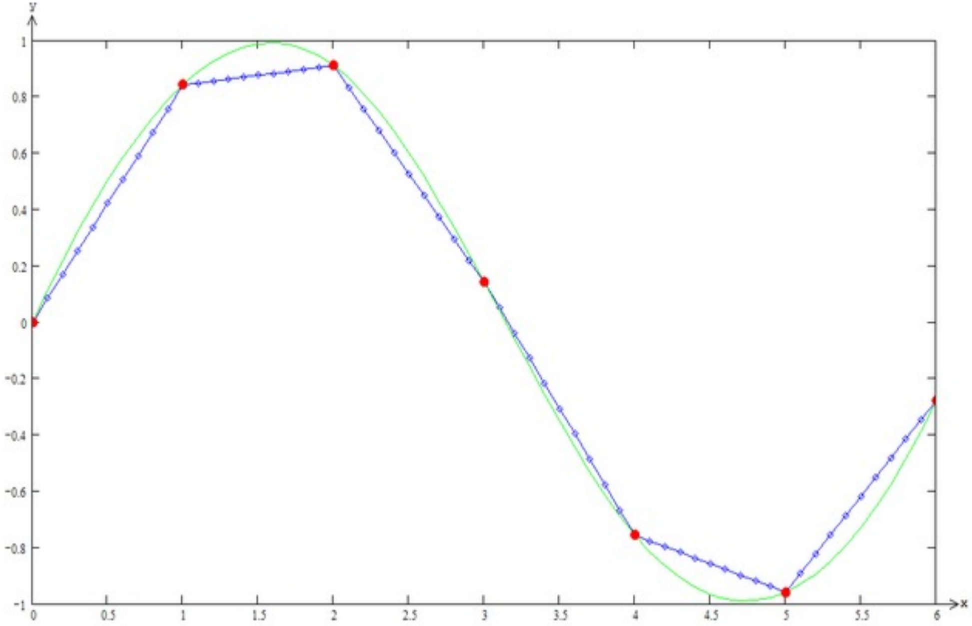
# Genauigkeit

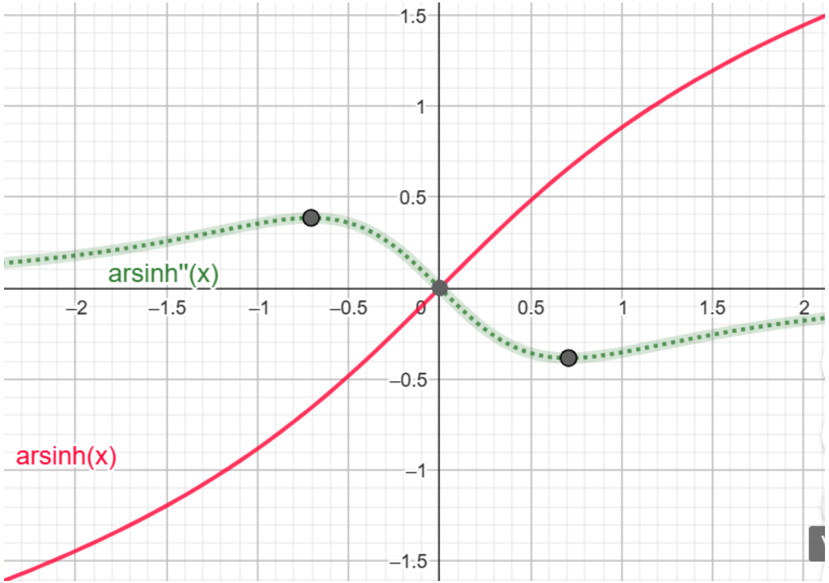
Pins

Vergleichswert: relativer Fehler



# Genauigkeit Reihenentwicklung



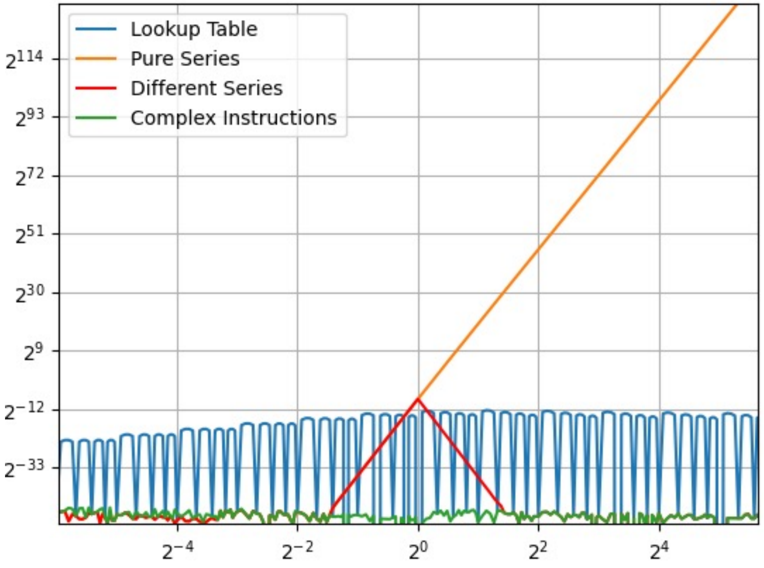


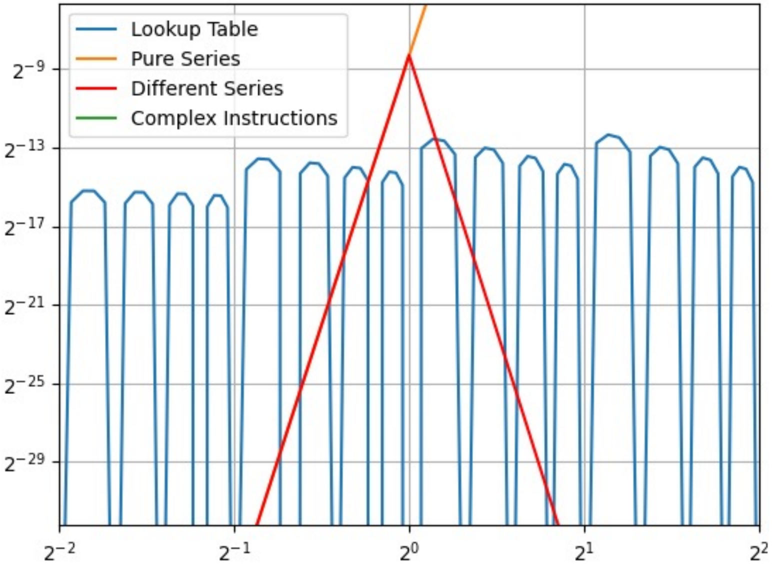
