CARL VON OSSIETZKY universität OLDENBURG

Pollards Rho-Methode zur Faktorisierung

Abschlusspräsentation Bachelorarbeit
Janosch Döcker

Carl von Ossietzky Universität Oldenburg
Department für Informatik
Abteilung Parallele Systeme

Inhaltsverzeichnis

Übersicht

Das Faktorisierungsproblem

Definition (Faktorisierungsproblem)

- Gegeben: Eine zusammengesetzte Zahl $N \geq 4$
- Gesucht: Die Primfaktorzerlegung von N

- Faktorisierungsproblem vermutlich nicht effizient lösbar
- Praktische Relevanz: Sicherheit von RSA

Wichtige Fakten zur Rho-Methode

- 1975 von John M. Pollard beschrieben
- Probabilistisches Verfahren
- Laufzeit abhängig von der Größe der Primfaktoren
- Analyse basiert auf Annahmen
- 1980 von Richard P. Brent verbessert
- Faktorisierung der Fermatzahl $F_8 = 2^{2^8} + 1 =$
  115 792 089 237 316 195 423 570 985 008 687 907 853
  269 984 665 640 564 039 457 584 007 913 129 639 937

Übersicht

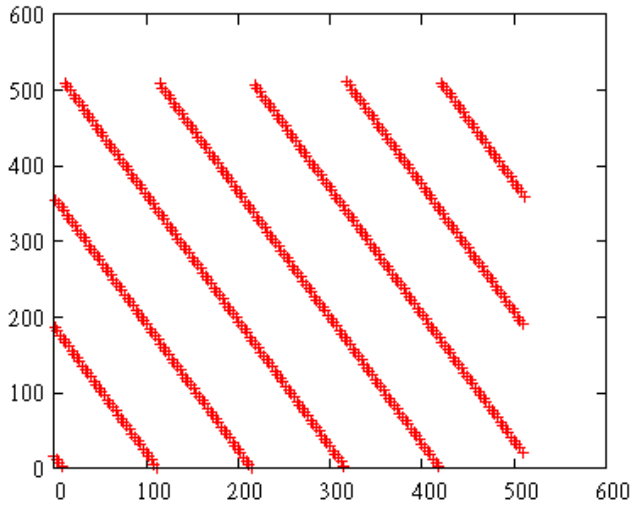
Pseudozufallszahlen

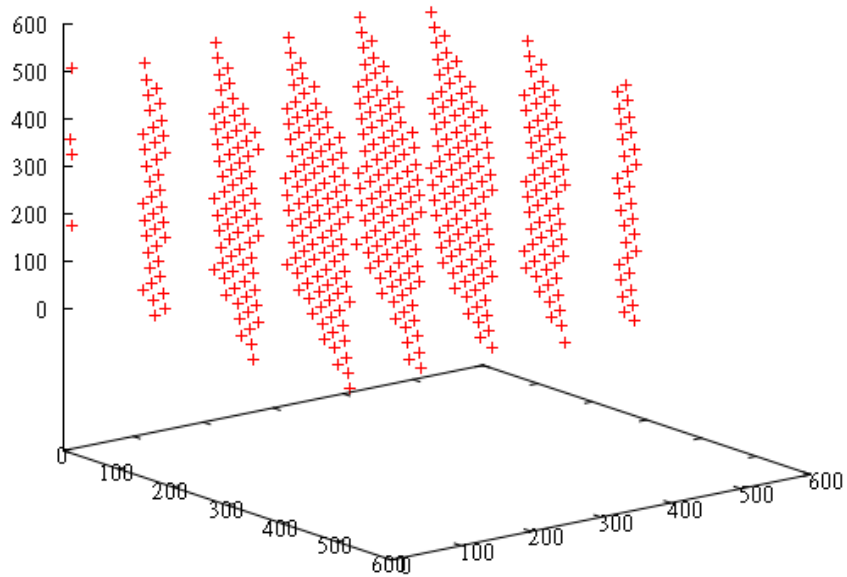
- Philosophische Frage: Gibt es echten Zufall?
- Pseudozufallszahlen deterministisch erzeugt
- Bekannte Methode: Linearer Kongruenzgenerator (LKG)
- Entwickelt von Derrick H. Lehmer
- LKG: $x_{i+1} = (ax_i + c) \bmod m$ mit $x_0 \in \{0, 1, \ldots, m-1\}$
- Ausreichende „Zufälligkeit“ für viele Anwendungen ...
- ... wenn die Parameter gut gewählt sind.

Linearer Kongruenzgenerator

- LKG hat Schwachstellen unabhängig von der Parameterwahl
- G. Marsaglia: „Random numbers fall mainly in the planes“ (1968)

