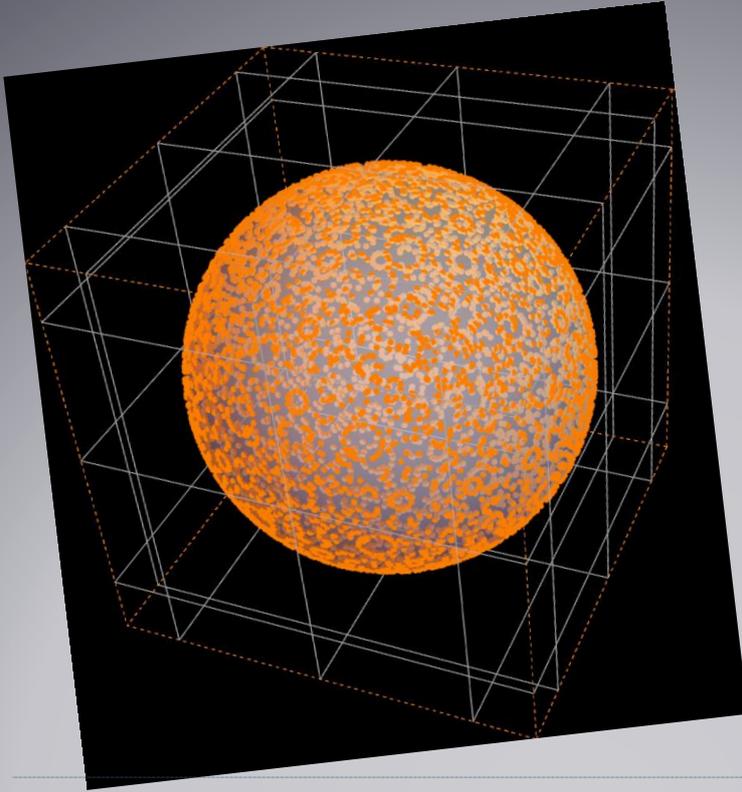


Karl-Heinz Kuhl



PRIMZAHLEN – ALTBEKANNTES UND NEUES

EIN STREIFZUG DURCH DIE LANDSCHAFT DER PRIMZAHLEN

Wundersame Eigenschaften und Einblicke. Nicht aus der Sicht eines Mathematikers, sondern eines Spaziergängers, der ein paar ‚Flecken‘ der Primzahl-Landschaft durchwandert und sich spielerisch und experimentierfreudig deren Geheimnissen nähert und ihre Faszination an interessierte Menschen weitergeben möchte.

Dritte, überarbeitete und aktualisierte Auflage (2020)

Primzahlen – Altbekanntes und Neues

Ein Streifzug durch die Landschaft der Primzahlen

Wundersame Eigenschaften und Einblicke.

Nicht aus der Sicht eines Mathematikers, sondern eines Spaziergängers, der ein paar ‚Flecken‘ der Primzahl-Landschaft durchwandert und sich spielerisch und experimentierfreudig deren Geheimnissen nähert und ihre Faszination an interessierte Menschen weitergeben möchte.

Dipl.-Phys. Karl-Heinz Kuhl

Parkstein, im Dezember 2020

$$1 + 2 + 3 + 4 + \dots = -\frac{1}{12}$$

(Ramanujan)

Web:

<http://yapps-arrgh.de>

(Yet another promising prime number source:
Amazing recent results from a guerilla hobbyist)

Hinweis für die aktuelle Online-Version (https://yapps-arrgh.de/Primzahlen_Online.pdf):

In der freien Online-Version wurden manche Textpassagen und Mathematica-Programme entfernt, die jedoch in der gedruckten Version und der E-Book-Version alle enthalten sind. Aktuelle Ergänzungen zum Buch findet man hier: https://yapps-arrgh.de/data/Primzahlen_Ergaenzungen.pdf.

Wer einen noch tieferen Einblick in die vielen Mathematica-Programme haben möchte, sei ermuntert, sich an den Autor zu wenden.

Kontakt: info@yapps-arrgh.de

Für Michèle

ISBN 978-3-939247-92-0

Verlag Eckhard Bodner, Pressath - 2017

Dritte, überarbeitete Auflage - Dezember 2020

Die Abbildung auf der Titelseite zeigt die Grafik aus Abbildung 82 im Kapitel 9.2.

Einbandentwurf: Karl-Heinz Kuhl

Urheberrechte: Dieses Werk und alle darin enthaltenen Abbildungen und Computerprogramme sind urheberrechtlich geschützt. Jede kommerzielle Verwertung, die nicht ausdrücklich vom Autor genehmigt wurde, ist verboten. Neue Methoden und Algorithmen sind (mit Datumsangabe) notariell geschützt.

Die Inhalte dieses Buches (bzw. der zum Download bereitgestellten Onlineversion inkl. der zugehörigen Dateien) dürfen für private oder nicht kommerzielle Zwecke verwendet, weitergegeben, im Internet dargestellt und in eigenen Publikationen verwendet werden, unter der Voraussetzung, dass alle Inhalte korrekt zitiert (Buchbezeichnung mit Datum und evtl. Auflagennummer, Namensangabe des Autors und Angabe der obigen Internetadresse), sowie alle Inhalte komplett und in unveränderter Form übernommen werden. Dies gilt für alle Texte, Grafiken und Computerprogramme und sonstigen Dateien. Insbesondere gilt für die in blauer Schrift dargestellten Teile, dass der Hinweis auf ‚Neues‘ mit enthalten sein sollte.

Haftung: Der Autor übernimmt keine Haftung für Schäden jedweder Art, die durch den Gebrauch der Mathematica-Programme in den Listen im Anhang sowie auf der beiliegenden CD entstehen können. Der Autor gibt außerdem keine Gewähr, dass alle Programme fehlerfrei sind und unter jeder Betriebssystem-Konfiguration funktionieren.

1 INHALT

1	Inhalt	3
2	Einleitung	8
2.1	Verwendete mathematische Symbole und Funktionen.....	10
3	Grundlegendes zu Primzahlen.....	14
3.1	Für Eilige: Was wissen wir sicher?.....	16
3.2	Für Eilige: Was vermuten wir, haben dafür jedoch keinen Beweis?	17
3.3	Für Eilige: Was ist noch unbekannt?	18
3.4	Für Eilige: Was ist neu?	19
4	Spezielle Arten von Primzahlen	20
4.1	Primzahl-Zwillinge.....	20
4.2	Primzahl-Drillinge und -Vierlinge	23
4.3	Primzahl N-Linge	25
4.4	Korrelationen der Endziffern in aufeinander folgenden Primzahlen	32
4.5	Mersenne-Primzahlen	34
4.5.1	GIMPS – Great Internet Mersenne Prime Search.....	39
4.6	Fermat-Primzahlen.....	40
4.7	Glückliche Primzahlen.....	42
4.8	Vollkommene Zahlen	44
4.8.1	Allgemeines und Definition	44
4.8.2	Eigenschaften	45
4.9	Sophie-Germain-Primzahlen	47
4.9.1	Berechnung und Eigenschaften	48
4.10	Fibonacci-Zahlen und andere rekursive Folgen	49
4.10.1	Lineare Rekursion: Ein mächtiges Instrument.....	53
4.10.2	Fibonacci-Primzahlen und Fibonacci Pseudo-Primzahlen.....	61
4.10.3	Meta-Fibonacci-Folgen	63
4.11	Carmichael- und Knödel-Zahlen.....	64
4.12	Mirp-Primzahlen.....	65
4.13	Wagstaff-Primzahlen	65
4.14	Wieferich-Primzahlen.....	67
4.15	Wilson-Primzahlen	70
4.16	Wolstenholme-Primzahlen.....	71
4.17	RG-zahlen (= rekursiv gödelisiert)	72
4.17.1	GOCRON-Typ 6 (,Prime-OCRONS')	72
4.17.2	GOCRON-Typ 4 (mit den Symbolen ,2','*','P','^')	76

5	Einschub: Die Riemannsche Zeta-Funktion $\zeta(s)$	79
5.1	Allgemeines.....	79
5.2	Die verschiedenen Darstellungen von $\zeta(s)$	85
5.3	Die Produktdarstellung von $\zeta(s)$ im Komplexen	87
5.4	Eine unerwartete Produktdarstellung einer etwas anderen $\zeta(s)$	93
5.5	Eine Zähl-Funktion für die Anzahl der Nullstellen	96
5.6	Die Zeta-Funktion und das Quantenchaos: Eine Brücke zur Physik.....	99
6	Einschub: Die Riemann-Funktion $R(s)$	103
7	Ein paar wichtige zahlentheoretische Funktionen	104
7.1	Die Omega-Funktionen: Anzahl der Primfaktoren.....	104
7.2	Die Liouville-Funktion.....	106
7.3	Die Tschebyschew-Funktion.....	108
7.4	Die Eulersche Phi-Funktion.....	111
7.4.1	Berechnung und graphische Darstellungen der Phi-Funktion	111
7.4.2	Eigenschaften der Phi-Funktion	113
7.5	Die Teilersummenfunktion (Sigma-Funktion).....	114
7.5.1	Definition, Eigenschaften.....	115
7.5.2	Graphische Darstellungen der Sigma-Funktion.....	118
7.6	Die Ramanujansche Tau-Funktion.....	122
7.7	Die Mertens-Funktion.....	126
7.8	Das Radikal.....	128
7.9	Ramanujan-Summen.....	129
7.9.1	Definition.....	130
7.9.2	Eigenschaften	134
7.9.3	Erweiterung auf \mathbb{R}	135
8	Funktionen zur Berechnung von Primzahlen.....	138
8.1	Funktionen, die exakt alle Primzahlen liefern.....	138
8.2	Funktionen, die immer Primzahlen liefern.....	139
8.3	Funktionen, deren positive Wertemengen die Menge der Primzahlen sind	139
8.4	Rekursive Formeln.....	140
8.5	Funktionen, deren ‚Nullstellen‘ oder Minima bei Primzahlen liegen.....	141
8.5.1	Eine Variante der der \mathfrak{J} -Funktion.....	141
8.5.2	Reed-Jameson-Funktion	142
8.5.3	Sonstige zahlentheoretische Funktionen mit Nullstellen bei Primzahlen ..	143
8.6	Formeln zur Berechnung der Anzahl von Primzahlen.....	144
8.7	Formeln zur Berechnung der n-ten Primzahl.....	150
8.8	Formeln zur Berechnung der n-ten Nicht-Primzahl.....	150

9	Jetzt wird's interessant: vierdimensionale Kugeln und Primzahlen	152
9.1	Zweite Dimension: Kreise und ganzzahlige Gitterpunkte	154
9.1.1	Formeln und Eigenschaften.....	157
9.2	Dritte Dimension: Kugeln und Gitterpunkte.....	159
9.2.1	Formeln und Eigenschaften.....	165
9.3	Vierte Dimension: Hyperkugeln und Gitterpunkte auf ‚Glomes‘	166
9.3.1	Formeln und Eigenschaften.....	174
10	Von OCRONs und GOCRONs: Gödel lässt grüßen.....	175
10.1	Was sind OCRONs bzw. GOCRONs?.....	175
10.1.1	Summendarstellung in Zahlensystemen.....	176
10.1.2	Produktdarstellung mit Primfaktoren.....	177
10.2	OCRONs mit Prim-Operator.....	179
10.2.1	OCRONs mit Prim-"P" und "*" -Operator.....	180
10.2.2	OCRONs mit Prim-"P", "*" - und "^" -Operator.....	182
10.2.3	OCRONs mit Prim-"P", "*" - "^" - und "Q" -Operator.....	200
10.2.4	OCRONs mit Prim- und Nicht-Prim-Operator.....	200
10.3	Die Welt der OCRON-Wesen und mathematischer Sprengstoff.....	206
11	Primzahlen und die „Matrix“-Software: Eine Gesetzmäßigkeit?	212
11.1	Regeln für Differenzen n-ter Ordnung.....	212
12	Die ABC-Vermutung.....	221
12.1	Allgemeines.....	221
12.2	ABC-Vermutung und GOCRONs: Eine Verbindung?.....	225
12.3	Die Menge <i>Mabc</i> und ihre Ebenengleichung.....	230
13	Primzahlen in den Naturwissenschaften.....	234
13.1	Primzahlen im DNA-Code.....	234
13.2	Spektrale Eigenschaften von ‚Primzahl-Signalen‘.....	237
14	Primzahlen und Online-Banking.....	239
14.1	Die RSA-Verschlüsselung.....	239
14.2	Die Sicherheit der RSA-Methode.....	245
14.3	Rechenbeispiele zur RSA-Ver- und Entschlüsselung.....	246
15	Primzahlen in der Musik.....	250
15.1	Eulers Konsonanztheorie und der Gradus Suavitatis.....	250
15.1.1	Mathematische Eigenschaften des Gradus Suavitatis.....	254
15.1.2	Das ‚Zurechthören‘ von komplexen oder irrationalen Intervallen.....	255
15.2	Primzahlen als rhythmische Muster.....	256
16	Primzahlen in der Poesie	259
16.1	Haikus und Tankas.....	259