# Genauigkeit

Pins

Vergleichswert: relativer Fehler

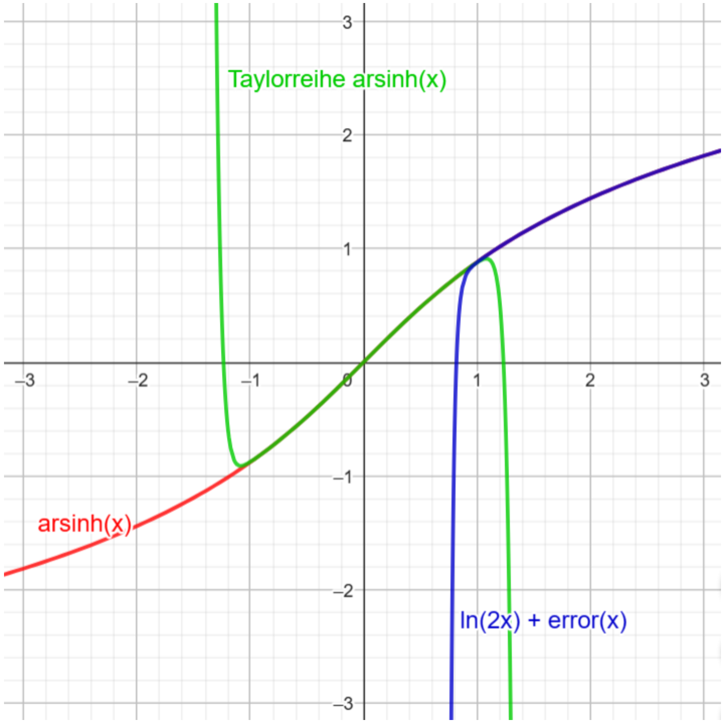
$$arsinh(x) = \sum_{k=0}^{\infty} \frac{(2k-1)!!(-x^2)^k}{(2k)!!(2k+1)} = \sum_{k=0}^{\infty} \frac{(-1)^k (2k)! x^{2k+1}}{(2k+1)(2^k * k!)^2}$$