Beispiel: $x_{i+1} = (169x_i + 17) \bmod 512$

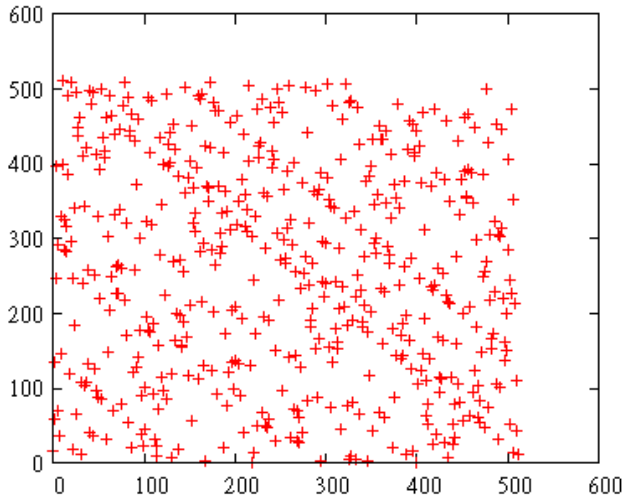


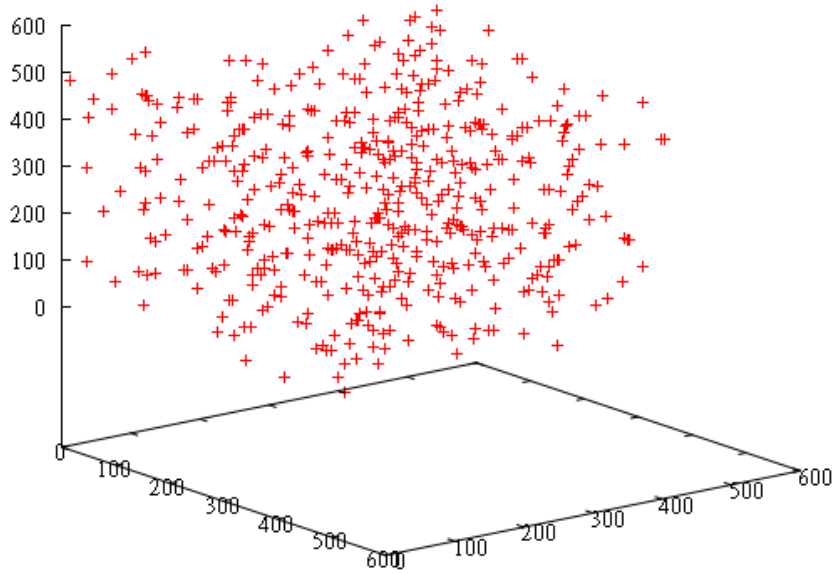


Quadratischer Kongruenzgenerator(QKG)

- Vorgeschlagen von Donald E. Knuth
- QKG: $x_{i+1} = (ax_i^2 + dx_i + c) \bmod m$ mit $x_0 \in \{0, 1, \ldots, m-1\}$

Beispiel: $x_{i+1} = (18x_i^2 + 23x_i + 17) \bmod 512$



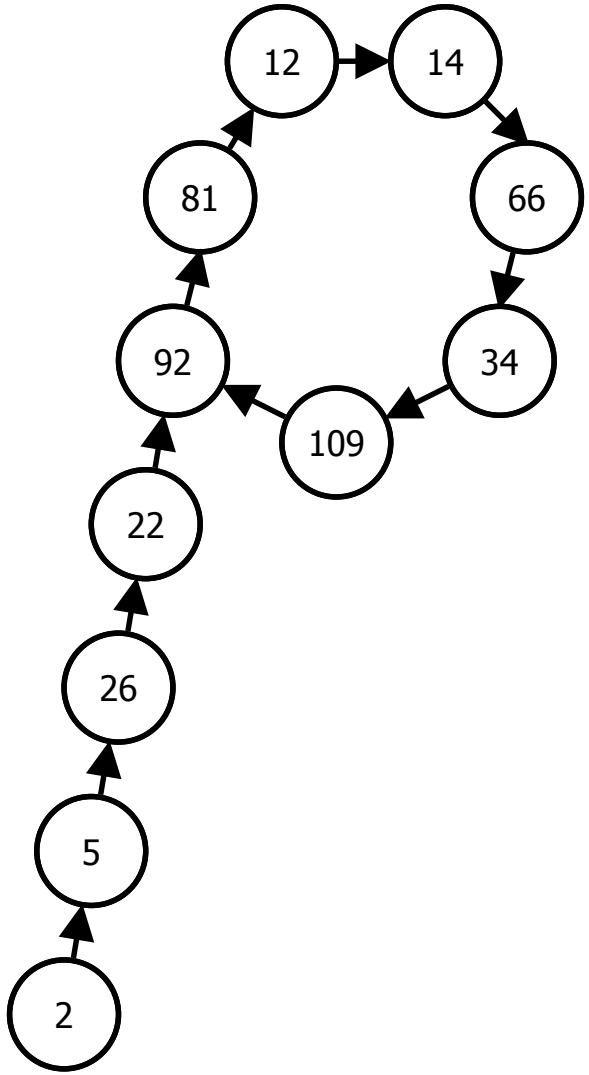


Vergleich zwischen LKG und QKG

- LKG einfacher zu berechnen
- LKG linear, QKG nicht linear
- Pseudozufallszahlen erscheinen beim QKG „zufälliger“
- Beide Generatoren erzeugen *schließlich periodische* Folgen

Beispiel: Schließlich periodische Folge

- $x_0 = 2$, $x_{i+1} = (x_i^2 + 1) \bmod 131$
- Länge der Vorperiode: 4
- Periodenlänge: 7



Floyds Algorithmus

- M endliche Menge, $f : M \to M$, $C \in M$
- Anwendbar auf schließlich periodische Folgen der Form:

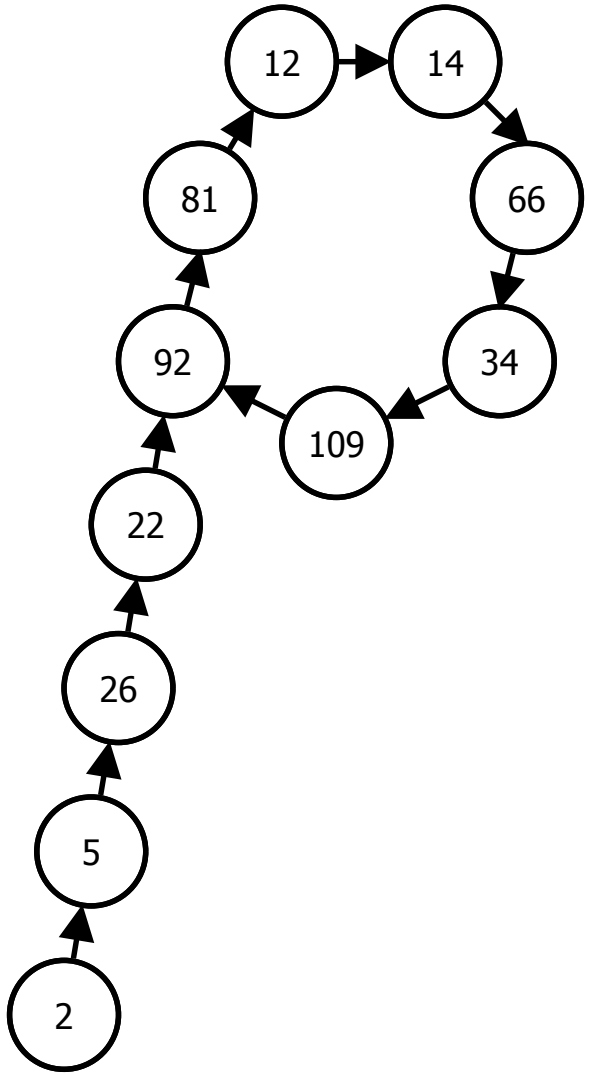
$$x_0 = C, \quad x_{i+1} = f(x_i)$$

- Bestimmung eines Vielfachen der Periodenlänge
- Auffinden eines wiederkehrenden Elements
- Idee: $x_k \stackrel{?}{=} x_{2k}$ für $k = 1, 2, \ldots$
- Nach Floyd: $x_0 = y_0 = C,\ x_{i+1} = f(x_i),\ y_{i+1} = f(f(y_i))$
- Sehr geringer Speicherverbrauch

Beispiel: Floyds Algorithmus

- $x_0 = 2$, $x_{i+1} = (x_i^2 + 1) \bmod 131$

k	x_k	$y_k = x_{2k}$
1	5	26
2	26	92
3	22	12
4	92	66
5	81	109
6	12	81
7	14	14



Übersicht

Wonach sucht die Rho-Methode?