16.2	Sestine.....	261
16.3	Zum Nachdenken.....	265
17	Primzahlen und außerirdische Lebensformen.....	267
17.1	Die Arecibo-Botschaft.....	269
18	Sonstiges.....	271
18.1	Die Zahl 12	271
18.2	Die Zahl 313.....	272
18.3	Primzahlen und Kunst.....	273
19	Schlussbetrachtung.....	274
20	Anhang.....	275
20.1	Die Catalansche Vermutung.....	275
20.2	Statistische Auffälligkeiten der Endziffern in der Primzahlfolge	275
20.3	Eine interessante Folge: Die Perrin-Folge.....	277
20.4	Weitere Vermutungen über Primzahlen.....	279
20.5	Primzahl n-Tupel: Konstellationen von Primzahlen	280
20.6	Explizite Lösungen aus Kapitel 4.10.1.....	282
20.7	Weitere Abbildungen zu RG-Folgen	283
20.8	Virtuelle OCRONS	286
20.9	Weitere ungelöste mathematische Probleme.....	290
20.9.1	Euklid-Mullin Sequenz.....	290
20.9.2	Aliquot-Sequenzen.....	291
20.9.3	Faktorisierung von Zahlen.....	309
20.10	Tabellen.....	315
20.10.1	Anzahl der Primzahlen bis zu einer Grenze N : $\pi(n)$	315
20.10.2	Mersenne-Primzahlen.....	319
20.10.3	Fermat-Primzahlen.....	320
20.10.4	Entartung von Typ4-OCRONS und -EOCRONS	321
20.10.5	Nullstellen der Ramanujanschen Tau-L-Funktion.....	323
20.10.6	abc-Vermutung: Fitparameter und C3-Werte von Ebenengleichungen verschiedener Gödelisierungsmethoden	325
20.10.7	Reed Jameson Pseudo-Primzahlen	328
20.11	Mathematica-Programme.....	328
20.11.1	Vergleich der Anzahl Von Primzahl-Zwillingen, -Cousins und Sexy-Primzahlen mit der Formel von Hardy-Littlewood.....	329
20.11.2	RG-Folgen.....	329
20.11.3	Riemannsche Zeta-Funktion.....	329
20.11.4	Reed Jameson und Perrin-Folgen.....	330

20.11.5	Gitterpunkte auf n-Spheres (n-dimensionalen Kugeln).....	330
20.11.6	Auswertung und Statistik für Primzahl-Differenzen.....	334
20.11.7	abc-Vermutung	335
20.11.8	Sonstige Mathematica-Programme	335
20.11.9	OCRONs- und abc-Vermutung: Programm-Bibliothek.....	337
20.11.10	Sound-Routinen.....	338
20.11.11	RSA-Verschlüsselung und Entschlüsselung	338
20.11.12	Aliquot-Sequenzen	341
20.11.13	Die Arecibo-Nachricht.....	342
20.11.14	Korrelationen in den letzten Ziffern der Primzahlfolge	343
20.11.15	Primzahl N-Linge und maximale Primzahldichte	343
	Literaturverzeichnis	343
	Abbildungsverzeichnis	344
	Tabellenverzeichnis.....	350
	Sachverzeichnis	351
	Computerprogramme – CD.....	355
	Animationen.....	355
	Mathematica-Notebooks.....	355
	Sounds	356
	Grafiken.....	356
	Danksagungen.....	357

2 EINLEITUNG

Primzahlen – Kaum ein anderer Begriff aus der Mathematik hat so viele Menschen derart fasziniert und inspiriert. Sie stehen im scheinbaren Kontrast zu den üblichen Eigenschaften, die man den mathematischen ‚Dingen‘ zuordnet: Geordnet zu sein, berechenbar zu sein...

Die Primzahlen scheinen keiner Gesetzmäßigkeit zu folgen, sie scheinen einfach zufällig und planlos zwischen den anderen natürlichen Zahlen zu sitzen. Man hat den Eindruck, der ‚Liebe Gott‘ hat bei der Erschaffung der Zahlen die Primzahlen einfach zufällig ‚ausgeschüttet‘, wie wild wachsendes Unkraut zwischen den natürlichen Zahlen. Die Sprache mancher Mathematiker wird zuweilen ungewohnt blumig und poetisch, wenn es um Primzahlen und deren verwandte Funktionen geht: Statt ‚Unkraut‘ findet man gelegentlich auch Begriffe wie ‚Perlen‘ oder ‚Edelsteine‘ (eine Anspielung auf die Tatsache, dass sehr große Primzahlen ähnlich schwer zu finden sind wie Edelsteine). Die mit den Primzahlen eng verwandte Zeta-Funktion (Kapitel 1) im Bereich der komplexen Zahlen wird gelegentlich mit einer ‚Landschaft‘ verglichen, die es zu erforschen gilt.

Es ist dieses gewisse Maß an Chaos und Unergründlichkeit, das die Primzahlen so interessant macht. Obwohl Primzahlen seit Menschengedenken einen großen Reiz auf Menschen ausübten, sind zahlreiche die Primzahlen betreffende Fragen auch heute noch ungeklärt, und das trotz der Bemühungen der genialsten und besten Mathematiker die gelebt haben und noch leben!

Die Menge der Bücher über Primzahlen, die heutzutage erhältlich sind, ist inzwischen sehr groß geworden. Dabei gibt es fast ausschließlich zwei Arten von Büchern: Die populärwissenschaftlichen Bücher, die so gut wie keine mathematischen Formeln enthalten, und die mathematisch-wissenschaftlichen Fachbücher, die zum großen Teil in mathematischer Sprache geschrieben sind und natürlich überwiegend aus mathematischen Herleitungen, Beweisen und Formeln bestehen, die selbst für den ambitionierten Hobby-Mathematiker nur sehr schwer zu lesen sind.

Dieses Buch versucht, einen anderen Zugang zur Mathematik zu vermitteln: Die Sprache ist möglichst einfach und verständlich gehalten. Der Leser wird darin nur ganz wenige Beweise finden. Auf Formeln und graphische Darstellungen wurde jedoch nicht verzichtet. Im Gegenteil: Es enthält sehr viele Formeln und Bilder. Der Grund ist ganz einfach: Mathematische Formeln besitzen, selbst wenn sie nicht immer vom Leser verstanden werden, eine gewisse Ästhetik und Anziehungskraft, die ihn neugierig machen und vielleicht auch einen Impuls setzen können, manche Gebiete weiter zu vertiefen. Mit den vielen graphischen Darstellungen verhält es sich nicht anders: Ein Bild kann mehr sagen als 1000 Worte. Der Autor wagt die Hypothese, dass ein Zugang zur Ästhetik der Mathematik auch möglich ist ohne in die letzten Tiefen der mathematischen Gedankengänge einzutauchen.

Der Autor gibt sich meist keine Mühe, mathematische Beweise oder Sätze nachzuvollziehen oder gar neue Gesetze streng mathematisch herzuleiten. Vielmehr betrachtet er die Mathematik und hier speziell die Mathematik der Primzahlen als eine große ‚Spielwiese‘ auf der man nach Herzenslust herumspazieren kann und einfache Experimente betreiben kann. Diese genügen selbstverständlich nicht den strengen Maßstäben, die die mathematische Gemeinde normalerweise von den Arbeiten ihrer Gilde verlangt. Bei

manchen Mathematikern mag diese Vorgehensweise ein gewisses Unbehagen hervorrufen. Es ist jedoch eine Methode, die Schönheiten der Mathematik auch Menschen, die keine ausgeprägte mathematische Vorbildung besitzen, nahezubringen, ähnlich wie man ein Konzert von J. S. Bach genießen kann, ohne die musikalischen Strukturen des Stückes vorher musiktheoretisch analysiert zu haben.

Staunen lässt einen die Tatsache, dass bei der Erforschung der Primzahlen plötzlich Brücken zwischen weit entfernten Gebieten der Mathematik, ja sogar hin zur Physik entstehen.

Ganz ohne mathematische Vorkenntnisse wird es jedoch nicht gehen. Abiturwissen im Fach Mathematik wäre zumindest in einigen Kapiteln nützlich. Aber auch ohne diese im Einzelnen zu verstehen kann der Leser die Ergebnisse (die oft in Form von Bildern und graphischen Darstellungen präsentiert werden) genießen...

Lange Zeit galt die Erforschung der Primzahlen als mathematische Grundlagenforschung, die keinen praktischen Nutzen für die Menschen hat. Dies hat sich in den letzten Jahren grundlegend geändert durch die Notwendigkeit von Verschlüsselungsmethoden beim Datenverkehr im Internet. Diese Verschlüsselungsmethoden basieren auf den Eigenschaften sehr großer Primzahlen (bzw. Eigenschaften sehr großer Zahlen, die aus wenigen großen Primzahlen zusammengesetzt sind). Näheres dazu im Kapitel ‚Primzahlen und Online-Banking‘.

Diese Arbeit deckt natürlich nicht alle Gebiete, die Primzahlen betreffen, ab. Es werden nicht einmal alle diesbezüglichen Themen, die in Frage kommen, erwähnt. Der Autor trifft eine Auswahl von Fragestellungen, die ihm interessant erscheinen und beschäftigt sich ausschließlich mit diesen Themen. Die meisten der hier behandelten Themen sind in zahlreichen Büchern der einschlägigen Fachliteratur, Veröffentlichungen sowie im Internet zu finden. Diese Arbeit enthält somit zum großen Teil eine Zusammenfassung dieser wichtigen, bekannten Sätze und Ausführungen, die zum Teil auch für das Verständnis der ausführlicheren Teile der Arbeit nützlich sind. Diese Teile der Arbeit sind daher mehr im Stil einer ‚Formelsammlung‘ gehalten. Die meisten dieser ‚herausgepickten‘ Themen werden auf den Internetseiten <https://en.wikipedia.org> und <http://mathworld.wolfram.com> ausführlich behandelt.

Dieses Buch wäre ohne die Software ‚Mathematica‘¹ nicht möglich gewesen. Die zahlreichen Illustrationen und Formeln sind zum großen Teil mit Hilfe dieser Software entstanden. Wer dieses Programm besitzt, sei ermuntert, mit den vielen Programm-Beispielen zu experimentieren. Die Programm-Codes können einfach in ein Mathematica-Notebook kopiert und dann ausgeführt werden.

Der Autor hat sich Mühe gegeben, möglichst viele Quellen zu zitieren. Um Missverständnissen wegen vergessenen Zitaten oder Quellenangaben vorzubeugen, gilt in dieser Arbeit folgende Konvention: Sämtlicher Text in normaler, schwarzer Schrift betrifft Teile, die bereits an anderer Stelle (von anderen Autoren) veröffentlicht und dargestellt wurden (sei es im Internet oder in der Fachliteratur). Der in schwarz gehaltene Text ist somit größtenteils eine Zusammenfassung der (dem Autor interessant erscheinenden Texte) aus den einschlägig bekannten Quellen im Internet. Dem Autor sei verziehen, wenn nicht immer alle diese Quellen angegeben wurden. Im Internet-Zeitalter

¹ Mathematica: <https://www.wolfram.com/mathematica>

mit seinen mächtigen Suchmaschinen dürften jedoch trotzdem alle Quellen in Sekundenschnelle zu finden sein.

Themen bzw. Formeln, die in der Literatur noch nicht behandelt wurden (oder dem Autor nicht bekannt sind), sowie neue Vermutungen und Erkenntnisse werden in blauer Schrift dargestellt.

Dem Autor ist bewusst, dass der Begriff ‚Neues‘ im Titel dieser Arbeit ein gewisses ‚Haltbarkeitsdatum‘ besitzt. Was heute noch neu ist, kann in ein paar Jahren schon ‚ein alter Hut‘ sein. Wo immer es ging, ist bei wichtigen Vermutungen und Aussagen deshalb eine ‚Zeitmarke‘ mit eingefügt worden.

Im Anhang findet der Leser zur Vertiefung zusätzlich geeignete Fachliteratur (siehe Literaturverzeichnis).