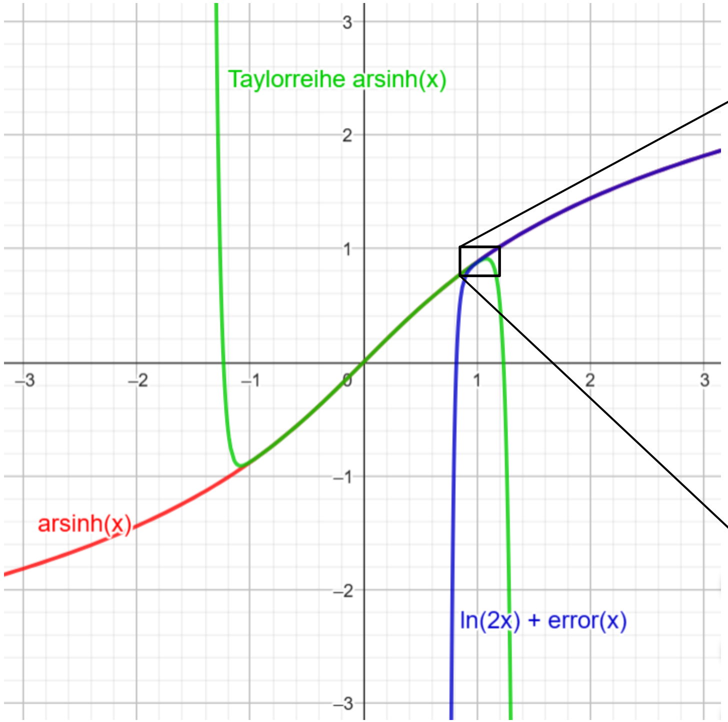
# Reihenentwicklungen

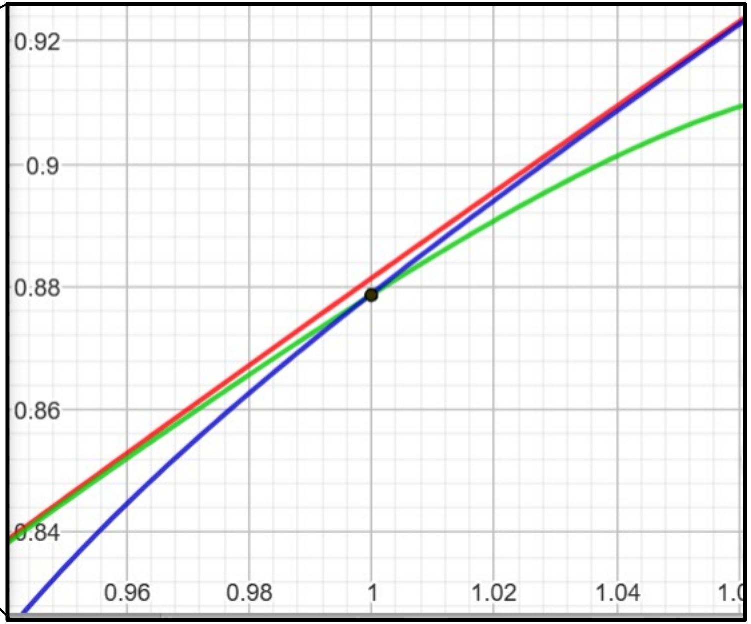
Pins



# Reihenentwicklungen

Pins



# (5) Einordnung und Ausblick

Ludwig Gröber, Julian Pins, Daniel Safyan

München, 21. August 2023

Pins

# Zusammenfassung

| | **Genauigkeit** | **Performanz (worst case)** | **Speicherverbrauch** |
|---|---|---|---|
| **Lookuptabelle** | Bis zu 0.02 % rel. Fehler für Eingaben zwischen Datenpunkten | 2.61 ns | Ca. 500 KB für Tabelle |
| **Reine Reihendarstellung** | Nur Konvergenzbereich $\lvert x\rvert < 1$ | 6.37 ns | Niedrig |
| **Gemischte Reihendarstellung** | Sehr hoch<br>Ungenau für $0.25 < \lvert x\rvert < 4$ | 29.21 ns | Niedrig |
| **Komplexe Instruktionen** | Sehr hoch<br>Ungenau für $\lvert x\rvert < 2^{-26}$ | 1.65 ns | Niedrig |

# Ausblick: weitere Optimierungen

Pins

Allgemein:

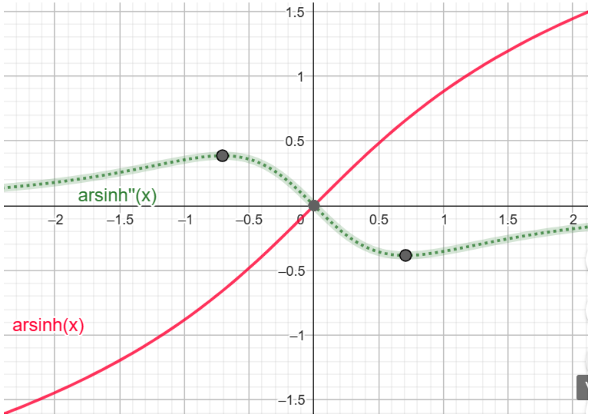
- Datentyp float für Speichereinsparung
- Mikrooptimierungen Assembly
- Verwendung von Reihendarstellung/ Lookuptabelle abhängig vom Eingabewert