- Sei $N \geq 4$ eine zusammengesetzte Zahl und p ein (unbekannter) Primfaktor von N.
- Gesucht: $x, y \in \mathbb{Z}$ mit $x \equiv y \pmod{p}$ und $x \not\equiv y \pmod{N}$
- Es gilt per definitionem:

  (1) $x \equiv y \pmod{p} \Leftrightarrow p \mid (x - y)$

  (2) $x \not\equiv y \pmod{N} \Leftrightarrow N \nmid (x - y)$

- Aus (1) und (2) folgt: $1 < \mathrm{ggT}(x - y,\ N) < N$
- Wichtiges Hilfsmittel: Euklidischer Algorithmus
- Wie können solche x, y (sinnvoll) bestimmt werden?

Herleitung der Rho-Methode

- Sei $f(x)$ eine ganzzahlige Polynomfunktion und $S \in \mathbb{Z}$
- Folge von Pseudozufallszahlen: $x_0 = S,\ x_{i+1} = f(x_i) \bmod N$
- Schließlich periodisch: $\tilde{x}_0 = S,\ \tilde{x}_{i+1} = f(\tilde{x}_i) \bmod p$
- $\tilde{x}_k \equiv x_k \pmod{p}$ für $k \geq 0$
- $\tilde{x}_k = \tilde{x}_{2k} \Rightarrow x_k \equiv x_{2k} \pmod{p} \Rightarrow p \mid \mathrm{ggT}(x_k - x_{2k},\ N)$
- Wie bei Floyds Algorithmus: $y_0 = S,\ y_{i+1} = f(f(y_i) \bmod N) \bmod N$
- Unterschied: $\mathrm{ggT}(x_k - y_k,\ N) \overset{?}{>} 1$ anstatt $x_k \overset{?}{=} y_k$

Beispiel Rho-Methode

Gesucht: Primfaktorzerlegung von $N = 74539$

Parameter: $x_0 = y_0 = 2$, $f(x) = (x^2 + 1) \bmod N$

k	$x_k = f(x_{k-1})$	$y_k = f(f(y_{k-1}))$	$\mathrm{ggT}(x_k - y_k, N)$
1	5	26	1
2	26	11096	1
3	677	536	1
4	11096	71723	1
5	57328	13078	1
6	536	3880	1
7	63680	23332	131

Damit ergibt sich $N = 131 \cdot 569$.

Eine einfache Implementierung

```
def pollard_rho(n)
  x = y = 2
  d = 1
  begin
    x = (x *  x + 1)  % n
    y = (y *  y + 1)  % n
    y = (y *  y + 1)  % n
    d = (x - y).gcd(n)
  end until d > 1
  puts d < n ? "Gefundener Faktor: #{d}" : "Fehlschlag!"
end
```

Demonstration

Demonstration der (kompletten) Implementierung anhand:

$$
\begin{aligned}
N_1 &= \quad 662\,835\,905\,978\,993\,515\,936\,337 \\
N_2 &= 1\,424\,842\,450\,293\,704\,631\,855\,941\,378\,617\,365\,082 \\
&\quad\; 792\,870\,362\,961\,939\,468\,399\,779\,353\,800\,137\,802 \\
&\quad\; 539\,831\,394\,422\,161\,828\,003\,733\,369\,548\,864\,158 \\
&\quad\; 809\,441\,716\,321
\end{aligned}
$$

Welche Zahl wird schneller in Primfaktoren zerlegt?