2.1 VERWENDETE MATHEMATISCHE SYMBOLE UND FUNKTIONEN

In dieser Arbeit werden neben den mathematischen Standard-Symbolen und elementaren Funktionen folgende mathematische Symbole, Funktionsnamen und Abkürzungen benutzt:

Mengen:

$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$: Mengen der natürlichen, ganzen, reellen sowie komplexen Zahlen

\mathbb{P} : Menge der Primzahlen

Operatoren und Symbole:

$O(\dots)$: Weitere Restglieder in der Größenordnung von (\dots)

Σ : Das Summen-Symbol

\prod : Das Produkt-Symbol

p_n : Die n -te Primzahl

$n!$: Das Fakultäts-Symbol

$p\#$: Produkt aller Primzahlen $p_1 \cdot p_2 \cdot p_3 \cdot \dots \cdot p_n$ bis $p_n = p$

F_n : Die n -te Fermat-Zahl

M_n : Die n -te Mersenne-Primzahl

ρ_n : n -te Nullstelle der Zetafunktion auf der ‚kritischen‘ Geraden

γ : Mascheroni-Konstante (auch Euler-Konstante): 0.57721566 ...

B : Bruns Konstante (Summe der inversen Primzahlzwillinge):
1.90216054

Π_2 : Primzahlzwillingskonstante: 0.6601618158

∞ :	Das Unendlichkeitssymbol
$\lfloor x \rfloor$:	Identisch mit $\text{floor}(x)$. Nimmt den ganzzahligen Anteil von x .
$\binom{n}{k}$:	Binomialkoeffizienten
(n, m) :	größter gemeinsamer Teiler (auch: $\text{ggT}(n, m)$ oder $\text{gcd}(n, m)$)
$\text{ggT}(n, m)$:	siehe (n, m)
$\text{kgV}(n, m)$:	kleinstes gemeinsames Vielfaches, auch $\text{lcm}(n, m)$

Funktionen $f(n)$:

$\mu(n)$:	Möbius-Funktion
	$\mu(n) = \begin{cases} (-1)^k & \text{wenn } n \text{ quadratfrei, } k: \text{ Anz. Primfaktoren} \\ 0 & \text{sonst} \end{cases}$
$M(n)$:	Mertensfunktion (Summation über die Möbius-Funktion)
$\Lambda(n)$:	Von Mangoldt-Funktion
	$\Lambda(n) = \begin{cases} \ln(p) & \text{wenn } n = p^k, p \text{ prim und } k > 0 \\ 0 & \text{sonst} \end{cases}$
$\varphi(n)$:	Eulersche Phi-Funktion (engl.: Totient), gibt an, wie viele zu n teilerfremde natürliche Zahlen es gibt, die kleiner gleich n sind
$\Phi(n)$:	Summationsfunktion über $\varphi(n)$
$\sigma_k(n)$:	Summe der k -ten Potenzen der positiven Teiler von n
$\sigma(n)$:	$= \sigma_1(n)$ (wird allgemein Sigma-Funktion genannt)
$s(n)$:	Aliquot-Summe: Summe der Teiler (ohne n), $s(n) = \sigma_1(n) - n$
$r_k(n)$:	Anzahl der Darstellungen von n als Summe von k Quadraten
$r(n)$:	$= r_2(n)$ (Anzahl der 2-dim. Gitterpunkte auf Kreis mit Radius n)
$r_4(n)$:	$= 8 \sigma(n) - 32 \sigma\left(\frac{n}{4}\right)$, wobei $\left(\frac{n}{4}\right) 0$ sein soll, wenn $4 \nmid n$
	Anzahl der 4-dimensionalen Gitterpunkte einer 4-dim. Kugel mit Rad. n
$\tau(n)$:	Ramanujansche Tau-Funktion
$c_q(n)$:	Ramanujan-Summen
\mathcal{F}_n :	Farey-Sequenz der Ordnung n
$\omega(n)$:	Anzahl der unterschiedlichen Primfaktoren einer Zahl n

Verwendete mathematische Symbole und Funktionen

$\Omega(n)$: Anzahl der Primfaktoren einer Zahl n

Funktionen $f(x)$:

$\pi(x)$: Primzahlzählfunktion: Gibt die Anzahl der Primzahlen bis x an

$\pi_2(x)$: Gibt die Anzahl der Primzahlzwillinge bis x an

$\pi_3(x), \pi_4(x)$: Gibt die Anzahl der Primzahl-Zwillinge bzw. -Drillinge bis x an

$\pi_n(x)$: Gibt die Anzahl der Primzahl n -Tupel bis x an

$\pi'_n(x)$: Gibt die Anzahl der Primzahl-Paare mit Abstand n bis x an

$\pi_0(x)$: wie $\pi(x)$, jedoch Unterschied wenn x eine Primzahl wird:

$$\pi_0(x) = \lim_{\varepsilon \rightarrow 0} \frac{\pi(x-\varepsilon) + \pi(x+\varepsilon)}{2} \quad \text{oder: } \pi_0(p) = \pi(p) - \frac{1}{2}$$

$\Theta(x), \vartheta(x)$: 1. Tschebyschew Funktion: $= \sum_{p \leq x} \ln(p)$ (Summe über Logarithmen aller Primzahlen $\leq n$)

$\psi(x)$: Tschebyschew Psi-Funktion: Summenfunktion der Von-Mangoldt-Funktion $\psi(x) = \sum_{p^k \leq x} \ln(p) = \sum_{n \leq x} \Lambda(n)$ (2. Tschebyschew-Funk.)

$\psi_0(x)$: wie $\psi(x)$, jedoch unterschiedlich wenn x eine Primzahl wird:

$$\psi_0(x) = \lim_{\varepsilon \rightarrow 0} \frac{\psi(x-\varepsilon) + \psi(x+\varepsilon)}{2}$$

$\zeta(s)$: Riemannsche Zetafunktion

$P(s)$: Prim-Zetafunktion

$\xi(s)$: Variante der Riemannsche Zetafunktion (hat auf der kritischen Geraden die gleichen Nullstellen wie $\zeta(s)$, jedoch reelle Funktionswerte)

$\Gamma(s)$: Gammafunktion

$R(x)$: Riemann-Funktion

$\ln(x), \text{Li}(x)$: Natürlicher Logarithmus, Integral-Logarithmus

$\text{Ei}(x)$: Integralexponentialfunktion

$E_n(x)$: Exponential-Integralfunktion n -ter Ordnung

$Z(t), \vartheta(t)$: Riemann-Siegel Funktionen

$L(s)$: Ramanujan-Tau-Dirichlet-L-Funktion

$Z(t)$: Ramanujan-Tau-Z-Funktion

$\Theta(t)$: Ramanujan-Tau-Theta-Funktion

$\text{rad}(n)$: Radikal: Produkt unterschiedlicher Primfaktoren

$$\text{rad}(n) = \prod_{\substack{p|n \\ p \in \mathbb{P}}} p$$

$\zeta(p)$: Z-Funktion

Sonstige Abkürzungen:

o : OCRON

$g(o)$: Gödel-Nummer eines OCRONs

OEIS: Online Encyclopedia of Integer Sequences (<http://oeis.org>)

OCRON: ‚Operator Chain Representation Of Number‘

GOCRON: ‚Gödelized Operator Chain Representation of Number‘

EOCRON: ‚Enhanced‘ OCRON, auch EOCRON4, EOCRON6... (Typen)

EGOCRON: ‚Enhanced‘ GOCRON, auch EGOCRON4, EGOCRON6... (Typen)

RG-Zahlen: Durch rekursive Anwendung des Algorithmus, der zu Gödelnummern führt, entstandene Folge

3 GRUNDLEGENDES ZU PRIMZAHLEN

Hier zunächst einige grundlegende wichtige Aussagen über Primzahlen, die in jedem Mathematik-Lehrbuch für Anfänger nachgelesen werden können:

- Eine Primzahl ist eine natürliche Zahl größer als 1, die exakt zwei natürliche Zahlen als Teiler hat, nämlich die ,1‘ und die Zahl selber. Primzahlen sind, um die Umgangssprache zu benutzen: nicht teilbar.
- Die ersten Primzahlen lauten: 2,3,5,7,11,13,17,19, ... usw. Die Folge der Primzahlen beginnt bei 2 und nicht bei der 1.
- Die Primzahlen werden immer seltener, je höher wir im Zahlenbereich hinaufsteigen.² Das wirft die Frage auf, ob es irgendwann eine allerletzte, größte Primzahl gibt. Jedoch hat schon der griechische Mathematiker Euklid vor mehr als 2000 Jahren bewiesen:
- Es gibt unendlich viele Primzahlen. Der Beweis von Euklid ist so einfach, dass er hier skizziert sei: Nehmen wir an, es gäbe eine größte Primzahl p_n . Dann bilden wir das Produkt aus allen n Primzahlen und addieren 1 dazu:

$$N = p_1 \cdot p_2 \cdot p_3 \cdot \dots \cdot p_n + 1$$

N ist selbstverständlich viel größer als p_n und muss deshalb nach unserer Annahme teilbar sein. Nach kurzem Nachdenken kommt man jedoch darauf, dass N weder durch 2 noch durch 3 noch durch 5 usw. noch durch p_n teilbar sein kann. N kann durch keine einzige der Primzahlen p_1 bis p_n teilbar sein. Folglich muss N entweder eine Primzahl sein oder durch eine Primzahl $p > p_n$ teilbar sein, was jedoch ein Widerspruch zur Annahme ist. Folglich ist unsere Annahme der Existenz einer größten Primzahl p_n falsch!

Die Menge der Primzahlen \mathbb{P} lässt sich problemlos für die Gaußschen komplexen Zahlen erweitern. Häufig nennt man den Begriff ,Primelemente‘ als Verallgemeinerung des Begriffes ,Primzahlen‘.

In ein Buch über Primzahlen gehören ein paar Listen mit Primzahlen (hier generiert mit Mathematica):

```
If[PrimeQ[#], Framed[#], #] & /@Range[1, 100]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}
```

² Siehe Tabelle ,Anzahl der Primzahlen...‘ im Anhang

3.1 FÜR EILIGE: WAS WISSEN WIR SICHER?

Hier in verkürzter Form die wichtigsten Sätze über Primzahlen und damit in Zusammenhang stehende zahlentheoretische Funktionen, die bewiesen sind:

1. Es gibt unendlich viele Primzahlen.
2. Jede Zahl, die zusammengesetzt ist (also keine Primzahl) lässt sich eindeutig als Produkt von mindestens zwei Primzahlen schreiben.
3. Die Anzahl der Primzahlen $\pi(n)$ bezeichnet die Anzahl von Primzahlen die bis zu einem Wert n existieren. Für $\pi(n)$ gibt es viele unterschiedlich gute Abschätzungen, die es erlauben $\pi(n)$ näherungsweise zu berechnen. Es gibt auch exakte Formeln für $\pi(n)$ (siehe: 1.1).
4. Auch für die n -te Primzahl gibt es Formeln, die eine näherungsweise Berechnung ermöglichen, jedoch auch exakte Formeln (siehe Formeln zur Berechnung der n -ten Primzahl).
5. Es gibt beliebig große ‚Lücken‘ zwischen zwei benachbarten Primzahlen. Die größte, derzeit bekannte Lücke umfasst einen Bereich von 3.311.852 zusammengesetzten aufeinander folgender Zahlen (Stand Nov. 2016).
6. Die Summe der Kehrwerte der Primzahlen divergiert (geht gegen unendlich).
7. Die größte, gegenwärtig bekannte Primzahl lautet: $2^{82589933} - 1$. Sie besitzt 24862047 Stellen, wenn sie im Dezimalsystem ausgeschrieben wird. (Stand Dezember 2020).
8. Es gibt keine arithmetische Zahlenfolge, die nur Primzahlen liefert, wie z. B. die Formel von L. Euler $n^2 + n + 41$, die für $0 \leq n < 40$ ausschliesslich Primzahlen liefert, nicht jedoch für $n = 40$! Es gilt jedoch: Viele arithmetische Zahlenfolgen liefern (unter anderem) unendlich viele Primzahlen.
9. Es sind z. Zt. 51 Mersenne-Primzahlen bekannt. Die ersten Mersenne Primzahl-Exponenten lauten:
2, 3, 5, 7, 13, 17, 19, 31 (Folge [A000043](#) in OEIS). Stand: Dez. 2020.
10. Wenn M_p eine Primzahl ist, dann ist auch p eine Primzahl.
11. Es sind z. Zt. fünf Fermat Primzahlen $F_n = 2^{2^n} + 1$ bekannt (von $n = 0$ bis $n = 4$). Diese lauten:
3, 5, 17, 257, 65537 (Folge [A000215](#) in OEIS). Stand: November 2016.
 F_5 bis F_{32} sind zusammengesetzte Zahlen. F_{33} ist die erste Fermat-Zahl von der man nicht weiß, ob sie zusammengesetzt oder prim ist. Stand: November 2016.
12. Jede gerade vollkommene Zahl N (d. h. die Summe seiner positiven Teiler ohne N selbst ergibt N) hat die Form $2^{n-1}(2^n - 1)$ wobei $2^n - 1$ prim ist. D. h. zu jeder Mersenne Primzahl gehört eine vollkommene Zahl!
13. Wenn gilt: $\phi(n) + \sigma(n) = 2n$, $n \geq 2$, dann ist n eine Primzahl, wobei $\phi(n)$ die Eulersche Totient-Funktion und $\sigma(n)$ die Teilersummenfunktion ist.
14. Wenn gilt: $\binom{n-1}{k} \equiv (-1)^k \pmod{n}$, dann ist n eine Primzahl, wobei $\binom{n}{k}$ Binomialkoeffizienten sind.
15. Für jede Primzahl p gelten folgende Beziehungen zur Teilersummenfunktion
 $\sigma_0(p) = 2$ (Jede Primzahl besitzt nur 2 Teiler: sich selbst und 1)
 $\sigma_0(p^n) = n + 1$
 $\sigma_1(p) = p + 1$

