

Geburtstagsparadox

Leonardo Maglanoc, Julia Spindler, Ronald Skorobogat

TUM

July 18, 2020

Gliederung

Einleitung

Lösungsansatz

Genauigkeit

Performanz

Ausblick

Gliederung

Einleitung

Lösungsansatz

Genauigkeit

Performanz

Ausblick

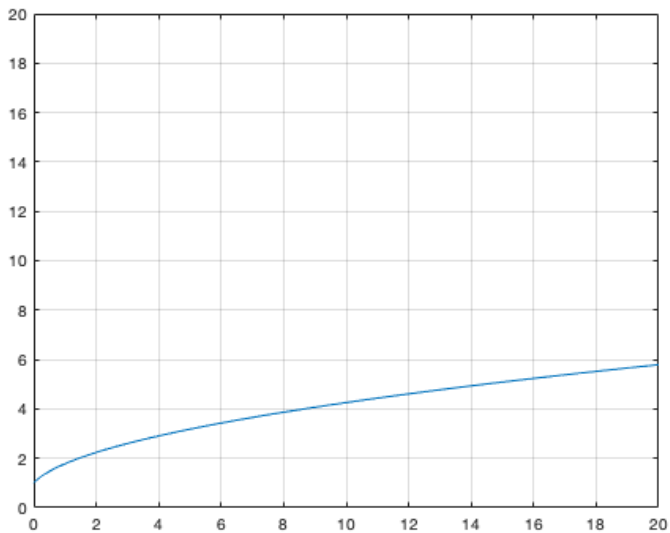
Einleitung

$$P = 1 - \left(\underbrace{\frac{n}{n} \cdot \frac{n-1}{n} \cdot \dots \cdot \frac{n-k+1}{n}}_{k\text{-mal}} \right) = 1 - \frac{n!}{n^k \cdot (n-k)!}$$

$$k \geq \frac{1 + \sqrt{1 + 8n \cdot \ln 2}}{2}$$

Näherungsverfahren für: $\sqrt{1 + 8n \cdot \ln 2}$

Verlauf der Geburtstagsfunktion



Gliederung

Einleitung

Lösungsansatz

Genauigkeit

Performanz

Ausblick

Lookup-Tabelle

- LUT speichert 2^{17} Float-Werte
- 1. Hälfte: $i \in [0 \dots 2^{16} - 1] \implies \sqrt{1 + 8 \cdot \ln(2) \cdot i}$
- Parameter kann direkt als Index verwendet werden
- Index i gibt den genauen Wert unter der Formel mit i als Eingabe

2. Tabellenhälfte mit exponentiellen Abständen

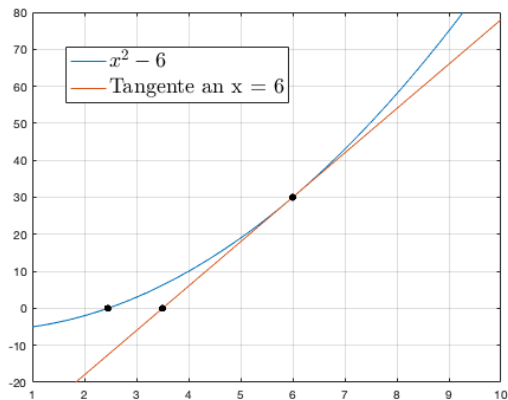
$$\text{table}[i] = \sqrt{(8 \cdot \ln(2) \cdot 2^{16}) + k^i}$$

$$i = \log_k(m) = \left\lfloor \frac{\text{Exponent} - 127}{\log_2(k)} + \log_k(\text{Mantisse}) \right\rfloor$$

Heron-Verfahren

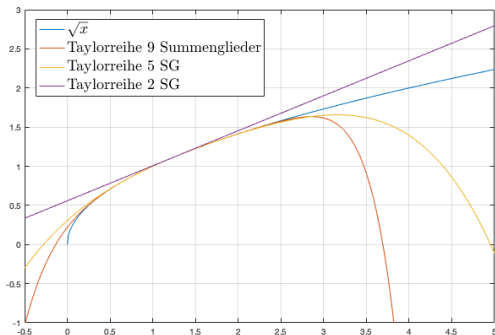
$$x_{n+1} = \frac{1}{2} \cdot \left(x_n + \frac{q}{x_n} \right)$$

q : die 1. Approximation



Reihendarstellung

- Taylorreihe nähert über Ableitungen die Wurzelfunktion an
- Näherung nur in einem Bereich möglich
- Gewählter Bereich: $[0.25, 2.25]$