Verallgemeinertes Geburtstagsproblem

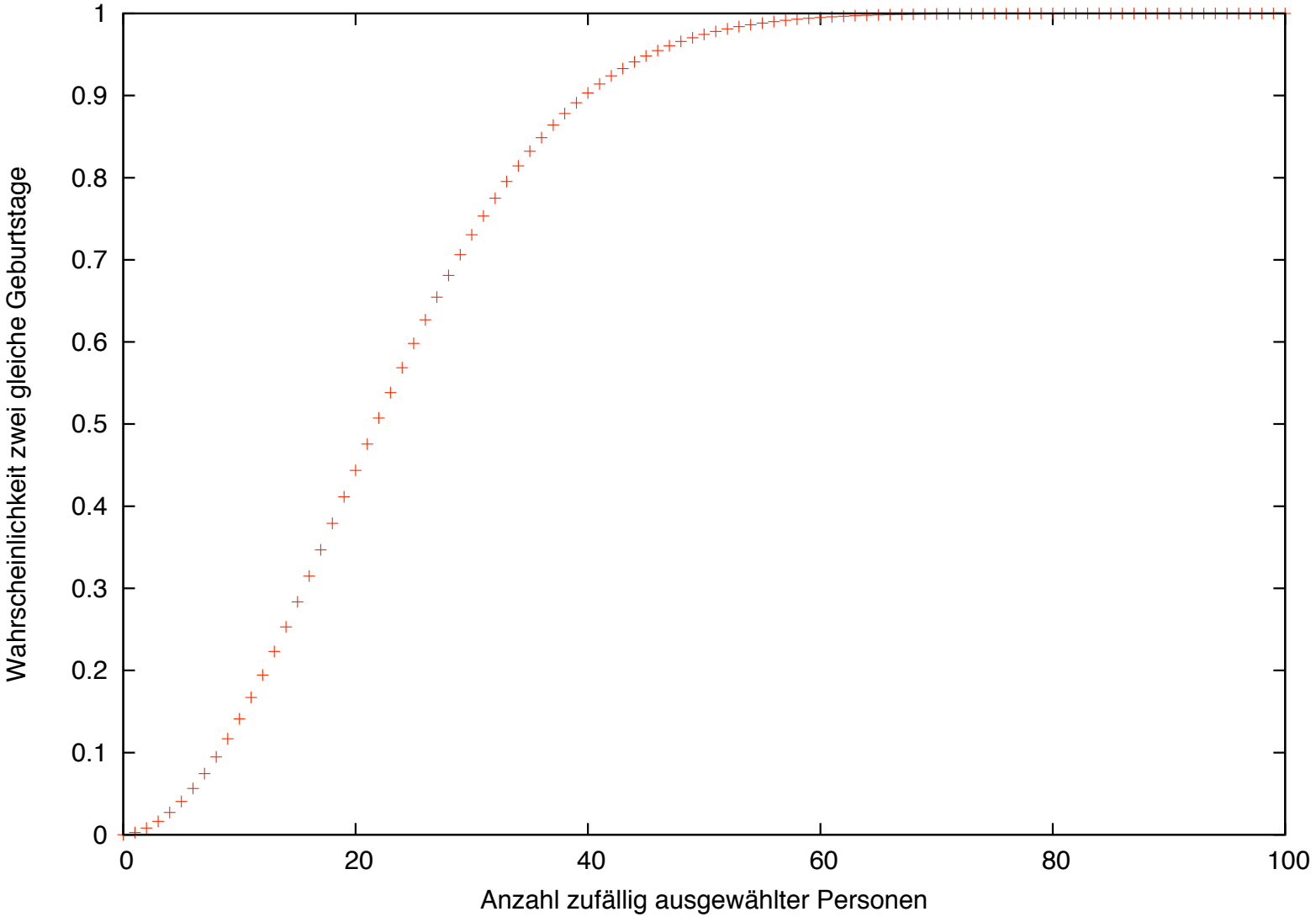
- Keine Analyse ohne Annahmen bekannt
- Intuitiver Zugang: Verallgemeinertes Geburtstagsparadoxon

Definition (Verallgemeinertes Geburtstagsproblem)

Wie viele ganze Zahlen müssen zufällig gewählt werden, damit $x \equiv y \pmod{p}$ für mindestens zwei der Zahlen mit Wahrscheinlichkeit $\geq \frac{1}{2}$ gilt?

- Wahrscheinlichkeit, dass q Zufallszahlen paarweise inkongruent modulo p sind: $\prod_{i=1}^{q-1} \left(1 - \frac{i}{p}\right) \approx exp(-\frac{q(q-1)}{2p})$
- Lösung von $exp(-\frac{q(q-1)}{2p}) = \frac{1}{2}$: $q = \frac{1}{2} + \sqrt{\frac{1}{4} + 2p \ln 2} \approx 1.18\sqrt{p}$

Geburtstagsparadoxon



Hintergrund von Pollards Analyse

- Sei $M := \{0, 1, \ldots, m-1\}$.
- Knuth untersuchte Folgen der Form $x_{i+1} = f(x_i)$ mit
  - zufälligem Startwert $x_0 \in M$ und
  - zufälliger Abbildung $f : M \to M$.
- Unter anderem zeigte er:
  - Mittlere Länge der Vorperiode: $\mu \approx \sqrt{\frac{\pi m}{8}} - \frac{2}{3} \approx 0.626657\sqrt{m}$
  - Mittlere Periodenlänge: $\lambda \approx \sqrt{\frac{\pi m}{8}} + \frac{1}{3} \approx 0.626657\sqrt{m}$

Pollards Analyse

- Sei $(\tilde{x}_i)$ die Folge der Rho-Methode modulo p.
- Form: $\tilde{x}_{i+1} = f(\tilde{x}_i) \bmod p$ mit ganzzahliger Polynomfunktion $f(x)$
- Annahme: $f(x) \mod p$ ist „zufällige“ Abbildung von $\mathbb{Z}_p$ in sich
- Gesucht: Mittelwert des kleinsten $k > 0$ mit $\tilde{x}_k = \tilde{x}_{2k}$
- Pollard zeigte unter der erwähnten Annahme:

$$k(p) \approx \frac{\pi^{\frac{5}{2}}}{12\sqrt{2}}\sqrt{p} \approx 1.030809\sqrt{p}$$

- Für lineare Polynomfunktionen ist die Annahme nicht plausibel.
- Pollard empfiehlt $f(x) = x^2 + c$ mit $c \neq 0, -2$.