3.2 FÜR EILIGE: WAS VERMUTEN WIR, HABEN DAFÜR JEDOCH KEINEN BEWEIS?

Hier in verkürzter Form die wichtigsten Aussagen über Primzahlen sowie über die mit den Primzahlen eng verbundene Riemannsche Zetafunktion, die vermutlich richtig sind (jedoch z. Zt. noch unbewiesen):

1. Jede gerade Zahl $n > 2$ lässt sich als Summe zweier Primzahlen darstellen (starke Goldbachsche Vermutung). Die Behauptung wurde numerisch verifiziert für alle $n < 4 \cdot 10^{18}$ (Stand: April 2012).
2. Jede ungerade Zahl > 5 lässt sich als Summe dreier Primzahlen darstellen (schwache Goldbachsche Vermutung). Bewiesen für $n > 10^{43000}$!
3. Zwischen n^2 und $(n + 1)^2$ gibt es mindestens 1 Primzahl (Vermutung von Oppermann, 1882).
4. Die ‚nicht-trivialen‘ Nullstellen der Zeta-Funktion liegen in der Gaußschen Zahlenebene alle auf der Geraden mit Realteil 0.5. Dies ist die berühmte Riemannsche Vermutung, die Riemann im Jahr 1859 aufstellte, und die bis heute nicht bewiesen wurde. Sie ist unter den ‚Top Seven‘ der ungelösten mathematischen Probleme. Für seine Lösung ist ein Preisgeld von einer Million US-Dollar ausgesetzt worden. Numerisch verifiziert wurde Riemanns Behauptung für die ersten 10^{13} Nullstellen). Es gibt also eine überraschende numerische Evidenz für die Richtigkeit der Riemannschen Vermutung.
5. Es gibt unendlich viele Mersenne-Primzahlen (Zahlen der Form $M_p = 2^p - 1$).
6. Es gibt unendlich viele zusammengesetzte Mersenne-Zahlen.
7. Es gibt nur 5 Fermat-Primzahlen.
8. Es gibt keine ungeraden vollkommenen Zahlen (s. oben).
9. Die ‚neue Mersennesche Vermutung‘:
Von folgenden drei Aussagen folgt jeweils die dritte, falls zwei richtig sind:
 - $n = 2^k \pm 1$ oder $n = 4^k \pm 3$
 - $2^n - 1$ ist prim (offensichtlich eine Mersenne-Primzahl)
 - $\frac{(2^n+1)}{3}$ ist prim
10. Es gibt unendlich viele Primzahl-Zwillinge (Primzahl-Zwillinge sind Primzahlen im Abstand von 2). Bekannt ist, dass die Summe der Kehrwerte der Primzahlzwillinge konvergiert (Brunns Konstante: 1.902160577783278, bewiesen von Brun 1919).
11. Die Anzahl N_{M_p} der Mersenne Primzahlen, die kleiner oder gleich N sind verhält sich asymptotisch wie: $N_{M_p}(N) \sim \frac{e^\gamma}{\ln(2)} \ln \ln(N)$.
12. Die Endziffern von hintereinander folgenden Primzahlen zeigen auffällige Korrelationen.

3.3 FÜR EILIGE: WAS IST NOCH UNBEKANNT?

Hier in verkürzter Form die wichtigsten ungeklärten Fragen über Primzahlen und verwandte Themen, von denen wir keine Vermutung haben, ob sie richtig oder falsch sind:

1. Sind alle Mersenne-Zahlen $M_p = 2^p - 1$ quadratfrei (d. h. in deren Primfaktorenzerlegung kommt jede Primzahl genau nur einmal vor)?
2. Gibt es unendlich viele Primzahl N -Tupel? (Das sind n hintereinander folgende Primzahlen in minimalen Abständen, definiert wie in Kapitel 4.3).
3. Gibt es unendlich viele ‚Wagstaff‘-Primzahlen, d. h. Primzahlen der Form $\frac{(2^p+1)}{3}$ (mit einer ungeraden Primzahl p)?
4. Gibt es unendlich viele ‚Sophie Germain‘-Primzahlen, d. h. Primzahlen der Form $2p + 1$ (wobei $2p + 1$ eine ‚sichere Primzahl‘ und p die betroffene ‚Sophie Germain‘-Primzahl ist)?
5. Gibt es unendlich viele Fibonacci-Primzahlen (Primzahlen, die in der Fibonacci-Folge vorkommen)?
6. Enthält die Euklid-Mullin-Sequenz alle Primzahlen?
7. Gibt es ein effizientes Faktorisierungsverfahren zur Primfaktorenzerlegung von großen Zahlen? Gemeint ist ein Verfahren, das die Faktorisierung mit ‚polynomiellen‘ Zeitaufwand bewerkstelligt. Da ein solches Verfahren nicht bekannt ist, können sehr große Zahlen im Moment noch nicht faktorisiert werden (die Rechenzeit hierfür wäre astronomisch hoch). Die derzeit schnellsten Verfahren sind das ‚Zahlenkörpersieb‘ (Pomerance et. al.) sowie die Methode der elliptischen Kurven. Stand: November 2016.

3.4 FÜR EILIGE: WAS IST NEU?

- 1) Eine neue Eigenschaft der Fibonacci-Zahlen (siehe 4.10).
- 2) Eigenschaften der ‚Reed-Jameson‘-Folge und Zusammenhang mit Primzahlen (siehe 4.10.1).
- 3) RG-Zahlenfolgen (rekursiv-, ‚gödelisierte‘) Zahlenfolgen (siehe 4.17).
- 4) ‚Spielereien‘ mit der Produktdarstellung von $\zeta(s)$ im Komplexen (siehe 5.3).
- 5) $\mathfrak{J}(s)$: Eine ‚Funktion‘, deren Minima bei Primzahlen liegen (siehe 8.5.1).
- 6) Die ‚Reed Jameson‘-Funktion: Nullstellen bei Primzahlen (siehe 8.5.2).
- 7) Primzahlen und Oberflächen von 4-dimensionalen Hyperkugeln (siehe 9.3).
- 8) Von OCRONs und GOCRONs (siehe Kapitel 10).
- 9) Kann man mit der ‚Matrix-Software‘ Primzahl-Regeln entdecken? (siehe 11).
- 10) Eine Ebenengleichung als Bindeglied zwischen GOCRONs und der abc-Vermutung (siehe 12.1).
- 11) Primzahlen als rhythmische Muster (siehe 15.2).
- 12) Differenzen und Quotienten von Aliquot-Sequenzen (siehe 20.9.2.5).

4 SPEZIELLE ARTEN VON PRIMZAHLEN

4.1 PRIMZAHL-ZWILLINGE

Primzahl-Zwillinge sind Primzahlen im Abstand von 2. Es existiert folgende Beziehung: n und $n + 2$ sind dann und nur dann genau ein Primzahl-Zwilling wenn gilt:

$$4[(n - 1)! + 1] + n \equiv 0 \pmod{n(n + 2)} \quad (1)$$

$$\begin{aligned} \phi(n)\sigma(n) &= (n - 3)(n + 1), \text{ mit } n \\ &= p(p + 2) \text{ (Produkt eines Primzahlpaares)} \end{aligned} \quad (2)$$

$(n, n + 2)$ sind Primzahlzwillinge, wenn gilt:

$$\sum_{i=1}^n i^a \left(\left\lfloor \frac{n+2}{i} \right\rfloor + \left\lfloor \frac{n}{i} \right\rfloor \right) = 2 + n^a + \sum_{i=1}^n i^a \left(\left\lfloor \frac{n+1}{i} \right\rfloor + \left\lfloor \frac{n-1}{i} \right\rfloor \right) \quad (3)$$

wobei $a \geq 0$ und $\lfloor \cdot \rfloor$ die floor()-Funktion ist.

Leider sind diese Formeln praktisch zur Berechnung von Primzahl-Zwillingen nicht geeignet.

Sei $\pi_2(x)$ die Anzahl der Primzahl-Zwillinge bis zu einem gegebenen Wert x . Bekannt ist seit dem 19. Jahrhundert die Abschätzung:

$$\pi_2(x) \leq c\Pi_2 \frac{x}{(\ln x)^2} \quad (4)$$

Hardy und Littlewood vermuteten, dass $c = 2$ und dass

$$\pi_2(x) \sim 2\Pi_2 \int_2^x \frac{dt}{(\ln t)^2} = 2\Pi_2 (\text{Li}(x) - \frac{x}{\ln(x)} - \text{Li}(2) + \frac{2}{\ln(2)}) \quad (5)$$

mit der Primzahl-Zwillingskonstanten:

$$\Pi_2 = \prod_{p \geq 3} \frac{p(p-2)}{(p-1)^2} = 0.6601618158$$

$$2\Pi_2 = 1.3203236316$$

Die Summe der Kehrwerte der Primzahlzwillinge konvergiert (Brun's Konstante: 1.902160577783278, bewiesen von Brun 1919):

$$B = \sum_{p=\text{Zwilling}} \left(\frac{1}{p} + \frac{1}{p+2} \right) = 1.90216054 \quad (6)$$

Tabelle 1: Anzahl der Primzahlzwillinge und Werte der Hardy-Littlewood-Funktion

n	$\pi_2(10^n)$	Hardy-Littlewood
1	2	4.84
2	8	13.54
3	35	45.80
4	205	214.21
5	1224	1248.71
6	8169	8248.03
7	58980	58753.82
8	440312	440367.79
9	3424506	3425308.16
10	27412679	27411416.53
11	224376048	224368864.67
12	1870585220	1870559866.69
13	15834664872	15834598303.94
14	135780321665	135780264884.86
15	1177209242304	1177208491777.05
16	10304195697298	10304192553765.33
17	9094883 9353159	9094883 3254536.36
18	8086758 88577436	8086759 01436127.88

Für $n = 10^{18}$ ist die Näherung von Hardy-Littlewood exakt bis auf einen Fehler von

$$1.59 \cdot 10^{-8} : \quad \frac{\pi_2(10^{18})}{\pi_{2_{approx}}(10^{18})} = 0.999999984$$

Die Übereinstimmung der Näherungen mit den exakten Werten für große n ist bemerkenswert und könnte als ‚numerischer Beweis‘ für die Unendlichkeit der Anzahl der Primzahl-Zwillinge angesehen werden (siehe 4.1).

Mathematica-Programm zum Erzeugen der Tabelle:

```
ile = 2; Do[Do[If[(PrimeQ[2 n - 1]) && (PrimeQ[2 n + 1]), ile = ile + 1], {n, 5*10^m, 5*10^(m + 1)}]; Print[{m, ile}], {m, 0, 7}]
```

Primzahl-Zwillinge

Hier der Vergleich zwischen exakten Werten und der Formel von Hardy und Littlewood für die ersten 3500 Zwillings-Primzahlen (Blau: exakt, Gelb: Hardy-Littlewood) :

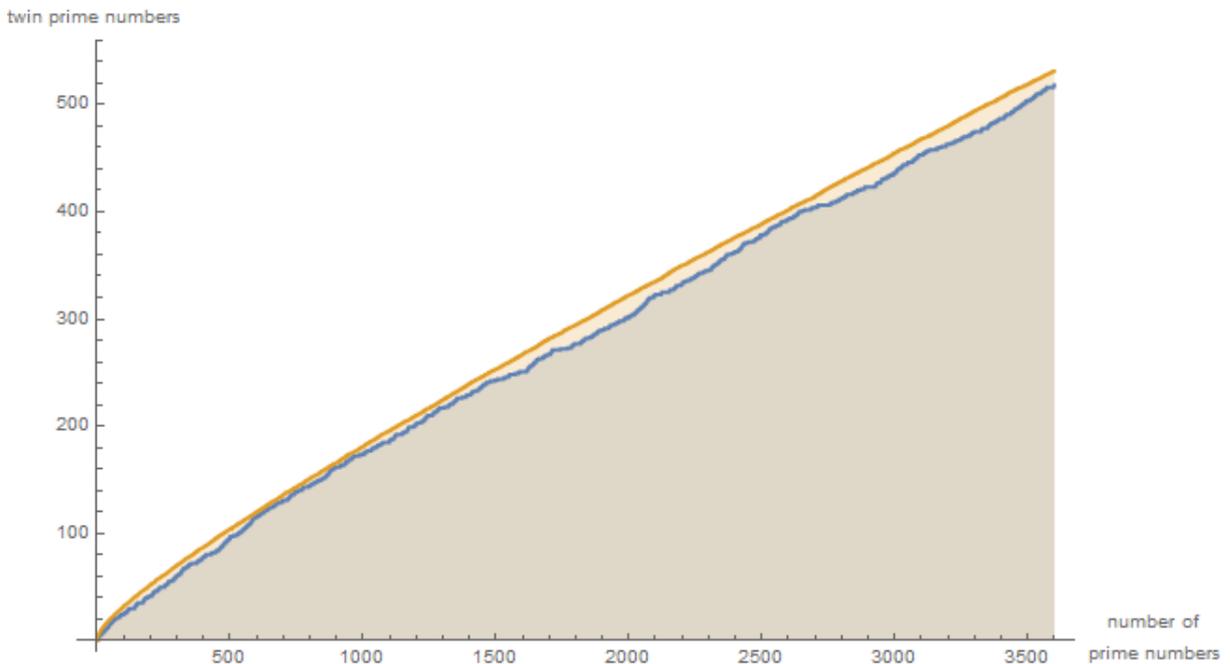


Abbildung 1: Anzahl der Primzahl-Zwillinge von 2 bis 3500

Das Mathematica-Programm zur Erzeugung des Plots findet sich im Anhang³. Mit folgendem Mathematica-Programm können Primzahl-Zwillinge (z. B. bis 10000) berechnet werden:

```
Select[Range[10000], (PrimeQ[#] && PrimeQ[#+2]) &]
```

Der größte bekannte Primzahl-Zwilling liegt bei $3756801695685 \cdot 2^{66669} \pm 1$ (In Dezimalschreibweise: $8.72966507873032809131455726874 \times 10^{200699}$)
Stand: Oktober 2015.