Umsetzung mit größeren Zahlen

- Verkleinern jeder Zahl auf den Bereich durch Division mit Quadrat einer Zweierpotenz
- Finden dieses Bereichs durch binäre Suche
- Zahl unter der Wurzel liegt letztendlich zwischen 0.5 und 2

Beispielrechnung

$$x = 489$$

$$\text{Intervall: }]2^7, 2^9]$$

$$\text{Zugeordnete Zweierpotenz: } \sqrt{2^8} = 2^4$$

$$x' = \frac{489}{2^{4^2}} \approx 1.91$$

$$\text{Nach Taylorreihe: } \sqrt{\frac{489}{2^{4^2}}} = \frac{\sqrt{489}}{2^4}$$

Rahmenprogramm

- -s Option Kein_Argument
- -t Option Kein_Argument
- -b Option Anzahl_von_Iterationen
- -h Option Kein_Argument

Ausführungsbeispiel:

```
.../team120/Implementierung$ ./main -b 100
```

Gliederung

Einleitung

Lösungsansatz

Genauigkeit

Performanz

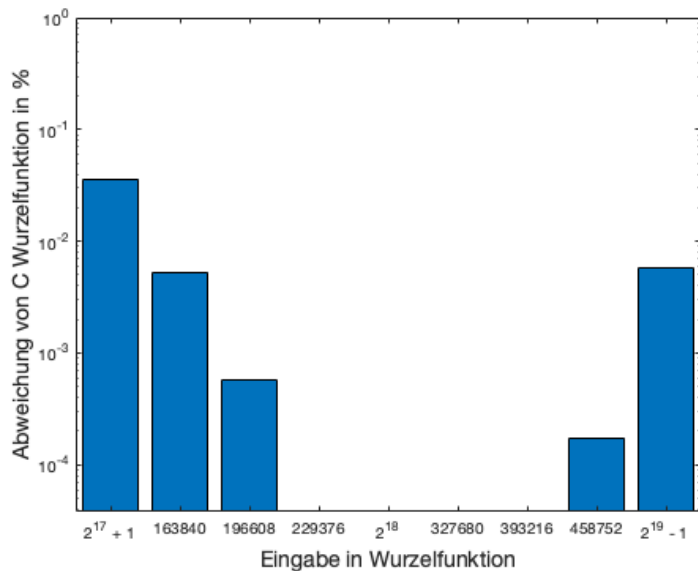
Ausblick

Genauigkeit

Eingabe	Abweichung Tabelle	Abweichung Reihe	Abweichung Asm
2	0.0000153%	-0.0003815%	0.0000153%
33	-0.0000153%	0.0009689%	-0.0000153%
73	0.0000153%	0.0000153%	0.0000153%
365	0%	0.0043488%	0%
737474	-0.0000153%	-0.0000153%	-0.0000153%
9898989	-0.0000153%	-0.0001450%	0%
9898242989	-0.0000153%	-0.0002060%	0%
532578665767	0%	-0.0024109%	0%
$2^{64} - 1$	0%	0%	0%

Durchschnittliche Genauigkeit: >99.95%

Genauigkeit der Reihenimplementierung



Gliederung

Einleitung

Lösungsansatz

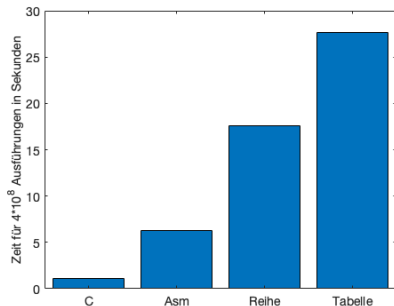
Genauigkeit

Performanz

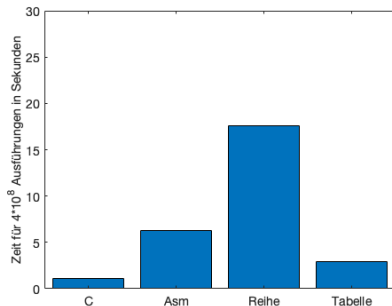
Ausblick

Performanz

Performanz auf Rechner mit CPU Intel Celeron T3500 @ 2.10GHz,
Ubuntu 20.04, Linux-Kernel 5.4.0



Performanzergebnisse im
Wertebereich $[2^{16}, 2^{64} - 1]$



Performanzergebnisse im
Wertebereich $[0, 2^{16} - 1]$

Gliederung

Einleitung

Lösungsansatz

Genauigkeit

Performanz

Ausblick

- LUT-Methode: Optimierung der Heron-Berechnungen
- Reihen-Methode: Einteilung des Eingabeparameters in mehr Bereiche
- eher sinnlos standardisierte (Mathe-)Funktionen selber zu implementieren