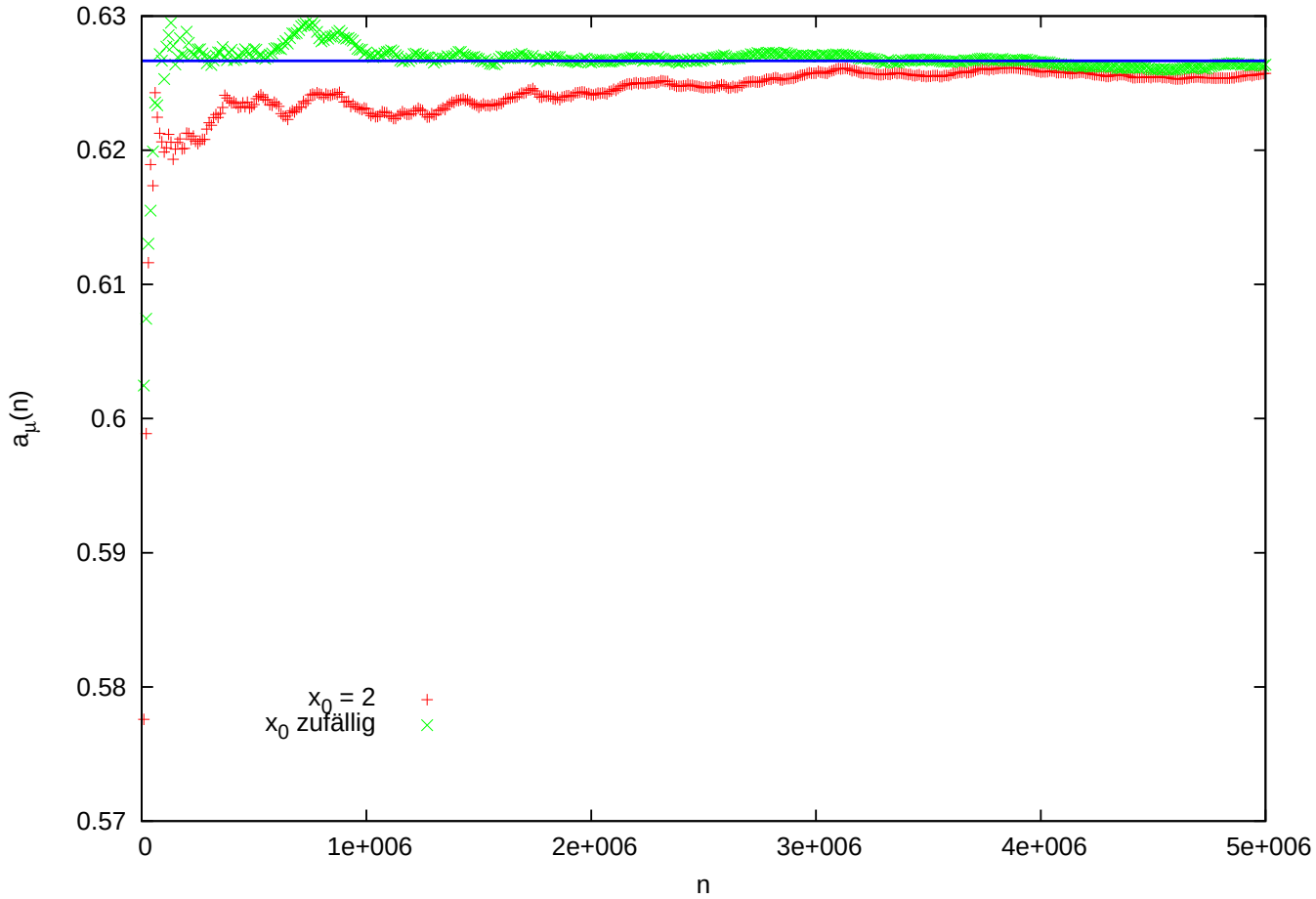
Übersicht

Experimente

- Folge der Form: $\tilde{x}_{i+1} = (\tilde{x}_i^2 + c) \bmod p$
- Eigenschaften von $(\tilde{x}_i)$:
  - Länge der Vorperiode: $\mu(p)$
  - Periodenlänge: $\lambda(p)$
  - Kleinstes $k > 0$ mit $\tilde{x}_k = \tilde{x}_{2k}$: $k(p)$
- Pollard untersuchte $(\tilde{x}_i)$ mit $\tilde{x}_0 = 2$ und $c = -1$ empirisch.

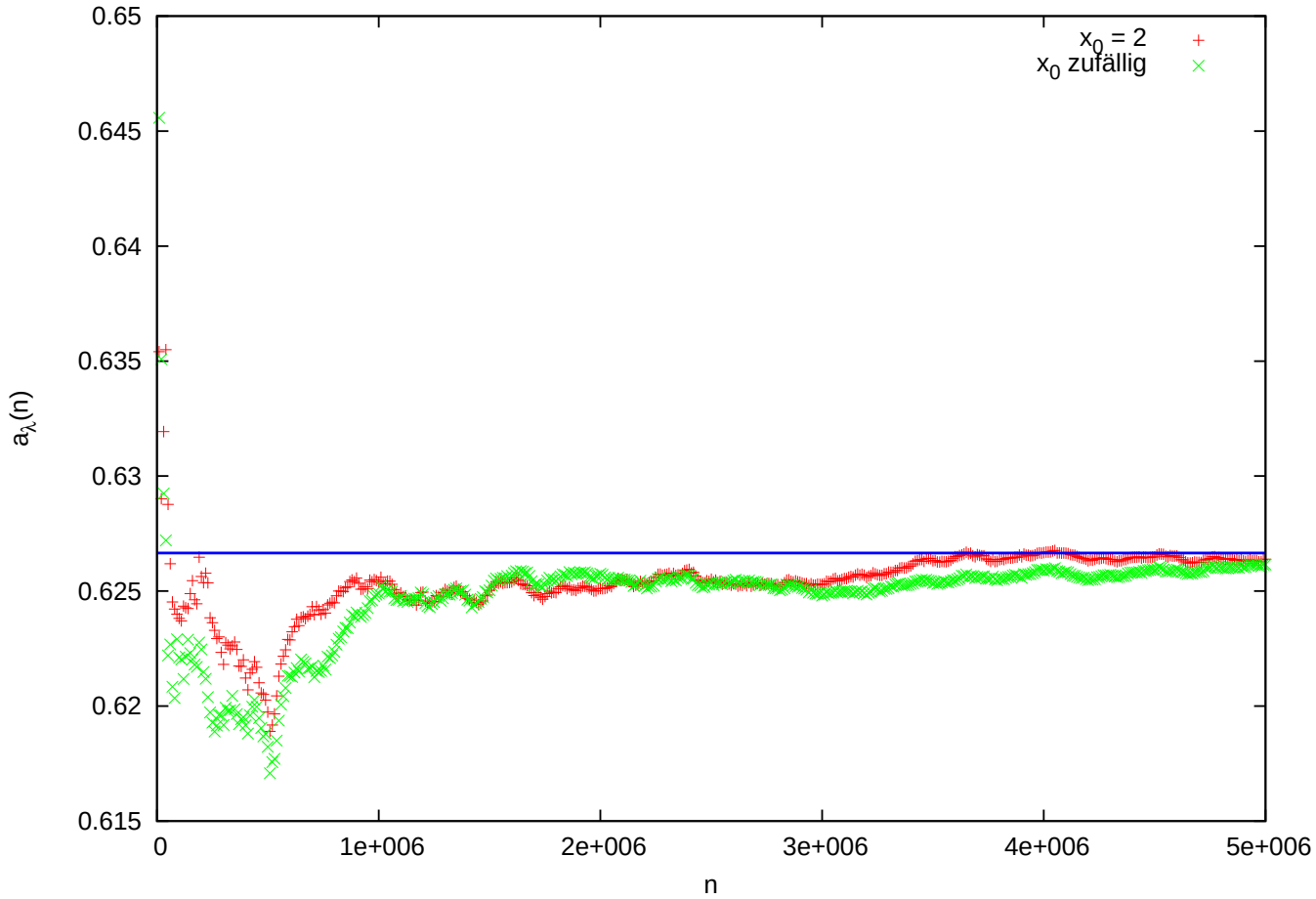
Die Funktion $a_\mu(n)$ für $c = 1$

$$a_\mu(n) = \frac{1}{|\{p \in \mathbb{P} : p \leq n\}|} \sum_{\substack{p \in \mathbb{P} \\ p \leq n}} \frac{\mu(p)}{\sqrt{p}}$$