Polignacs Vermutung:

Diese Vermutung besagt, dass es für jede gerade Zahl n unendlich viele Primzahlen-Paare gibt, die in diesem Abstand voneinander liegen. Für $n = 2$ erhalten wir den Spezialfall der Primzahl-Zwillinge. Die Hardy-Littlewood Vermutung lässt sich auch für diesen Fall verallgemeinern:

$$\pi'_n(x) \sim 2C'_n \int_2^x \frac{dt}{(\ln t)^2} = 2C'_n \left(\text{Li}(x) - \frac{x}{\ln(x)} - \text{Li}(2) + \frac{2}{\ln(2)} \right) \quad (7)$$

mit

³ Mathematica-Programme: Vergleich der Anzahl Von Primzahl-Zwillingen, -Cousins und Sexy-Primzahlen mit der Formel von Hardy-Littlewood

$$C'_n = \prod_2 \sum_{q|n} \frac{q-1}{q-2} \quad (8)$$

Spezialfälle:

$n = 4$: **Cousin-Primzahlen**: Hier ist $C'_4 = C'_2 = C_2$. Cousin-Primzahlen (im Abstand 4) und Primzahl-Zwillinge besitzen die gleich asymptotische Dichte. Es gibt von beiden Arten gleich viele Exemplare!

$n = 6$: **Sexy Primzahlen**: Hier ist $C'_6 = 2C'_2$. Sexy Primzahlen (im Abstand 6) besitzen eine doppelt so hohe asymptotische Dichte wie Primzahl-Zwillinge. Es gibt doppelt so viele Sexy Primzahlen wie Primzahl-Zwillinge!

4.2 PRIMZAHL-DRILLINGE UND -VIERLINGE

Auch für Primzahl-Drillinge und -Vierlinge gibt es Abschätzungen (Hardy-Littlewood-Vermutung) der Anzahl von Drillingen und Vierlingen bis zu einem gegebenen x :

Drillinge:

$$\pi_3(x) \leq \frac{9}{2} \prod_{p \geq 5} \frac{p^2(p-3)}{(p-1)^3} \int_2^x \frac{dt}{(\ln t)^3} = 2.858248596 \int_2^x \frac{dt}{(\ln t)^3} \quad (9)$$

Explizit:

$$\pi_3(x) \sim 2.858248596 \left(\frac{1}{2} \text{Li}(x) - \frac{x}{2 \ln^2(x)} - \frac{x}{2 \ln(x)} + \frac{1}{\ln(2)} + \frac{1}{\ln^2(2)} - \frac{1}{2} \text{Li}(2) \right) \quad (10)$$

Oder:

$$\pi_3(x) \sim 2.858248596 \left[\ln^{-2}(x) \left(-E_3(-\ln(x)) \right) - \ln^{-2}(2) \left(-E_3(-\ln(2)) \right) \right] \quad (11)$$

Der größte derzeit bekannte Primzahl-Drilling:

$$6521953289619 \cdot 2^{55555} + d, d = -5, -1, 1 \text{ (16737 Dezimalstellen)}$$

Tabelle 2: Anzahl der Primzahl-Drillings und Werte der Hardy-Littlewood-Funktion⁴

n	$\pi_3(10^n)$	Hardy-Littlewood	H-L / $\pi_3(10^n)$
1	1	8.49	8.490
2	4	13.86	3.465
3	15	25.57	1.70467
4	55	69.34	1.26073
5	259	279.36	1.07861
6	1393	1446.17	1.03817
7	8543	8591.23	1.00565
8	55600	55490.86	0.99804
9	379508	379802.73	1.00078
10	2713347	2715291.84	1.00072
11	20093124	20089653.88	0.99983
12	152850135	152830566.82	0.99997
13	1189795268	1189763105.37	0.999999
14	9443899421	9443890414.16	0.999999
15	76218094021	76217780005.59	0.999996
16	624026299748	624025187564.06	0.999998

Die Übereinstimmung der Näherungen mit den exakten Werten für große n ist bemerkenswert.

Mathematica-Programm zum Erzeugen der Näherungswerte:

```
ch=2.858248596; (*Pi3!*)
n=3;
Do[Print[N[Re[SetPrecision[ch,50]*((Log[10^i])^(1-n)(-ExpIntegralE[n,-
Log[10^i]])-(Log[2])^(1-n)(-ExpIntegralE[n,-Log[2]]))],{Infinity,3}
]],{i,1,16}]
```

Vierlinge:

$$\pi_4(x) \leq \frac{27}{2} \prod_{p \geq 5} \frac{p^3(p-4)}{(p-1)^4} \int_2^x \frac{dt}{(\ln t)^4} = 4.151180864 \int_2^x \frac{dt}{(\ln t)^4} \quad (12)$$

Oder:

$$\pi_4(x) \sim 4.151180864 \left[\ln^{-3}(x) \left(-E_4(-\ln(x)) \right) - \ln^{-3}(2) \left(-E_4(-\ln(2)) \right) \right] \quad (13)$$

⁴ Quelle: Thomas R.Nicely, http://www.trnicely.net/quads/t3a_0000.htm

Tabelle 3: Anzahl der Primzahl-Quadrupel und Werte der Hardy-Littlewood-Funktion⁵:

n	$\pi_4(10^n)$	Hardy-Littlewood	H-L / $\pi_4(10^n)$
1	1	11.29	11.29
2	2	13.60	6.80
3	5	16.49	3.30
4	12	24.17	2.01
5	38	52.88	1.39
6	166	183.68	1.1065
7	899	862.95	0.9599
8	4768	4734.64	0.99300
9	28388	28396.84	1.00031
10	180529	181074.93	1.00302
11	1209318	1209956.22	1.00053
12	8398278	8394578.03	0.99956
13	60070590	60075438.37	1.00008
14	441296836	441290732.40	0.999986
15	3314576487	3314550290.38	0.999992
16	25379433651	25379441340.00	1.0000000

Die Übereinstimmung der Näherungen mit den exakten Werten für große n ist auch hier bemerkenswert.

Mathematica-Programm zum Erzeugen der Näherungswerte:

```
ch=4.151180864; (*Pi4!*)
```

```
n=4;
```

```
Do[Print[N[Re[SetPrecision[ch, 50]*((Log[10^i])^(1-n) (-ExpIntegralE[n, -Log[10^i]]) - (Log[2])^(1-n) (-ExpIntegralE[n, -Log[2]]))]], {Infinity, 3}]], {i, 1, 16}]
```

Der größte derzeit bekannte Primzahl-Vierling: (Quellen: T.Forbes⁶)

$$2673092556681 \cdot 15^{3048} + d, d = -4, -2, 2, 4$$

$$= 1.42289088832921708944844369162 \cdot 10^{3597}$$

Stand: Oktober 2015.

4.3 PRIMZAHL N-LINGE

Die Definition für Primzahl N-ling oder Primzahl n -Tupel ist im Allgemeinen als eine Sequenz von aufeinanderfolgenden Primzahlen $(p_1, p_2, p_3, \dots, p_n)$ beschrieben, wobei der Abstand zwischen der kleinsten und der größten Primzahl $s(n) = p_n - p_1$ einen festen

⁵ Quelle: Thomas R.Nicely, http://www.trnicely.net/quads/t4_0000.htm

⁶ <http://anthony.d.forbes.googlepages.com/ktuplets.htm>:

minimalen Wert hat (siehe Tabelle unten). So ist z. B. $s(4) = 8$ für Quadrupel oder $s(5) = 12$ für Quintupel. Im Allgemeinen gibt es für ein gegebenes Primzahl n -Tupel mit festem $s(n)$ mehrere Lösungen für die entsprechende Primzahl Sequenz. Z. B. können Primzahl-Tripel folgende zwei Formen haben:

$(p, p + 2, p + 6)$ sowie $(p, p + 4, p + 6)$. Diese ‚Entartung‘ wächst ziemlich schnell mit der Länge n der n -Tupel an. So beträgt für $n = 13$ die Entartung bereits 6; für $n = 25$ haben wir eine Entartung von 18 verschiedenen Anordnungsmöglichkeiten für ein Primzahl 25-Tupel mit $s(25) = 110$.

Um diese Mehrdeutigkeit oder ‚Entartung‘ zu vermeiden sei hier eine andere Definition von Primzahl n -Tupeln vorgestellt. Wir konstruieren eine Folge von Primzahlen, die in einem beliebig hohen Zahlenbereich angesiedelt ist, mit der höchst möglichen Primzahldichte auf folgende Art und Weise (das Prinzip hat eine gewisse Ähnlichkeit mit dem ‚Sieb des Eratosthenes‘).

- 1) **Wir beginnen eine Liste und nehmen an, das erste Element dieser Liste sei eine beliebig große Primzahl p (die offensichtlich ungerade ist). Wir markieren das erste Listenelement mit ‚ p ‘. Alle anderen Listenelemente sind noch unmarkiert (‚frei‘).**
- 2) **Wir setzen $n = 1$ (folglich ist $p_n = 2$, die erste Primzahl)**
- 3) **Solange im Bereich zwischen p und $p + p_n - 1$ (also zwischen dem ersten und dem p_n -ten Element) noch mehr als ein Listenelement durch p_n teilbar sein kann (d.h. alle im Abstand $i \cdot p_n$ folgenden Elemente ‚kollidieren‘ nicht mit einem als ‚ p ‘ markiertem Element) schränken wir die Mehrdeutigkeit immer weiter ein, in dem wir jeweils die nächste freie (noch nicht mit teilbaren Zahlen oder ‚ p ‘ markiert) Position mit einem ‚ p ‘ markieren.**
- 4) **Zwischen p und $p + p_n - 1$ (zwischen dem ersten und dem $p_n - \text{ten}$) Element existiert jetzt nur noch ein Listenelement $p + j$, das durch p_n teilbar ist. Wir sieben (d.h. markieren mit dem Wert p_n) alle folgenden Zahlen (Listenelemente) $p + j + i \cdot p_n$, $i = 0, 1, \dots \infty$**
- 5) **Wir setzen die nächstmögliche Primzahl auf das nächstmögliche freie Feld und markieren dieses Feld mit einem ‚ p ‘.**
- 6) **Wir erhöhen den Zähler $n = n + 1$ und fahren fort mit 3).**

Wir erhalten somit eine Folge von (möglichen) Primzahlen (die in unserer Liste mit ‚ p ‘ markiert wurden, die die maximale Primzahldichte repräsentieren (unabhängig vom Zahlenbereich, in dem wir starteten):

$$p, p + 2, p + 6, p + 8, p + 12, p + 18, p + 20, p + 26, p + 30, p + 32, p + 36, p + 42, p + 48, p + 50, p + 56, p + 62, p + 68, p + 72, p + 78, p + 86, p + 90$$

Die durch dieses Konstruktionsprinzip erstellten Primzahl n -Tupel weichen von der Tabelle der Primzahl n -Tupel in T. Forbes Tabelle⁷ ab. Die Zahlen haben die Bedeutung von Indices i für $(p + i)$:

⁷ <http://anthony.d.forbes.googlepages.com/ktuplets.htm>

6-Tupel: (0-2-6-8-12-18).