Reihendarstellungen:

- Anzahl Reihenglieder abhängig von Eingabewert

Lookuptabelle:

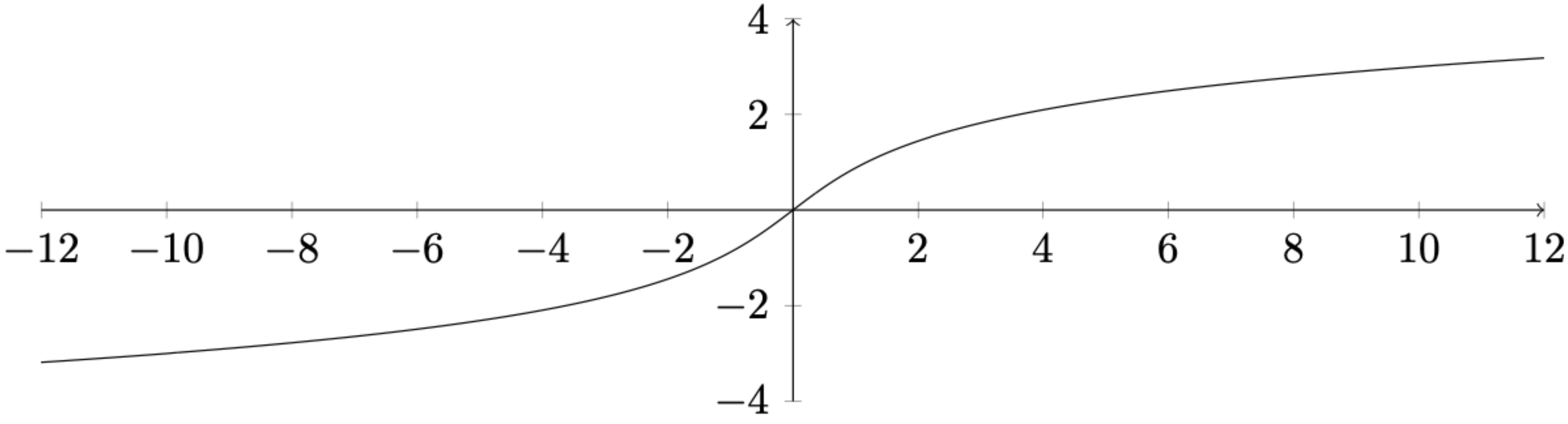
- Splines
- Umverteilung der Messpunkte in der Lookuptabelle



Ludwig Gröber, Julian Pins, Daniel Safyan

Pins

# Danke für die Aufmerksamkeit & Zeit für Fragen




$$— \; arsinh(x) = ln(x + \sqrt{x^2 + 1})$$

# Quellen


**O.V. (2019),** https://meerdavon.com/wipe-out-aengste-surfen/ (Aufgerufen am: 15.07.2023) „Surfen“

**Preuß, M. (2019),** https://science-to-go.com/die-kettenlinie-2/ (Aufgerufen am: 15.07.2023) „Kettenlinie“

**Hartung, L. (2019),** https://www.spektrum.de/news/weisser-zwerg-nagt-riesenplaneten-an/1689980 (Aufgerufen am: 15.07.2023) „Gravitationswelle“

**O.V. (2014),** https://commons.wikimedia.org/wiki/File:History_of_the_Universe_%28multilingual%29.svg (Aufgerufen am: 15.07.2023) „Ausdehnung des Universums“

**Stolfi, J. (2009),** https://commons.wikimedia.org/wiki/File:Hash_table_3_1_1_0_1_0_0_SP.svg (Aufgerufen am 16.07.2023) “Hash-Map”

**Rtnick (2010),** https://commons.wikimedia.org/wiki/File:Lin_interp_w-legend.png (Aufgerufen am 16.07.2023) “Linear interpolation”

# Quellen

**O.V. (2016),** https://commons.wikimedia.org/wiki/File:Runge's_phenomenon_in_Lagrange_polynomials.svg (Aufgerufen am 16.07.2023) “Runge Effekt”

**O.V. (2022),** https://commons.wikimedia.org/wiki/File:Binary32_format_for_single-precision_floating_point_number.png (Aufgerufen am 16.07.2023) “double”