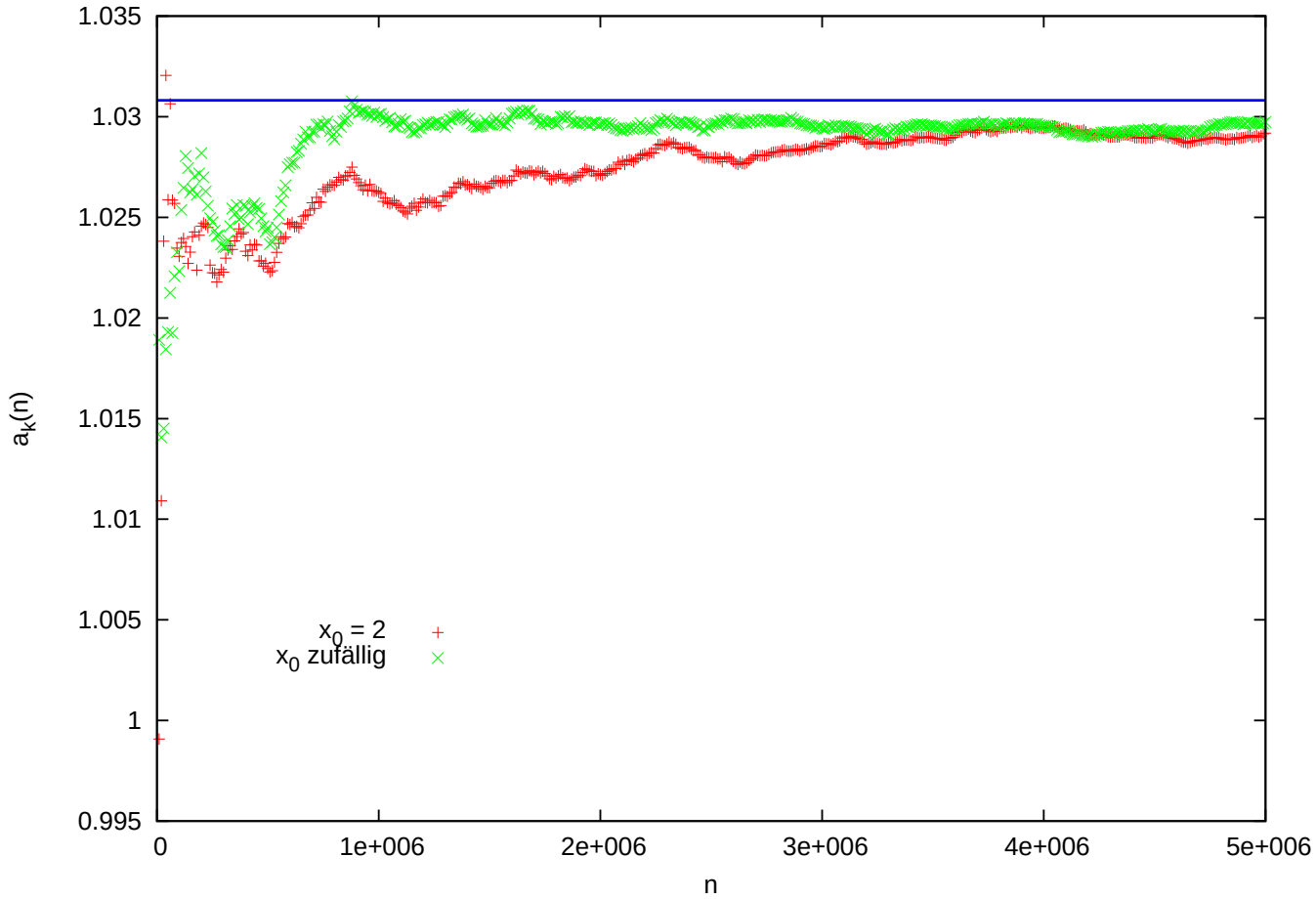
Die Funktion $a_\lambda(n)$ für $c = 1$

$$a_\lambda(n) = \frac{1}{|\{p \in \mathbb{P} : p \leq n\}|} \sum_{\substack{p \in \mathbb{P} \\ p \leq n}} \frac{\lambda(p)}{\sqrt{p}}$$