Bei Forbes: (0-4-6-10-12-16)

16-Tupel: (0-2-6-8-12-18-20-26-30-32-36-42-48-50-56-62)

Bei Forbes: (0-4-6-10-16-18-24-28-30-34-40-46-48-54-58-60)

Oder (0-2-6-12-14-20-26-30-32-36-42-44-50-54-56-60)

Ab den 16-Tupeln häufen sich die Abweichungen.

Interessant ist, dass durch dieses Konstruktionsprinzip, welches für beliebig große Zahlen eine maximale Primzahldichte ergibt, sich die Sequenz der Primzahlen ab der Zahl 11 ergibt. Zumindest beim ersten Blick... Beim genaueren Hinschauen bemerken wir, dass die **Primzahl 71** (die wäre in unserer Liste an Position 60) nicht dabei ist! **Die einzig mögliche Interpretation lautet, dass es für hinreichend große Zahlenbereiche keine Primzahl-16-Tupel der Form (0-2-6-8-12-18-20-26-30-32-36-42-48-50-56-60) (wie durch die 16 Primzahlen von 11 bis 71 dargestellt) geben kann. Im Bereich über 100 häufen sich die Abweichungen von der ‚natürlichen‘ Primzahl-Sequenz ab 11.**

Es ist offensichtlich, dass, wenn wir dieses Konstruktionsprinzip immer weiter führen, die entstehenden Primzahlpositionen für Primzahl- n -Tupel sich im Vergleich zur natürlichen Primzahlfolge ab der Zahl 11 immer weiter ausdünnen werden. Das ist auch einleuchtend, denn es hätte wohl niemand erwartet, dass sich alle Primzahlkonstellationen der ‚kleinen‘ Zahlen auch in beliebig großen Zahlenbereichen unendlich oft wiederholen.

Fazit: Nicht nur die Primzahlkonstellation innerhalb der ersten 10 natürlichen Zahlen ist einzigartig. Auch für Zahlen größer als 11 gibt es Primzahlkonstellationen, die nur einmal auftreten.

Die Sequenz der möglichen Primzahl-Positionen, die eine maximale ‚Dichte‘ aufweisen lautet:

{1,3,7,9,13,19,21,27,31,33,37,43,49,51,57,63,69,73, ... } und sie ist nicht unbekannt (siehe A020498 in <https://oeis.org>).

Sehen wir uns die Sequenzen der Differenzen an, die sich nach den Sieb-Vorgängen bis einschließlich zur Primzahl Nr. n ergeben. Diese Sequenzen können einfach mit Hilfe der folgenden Formel berechnet werden:

```
RotateRight[Differences[Select[Range[primorial[n]+1], GCD[#, primorial[n]]==1&]], nRotation]
```

Die Sequenzen besitzen Zyklen, deren Länge anwächst. Der Parameter ‚nRotation‘ aus der folgenden Tabelle entnommen werden. `Primorial[n]` ist das Produkt der ersten n Primzahlen.

Die Längen der Zyklen können ebenfalls leicht mit folgender Formel berechnet werden:

$$a(0)=1; \text{ for } n>0, a(n)=(\text{prime}(n)-1)*a(n-1)$$

Primzahl N-Linge

Mathematica:

```
RecurrenceTable[{a[0]==1, a[n]==(Prime[n]-1)a[n-1]}, a, {n, 10}]
```

Die Sequenzen, die sich aus den Differenzen der möglichen Primzahl-Positionen ergeben (durch Siebungen) wiederholen sich nach folgenden Zyklen:

n	Siebung bis n-ter Primzahl	Länge Zyklus	Parameter für RotateRight[]	Sequenzen (Anfang und Ende)
1	2	1	0	2
2	3	2	1	2,4
3	5	8	6	2,4,2,4,6,2,6,4
4	7	48	47	2,4,2,4,6,2,6,4,2,4,6,6,2,6,4,...,10,2,10
5	11	480	218	2,4,2,4,6,2,6,4,2,4,6,6,2,6,6,...,10,2,10
6	13	5760	2861	2,4,2,4,6,2,6,4,2,4,6,6,2,6,6,...,10,2,10
7	17	92160	2695	(wie oben)
8	19	1658880	???	(wie oben)

Figure 1: Zyklen-Längen bei Siebmethode zur Erzeugung maximaler Primzahldichte

Die Siebmethode wird deutlicher, wenn wir uns die folgende Tabelle anschauen (siehe Anhang für das benutzte Mathematica-Programm). Die gelben Markierungen zeigen uns die möglichen Primzahl-Positionen an, die in beliebig hohen Zahlenbereichen auftreten können. Die erste Zeile zeigt die tatsächlich existierenden Primzahlen ab der Zahl 11. Die erste Abweichung findet man bei der Primzahl 71. Die Abweichungen werden häufiger, wenn wir zu höheren Zahlenbereichen gehen.

11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
p	2	p	2		2	p	2	p	2		2	p	2		2		2	p	2	p	2		2		2	p	2		2	p	2	p	2		2
p	3	p		3		p	3	p		3		p	3		3		p	3	p		3		3		3	p		3	p	3	p		3		
p	p	p	5		p	p	5		p		5		p	5	p		5		p	5	p		5		5	p		5	p	5	p		5		
p	p	7			p	p	7		p		7		p	7	p		7		p	7	p		7		7	p		7	p	7	p		7		
p		p		11		p		p		11		p		11		p		p		p		11		p		11		p	11		p		11		
p	13	p			p	p		p		13		p		13		p		p		p		13		p		13		p	13		p		13		
p	p	p			p	17	p		p		17		p		17		p		p		17		p		17		p	17		p		17			
p	p	p			p	p		p		19		p		19		p		p		p		19		p		19		p	19		p		19		
47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82
p	2		2		2	p	2		2		2	p	2	p	2		2		2	p	2		2		2	p	2		2		2	p	2		2
p	3			3		p	3		3		p	3	p		3		3		3	p		3		3		3	p		3		3	p		3	
p	p		5		p	p	5		p		5		p	5	p		5		p	5	p		5		5	p		5		p	5	p		5	
p	p	7			p	p	7		p		7		p	7	p		7		p	7	p		7		7	p		7		p	7	p		7	
p		11			p		p		p		11		p		11		p		p		11		p		11		p		11		p		11		
p			13		p		p		p		13		p		13		p		p		13		p		13		p		13		p		13		
p				17	p		p		p		17		p		17		p		p		17		p		17		p		17		p		17		
p					p		p		p		19		p		19		p		p		19		p		19		p		19		p		19		
83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113					
p	2		2		2	p	2		2		2		2	p	2		2		2	p	2		2		2	p	2	p	2		2		2		
p	3			3		p	3		3		3		3	p	3		3		3	p	3		3		3	p	3	p	3		3		3		
p	p	5			p	p	5		p		5		p	5	p		5		p	5	p		5		5	p		5		p	5	p		5	
p	7				p		7		p		7		p	7	p		7		p	7	p		7		7	p		7		p	7	p		7	
p					p		11		p		11		p		11		p		p		11		11		p		11		p	11		p		11	
p					p		13		p		13		p		13		p		p		13		13		p		13		p	13		p		13	
p			17		p		p		17		17		p		17		p	17		p		17													
p	19				p		19		p		19		p		19		p		p		19		19		p		19		p	19		p		19	

Figure 2: Siebmethode zur Erzeugung einer maximalen Primzahl-Dichte

Die Internetseite von T. Forbes ist eine wahre Fundgrube zu diesem Thema. Die folgenden Formeln wurden zum größten Teil von dieser Seite übernommen.

Verallgemeinern wir die Abschätzung von Hardy-Littlewood für beliebige n , (Primzahl n -Tupel, oder Primzahl n -Länge), so wird vermutlich:

$$\pi_n(x) \sim C_n \left[\ln^{1-n}(x) \left(-E_n(-\ln(x)) \right) - \ln^{1-n}(2) \left(-E_n(-\ln(2)) \right) \right] \quad (14)$$

mit den Konstanten C_n . Dabei ist $E_n()$ die Integral-Exponential-Funktion n -ter Ordnung.

Die Konstanten C_n berechnen sich wie folgt:

$$C_n = H_n \cdot K_n$$

und

$$K_n = \prod_{p \geq n+1} \frac{p^{n-1}(p-n)}{(p-1)^n} \quad (15)$$

Und hier eine Formel für C_n die sehr viel schneller konvergiert:

$$\ln(C_k) = \sum_{n=2}^{\infty} \ln \left[\zeta(n) \prod_{p \text{ prime}, p \leq k} \left(1 - \frac{1}{p^n} \right) \right] / n \cdot \sum_{d|n} \mu \left(\frac{n}{d} \right) (k^d - k) \quad (16)$$

Primzahl N-Linge

Tabelle 4: Die Hardy-Littlewood-Konstanten

Name	Abstände, $p_{max} - p_{min}$	H_n	K_n	C_n
Zwillinge	2 (2)	2	0.66016182	1.3203236
Drillinge	2-4 (6)	$\frac{9}{2}$	0.63516635	2.8582486
Vierlinge	2-4-2 (8)	$\frac{27}{2}$	0.30749488	4.1511809
5-Tupel	2-4-2-4 (12)	$\frac{15^4}{2^{11}}$	0.40987489	10.131795
6-Tupel (*)	4-2-4-2-4 (16)	$\frac{15^5}{2^{13}}$	0.18661430	17.298612
6-Tupel	2-4-2-4-6 (18)	$\frac{?}{?}$?	?
7-Tupel	2-4-2-4-6-2 (20)	$\frac{35^6}{3 \cdot 2^{22}}$	0.36943751	53.971948
8-Tupel	2-4-2-4-6-2-6 (26)	$\frac{5^6 \cdot 7^7}{2^{24}}$	0.23241933	178.26195
9-Tupel	2-4-2-4-6-2-6-4 (30)	$\frac{5^9 \cdot 7^8}{2^{31}}$	0.12017121	630.06436
10-Tupel	2-4-2-4-6-2-6-4-2 (32)	$\frac{5^{10} \cdot 7^9}{9 \cdot 2^{30}}$	0.041804051	1704.7409
11-Tupel	2-4-2-4-6-2-6-4-2-4 (36)	$\frac{7^{11} \cdot 11^{10}}{45 \cdot 2^{45}}$	0.094530829	3062.0793
12-Tupel	2-4-2-4-6-2-6-4-2-4-6 (42)	$\frac{7^{12} \cdot 11^{11}}{25 \cdot 2^{49}}$	0.035393260	9931.3156

Tabelle 5: Anzahl der Primzahl-Quintupel und Werte der Hardy-Littlewood-Funktion

n	$\pi_5(10^n)$	Hardy-Littlewood	$H-L / \pi_5(10^n)$
1	1	27.93	-
2	2	14.84	7.42
3	3	30.90	10.30
4	4	33.17	8.29
5	10	39.79	3.98
6	34	64.49	1.90
7	160	173.20	1.0825
8	697	711.00	1.02009
9	3633	3615.11	0.99508
9.59868	10000	10094.08	1.00941
10	20203	20401.37	1.00982
11		122857.37	?
12		776698.49	?
13		$5.10724390 \cdot 10^6$?
14		$3.4706125667 \cdot 10^7$?

15		$2.42544985095 \cdot 10^8$?
16		$1.73651359676 \cdot 10^9$	

Mathematica-Programm zum Erzeugen der Näherungswerte:

```
ch=10.131795; (*Pi5!*)
```

```
n=5;
```

```
Do[Print[N[Re[SetPrecision[ch,50]*((Log[10^i])^(1-n)(-ExpIntegralE[n,-Log[10^i]])-(Log[2])^(1-n)(-ExpIntegralE[n,-Log[2]]))]],{Infinity,3}]],{i,1,16}]
```

(blaue Werte wurden analytisch mit der Hardy-Littlewood-Formel berechnet und sind nicht exakt...)

Tabelle 6: Anzahl der Primzahl-n-Tupel in Abhängigkeit von n

n	$\pi_n(10^3)$	$\pi_n(10^4)$	$\pi_n(10^5)$	$\pi_n(10^6)$	$\pi_n(10^7)$	$\pi_n(10^8)$	$\pi_n(10^9)$
1	168	1229	9592	78498	664579	5761455	50847534
2	35	205	1224	8169	58980	440312	3424506
3	15	55	259	1393	8543	55600	379508
4	5	12	38	166	899	4768	28388
5	3	4	10	34	160	697	3633
6	2	3	3	6	16	63	
7							
8							
9							
10							
11							
12							
n	$\pi_n(10^{10})$	$\pi_n(10^{11})$	$\pi_n(10^{12})$	$\pi_n(10^{13})$	$\pi_n(10^{14})$		
1	455052511	4118054813	37607912018	346065536839	3204941750802		
2	27412679	224376048	1870585220	15834664872	135780321665		
3	2713347	20093124	152850135	1189795268	9443899421		
4	180529	1209318	8398278	60070590	441296836		
5	20203	122857	776698	5107243	34706125		
6	?	?	?	?	?		
7	424	1334	6185	33412	193160		
8	?	?	?	4601	21405		
9							
10							
11							
12							
n	$\pi_n(10^{15})$	$\pi_n(10^{16})$					
1	29844570422669	279238341033925					
2	1177209242304	10304195697298					
3	76218094021	624026299748					
4	3314576487	25379433651					
5	242544985	1736513596					
6							
7							
8							
9							
10							
11							

4.4 KORRELATIONEN DER ENDZIFFERN IN AUFEINANDER FOLGENDEN PRIMZAHLEN

Im Frühjahr 2016 ging eine Aufsehen erregende Mitteilung durch die Presse: Mathematiker hätten auffällige Muster in Primzahlen gefunden. Die statistischen Häufigkeiten bei den Endziffern aufeinander folgender Primzahlen zeigen deutliche Korrelationen. Da eine Primzahl (von der ersten und dritten Primzahl 2 und 5 abgesehen) nur mit einer der vier Ziffern 1,3,7,9 enden kann, erwartet man wegen des ‚zufälligen‘ Auftretens der Primzahlen, dass die Endziffern 1,3,7,9 jeweils mit einer Wahrscheinlichkeit von 25 % auftreten. Dies ist auch tatsächlich so. Eine Auswertung der Endziffern der ersten Million Primzahlen ergibt, dass die Wahrscheinlichkeiten der Endziffer gleichverteilt sind (25 %):

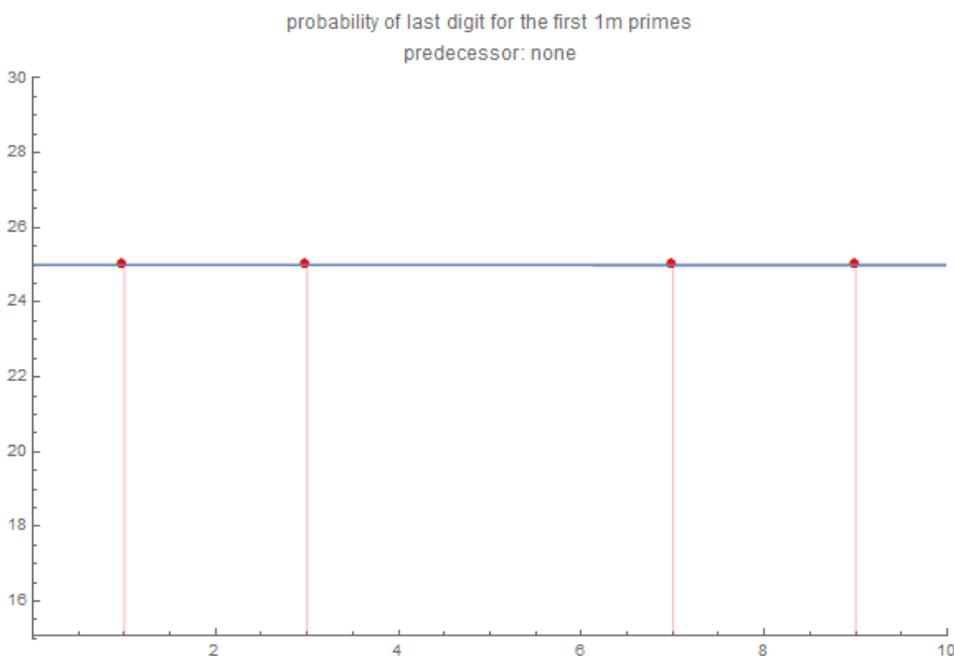


Abbildung 2: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (ohne Vorgänger)

```

Mathematica:
data={{1,24.99},{3,25.01},{7,25.00},{9,24.99}}
line=Fit[data,{1,x},x]
Show[ListPlot[data,PlotStyle->Red,AxesLabel->Automatic,Filling->Axis,PlotMarkers->Automatic,PlotRange->{{0,10},{15,30}},PlotLabel->TextString["probability of last digit for the first 1 Mio primes\npredecessor: none"],ImageSize->Large],Plot[line,{x,0,10}]]
    
```

Untersuchen wir jedoch die statistischen Eigenschaften der möglichen Primzahl-Nachfolger für eine fest vorgegebene Ziffer, z. B. die 1, dann stellt man fest, dass die Wahrscheinlichkeit für eine nachfolgende 1 deutlich unter 25 % liegt. Auch alle anderen möglichen Nachfolge-Ziffern zeigen deutliche Abweichungen von der zunächst zu erwartenden 25 % Abschätzung. Im Falle einer 1 als letzter Ziffer, ist die Wahrscheinlichkeit, dass die nächste Primzahl wieder mit einer 1 endet nur ca. 18 %. Man könnte es salopp formulieren: Primzahlen wiederholen ihre Endziffer nur ungern. Tatsächlich stellt man diese Tendenz für alle Ziffern fest. Für die ersten 10 Millionen Primzahlen finden wir folgende statistischen Abhängigkeiten der letzten Primzahl-Ziffern:

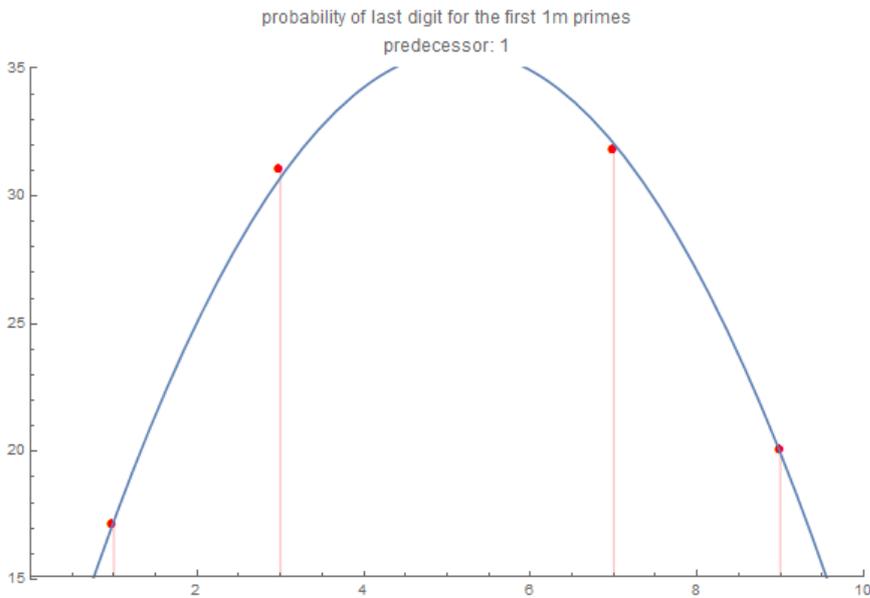


Abbildung 3: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger: ,1')

Mathematica:

```
data={{1,17.15},{3,31.00},{7,31.79},{9,20.07}}
line=Fit[data,{1,x,x^2},x]
Show[ListPlot[data,PlotStyle->Red,AxesLabel->Automatic,Filling->Axis,PlotMarkers->Automatic,PlotRange->{{0,10},{15,35}},PlotLabel->TextString["probability of last digit for the first 1 Mio primes\npredecessor: 1"],ImageSize->Large],Plot[line,{x,0,10}]]
```

Hier die Ergebnisse für alle vier möglichen Endziffern:

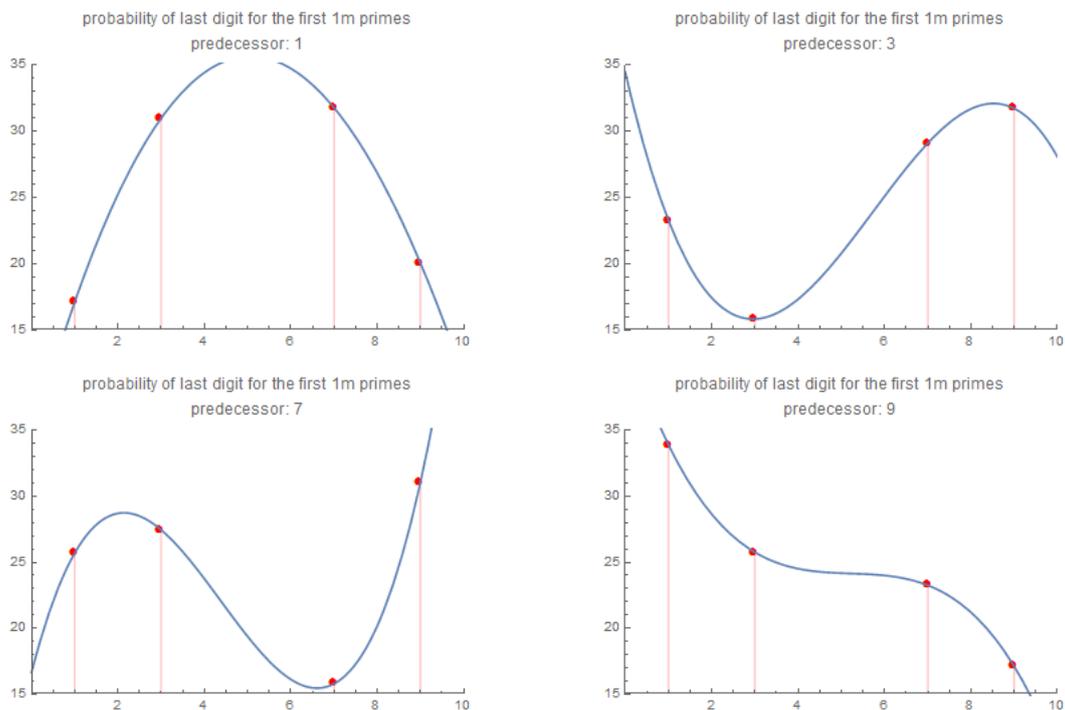


Abbildung 4: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (alle mögl. Vorgänger)

Mersenne-Primzahlen

Mathematica:

(Programme siehe Anhang).

Man mag sich fragen, wie diese statistischen Auffälligkeiten aussehen, wenn man noch mehr vorangegangene Primzahlen in die Untersuchung mit einbezieht. Ergebnisse, wenn wir nicht nur die Vorgänger, sondern auch die Vor-Vorgänger berücksichtigen, finden sich im Anhang (Kapitel 20.1).

Diese Korrelationen der letzten Ziffern aufeinanderfolgender Primzahlen gelten übrigens nicht nur für die Ziffern der Darstellung im Dezimalsystem. Auch in allen anderen Darstellungen (z. B. Binärsystem oder Vierer-System) findet man die gleichen Auffälligkeiten.

Eine genauere Untersuchung hat inzwischen ergeben, dass die beobachteten Korrelationen eine direkte Konsequenz der (noch unbewiesenen) Hardy-Littlewood Vermutung sind (siehe Formel (14) im Kapitel 4.3). Die Beobachtung, dass diese Korrelationen immer schwächer werden, je größer die betrachteten Zahlenbereiche sind, lässt sich ebenfalls mit Hilfe der Hardy-Littlewood-Vermutung ableiten. Vermutlich werden diese statistischen Auffälligkeiten ganz verschwinden wenn man in beliebig hohe Zahlenbereiche vordringt. Diese müssten dann allerdings astronomisch hoch sein, denn die Korrelationen werden nur ganz allmählich kleiner.

Diese langsame Geschwindigkeit, mit der die Auffälligkeiten ‚verschwinden‘ ist eigentlich das einzig Merkwürdige an der ganzen Geschichte.

4.5 MERSENNE-PRIMZAHLEN

Zu den Mersenne-Primzahlen gibt es eine unüberschaubare Menge an Literatur, so dass hier nur die wichtigsten Formeln und Aussagen aufgeführt werden sollen:

Derzeit sind 51 Mersenne-Primzahlen bekannt (Stand: Dez. 2020). Viele Fragen zu dem Mersenne-Primzahlen sind noch ungelöst (siehe 3.2 Grundlegendes zu Primzahlen).

Mersenne-Primzahlen haben die Form $M_n = 2^p - 1$ wobei p eine Primzahl sein muss, jedoch nicht jede Primzahl p eine Mersenne-Primzahl M_n ergibt. Die Mersenne-Primzahlen sind sehr selten und die Suche nach ihr ist wie die Suche nach den Edelsteinen unter den Zahlen. Die größten bekannten Primzahlen sind alle Mersenne Primzahlen. Das liegt daran, dass für diesen Typ von Primzahlen ein sehr schneller Primzahltest existiert, der es erlaubt auch Primzahlen mit wahrhaft titanischen Ausmaßen zu testen. Die derzeit größte bekannte Primzahl ist die Mersenne-Primzahl $2^{82589933}-1$. Sie besitzt 24862048 Stellen im Dezimalsystem (Stand: Dez. 2020).