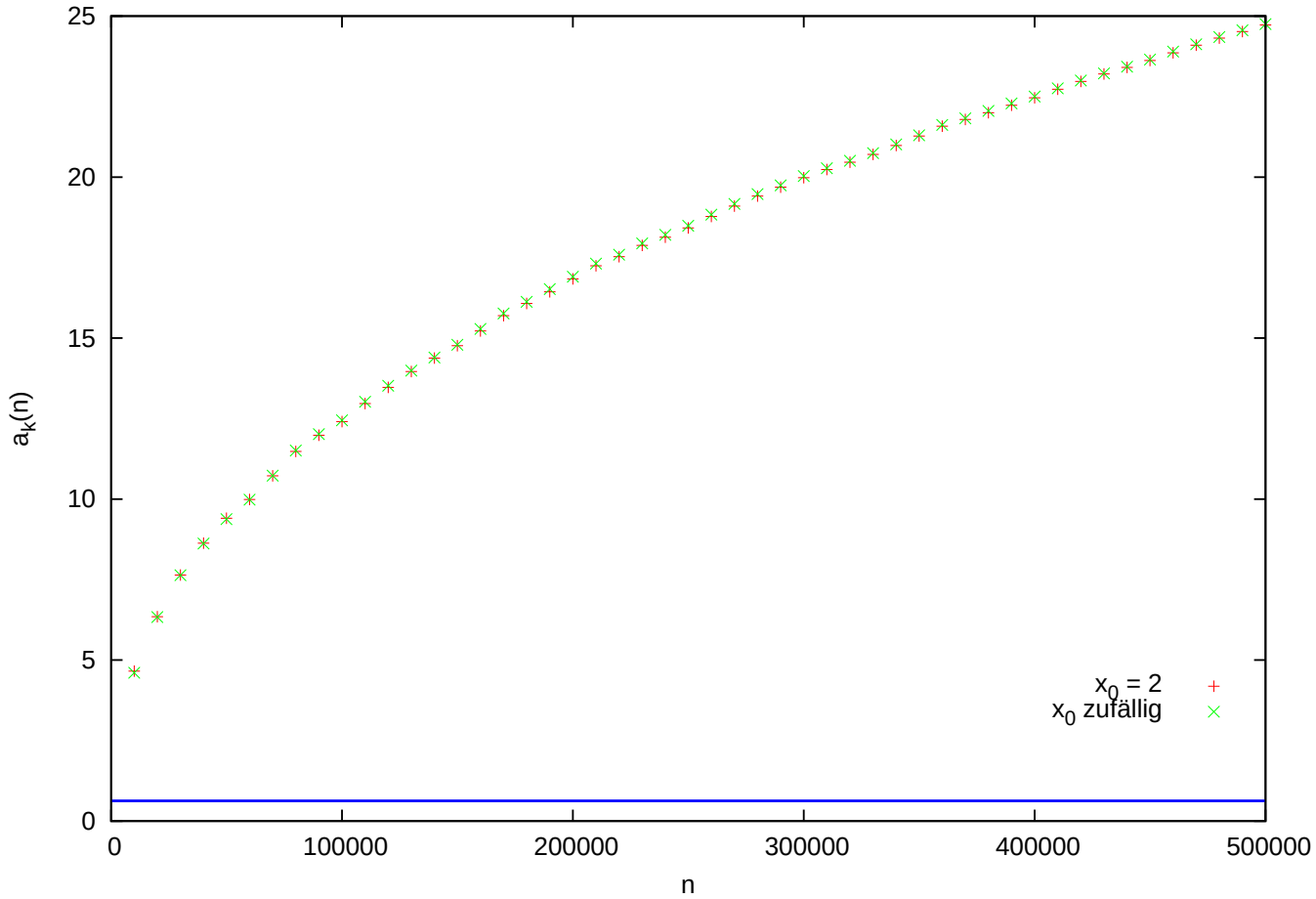
Die Funktion $a_k(n)$ für $c = 1$

$$a_k(n) = \frac{1}{|\{p \in \mathbb{P} : p \leq n\}|} \sum_{\substack{p \in \mathbb{P} \\ p \leq n}} \frac{k(p)}{\sqrt{p}}$$



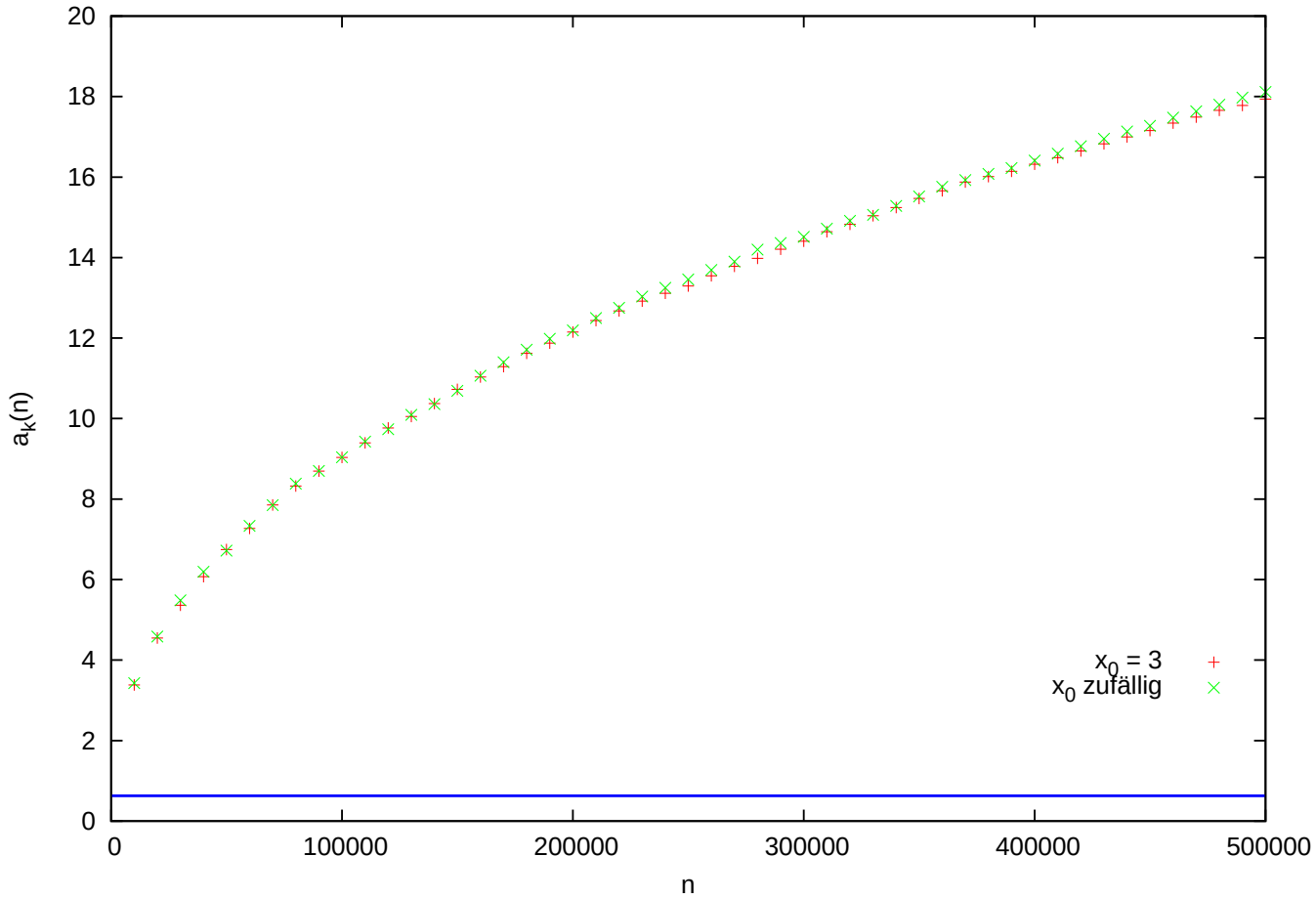
Die Funktion $a_k(n)$ für $c = 0$

$$a_k(n) = \frac{1}{|\{p \in \mathbb{P} : p \leq n\}|} \sum_{\substack{p \in \mathbb{P} \\ p \leq n}} \frac{k(p)}{\sqrt{p}}$$

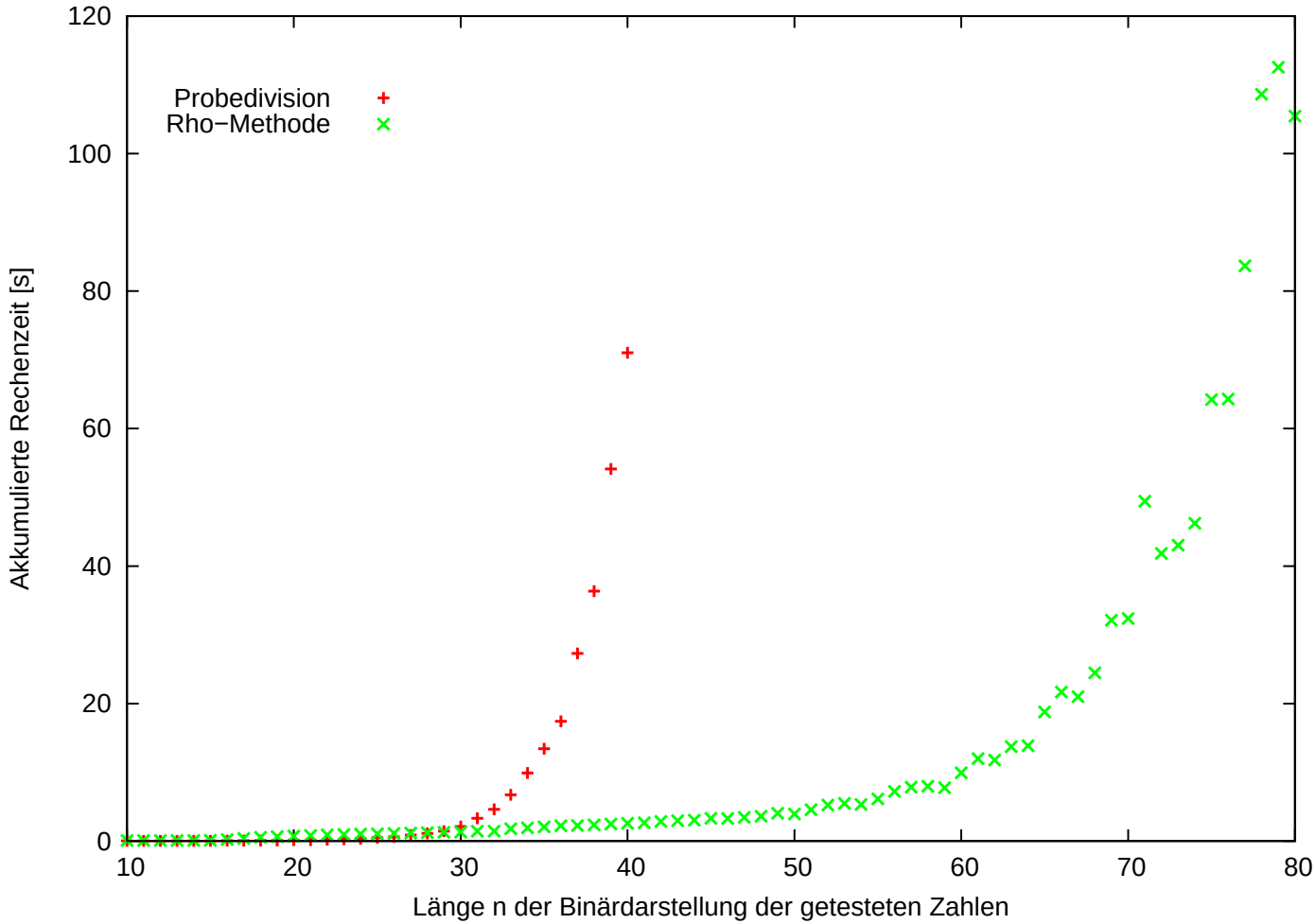


Die Funktion $a_k(n)$ für $c = -2$

$$a_k(n) = \frac{1}{|\{p \in \mathbb{P} : p \leq n\}|} \sum_{\substack{p \in \mathbb{P} \\ p \leq n}} \frac{k(p)}{\sqrt{p}}$$



Probedivision vs. Rho-Methode



Übersicht

Fazit

- Sehr gut zum Entfernen kleiner Primfaktoren
- Deutlich schneller als Faktorisierung durch Probedivision
- Kombination mit asymptotisch schnelleren Verfahren

Übersicht

Referenzen und weiterführende Literatur I

- John M. Pollard. A monte carlo method for factorization. *BIT Numerical Mathematics*, 15:331–334, 1975.
- Donald E. Knuth. *Seminumerical Algorithms*. Band 2 von *The Art of Computer Programming*. *Zweite Auflage, Addison-Wesley, 1981.*
- Otto Forster. *Algorithmische Zahlentheorie*. Vieweg, 1996.
- Hans Riesel. *Prime Numbers and Computer Methods for Factorization*. Birkhäuser, 1985.

Referenzen und weiterführende Literatur II

- Henri Cohen. *A course in computational algebraic number theory*. Springer, 1996.
- Richard P. Brent. An improved monte carlo factorization algorithm. *BIT Numerical Mathematics*, 20:176–184, 1980.
- Richard P. Brent und John M. Pollard. Factorization of the eighth fermat number. *Mathematics of Computation*, 36:627–630, 1981.
- R.L. Rivest und A. Shamir und L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21:120–126, 1978.
- George Marsaglia. Random numbers fall mainly in the planes. *Proceedings of the National Academy of Sciences*, 61:25–28, 1968.

Vielen Dank für Ihre Aufmerksamkeit!