Der schnelle Primzahltest für Mersenne-Primzahlen ist der Lucas-Lehmer Test⁸, der mit anderen Verfahren kombiniert wird. Für einen Primzahltest einer Zahl dieser Größenordnung braucht auch ein schneller PC mit 4 Prozessorkernen ca. 1 Monat (Stand Nov. 2016). Der Lucas-Lehmer Test und die verwendeten Faktorisierungsmethoden (P1-Test

⁸ <https://de.wikipedia.org/wiki/Lucas-Lehmer-Test>

und Trial-Factoring) sind vielfach ausführlich beschrieben worden und sollen hier nicht erläutert werden.⁹

Die zur Zeit (Dez. 2020) bekannten 51 Mersenne-Primzahl-Exponenten lauten:

**2, 3, 5, 7, 13, 17, 19, 31, 61, 89,
107, 127, 521, 607, 1279, 2203, 2281, 3217, 4253, 4423,
9689, 9941, 11213, 19937, 21701, 23209, 44497, 86243,
110503, 132049, 216091, 756839, 859433, 1257787, 1398269,
2976221, 3021377, 6972593, 13466917, 20996011, 24036583,
25964951, 30402457, 32582657, 37156667, 42643801, 43112609
57885161, 74207281, 77232917, 82589933**

Mathematica-Programm zum Erzeugen der Mersenne-Primzahlen:

```
Flatten[Position[EulerPhi[2^#-]+2==EulerPhi[2^#]&/@Range[1,100],True]-1]
```

Der Bereich der ersten 48 Mersenne Primzahlen ist zu 100 Prozent getestet. Die Nummerierung der letzten vier Zahlen (49 bis 51) ist noch unsicher, d.h. es kann nicht ausgeschlossen werden, dass es in diesem Bereich noch weitere Mersenne Primzahlen gibt.

(Folge [A000043](#) in OEIS). Stand: Dez. 2020

Ungelöste Fragen zu den Mersenne-Primzahlen:

Gibt es unendlich viele Mersenne-Primzahlen? Alles deutet darauf hin, dass die Antwort ‚ja‘ lautet.

Gilt die ‚neue Mersennesche Vermutung‘?

Von folgenden drei Aussagen folgt jeweils die dritte, falls zwei richtig sind:

- 1) $n = 2^k \pm 1$ oder $n = 4^k \pm 3$
- 2) $2^n - 1$ ist prim (offensichtlich eine Mersenne-Primzahl)
- 3) $\frac{(2^n+1)}{3}$ ist prim

Gibt es unendlich viele zusammengesetzte Mersenne-Zahlen? Wahrscheinlich: Ja
Die Anzahl N_{M_p} der Mersenne-Primzahlen, die kleiner oder gleich N sind, verhält sich asymptotisch wie:

$$N_{M_p}(N) \sim \frac{e^\gamma}{\ln(2)} \ln \ln(N) \quad (17)$$

Graph:¹⁰

⁹ <http://www.mersenne.org/various/math.php>

¹⁰ <http://primes.utm.edu/notes/faq/NextMersenne.html>

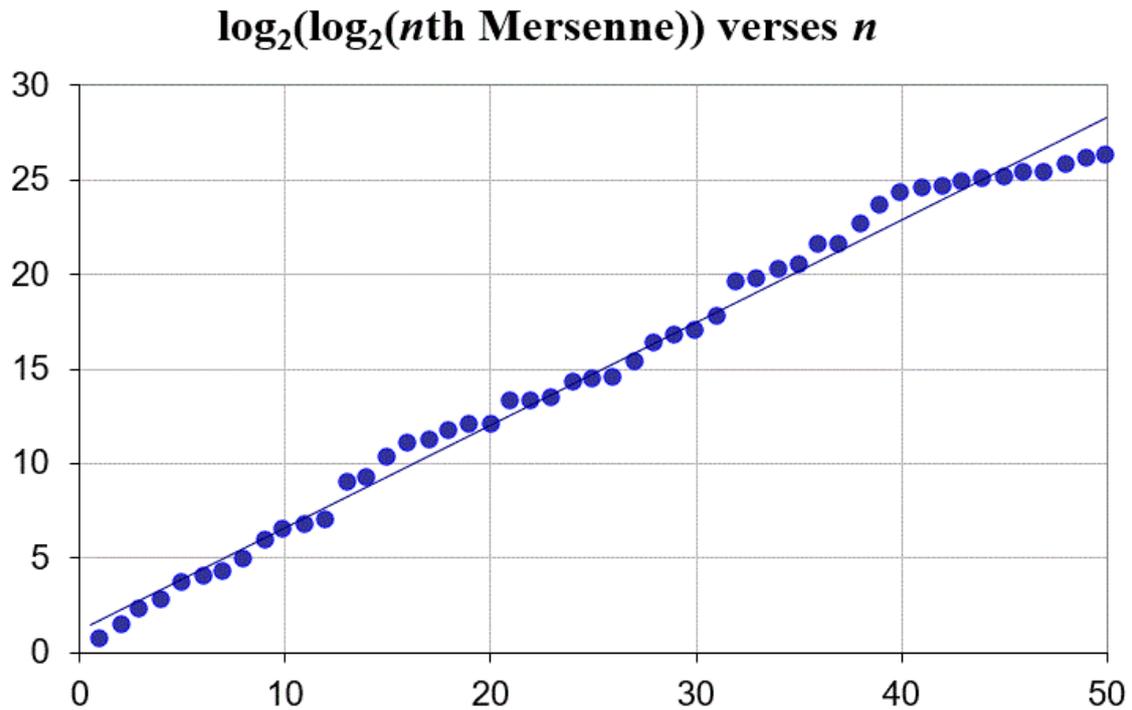


Abbildung 5: n-te Mersenne Primzahl (doppelt logarithmische Darstellung)

Man sieht, dass die asymptotische Abschätzung sehr gut passt.

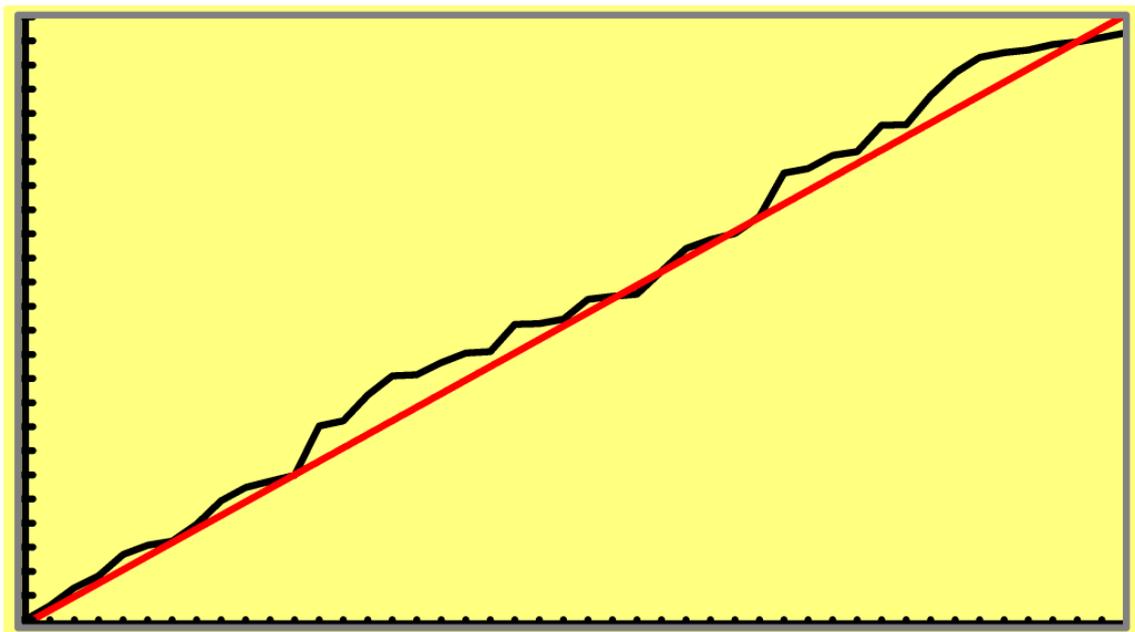


Abbildung 6: n-te Mersenne Primzahl (doppelt logarithmische Darstellung)

Abb.: Asymptotische Abschätzung (rot), $\ln(\ln(M_p))$ (schwarz)

Erstellt mit [KVEC](#) und folgender Parameter-Datei:

```
vnull  
MersennePrimesAsymptotic_KVEC.png
```

```

-antialias 2 -dimension 1024 -xdim 1025 -ydim 576
-format png -xmin 0.000000 -xmax 45.000000
-drcolor 0 0 0 -bkcolor 255 255 128 -nstep 2000 -lwidth 200
-scmode 2 -mode aniso -reduce all -smooth on
-function
imin 0; imax 49; drcolor 0 0 0;
f1(x)=log(KV_MPRIMES[x])/M_LN2;
drcolor 255 0 0;
f2(x)=exp(-M_G)*x+0.8255;
endfunc

```

Das Wenige, was wir über die analytische Mathematik der Mersenne-Primzahlen wissen und vermuten, ist hier gut beschrieben: <http://primes.utm.edu/notes/faq/>

Hier ein Plot einer Phasenraumdarstellung, der logarithmischen Mersenne-Primzahlen:¹¹

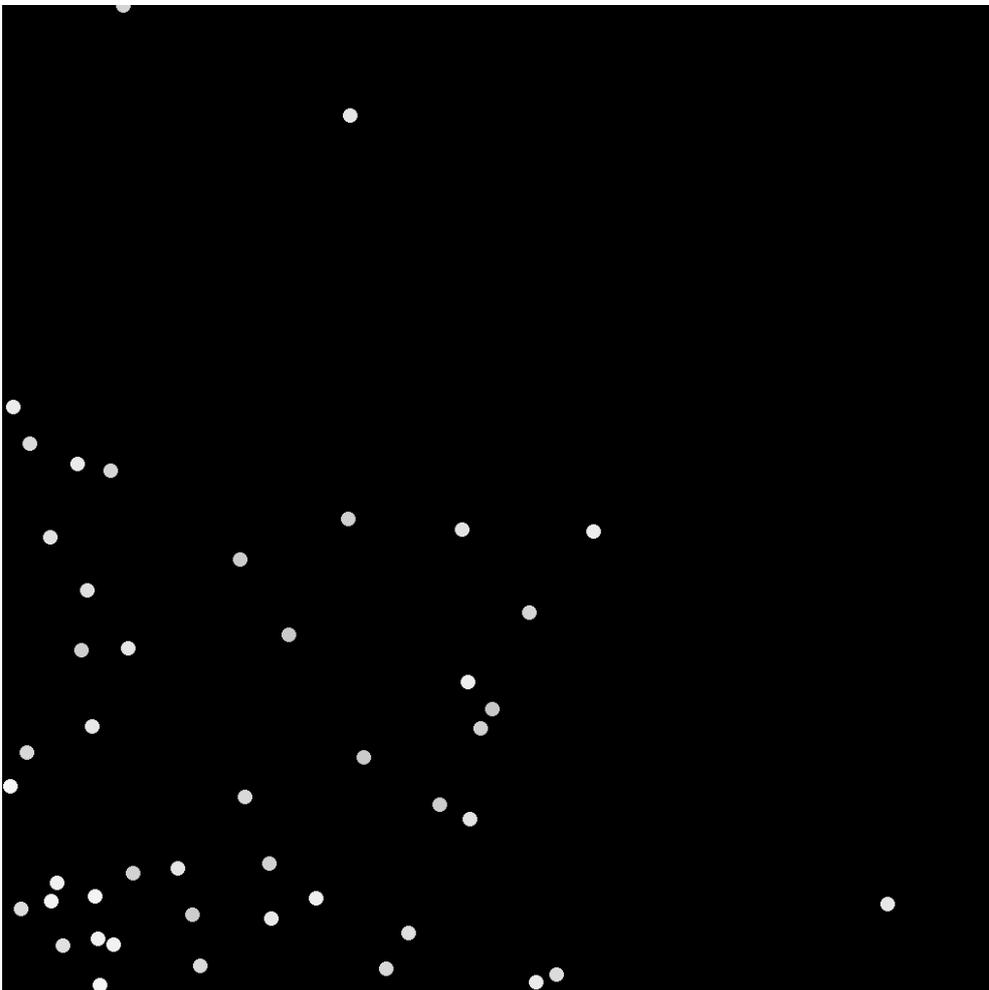


Abbildung 7: n-te Mersenne-Primzahl (doppelt logarithmische Phasenraumdarstellung))

Erzeugt mit KVEC und folgender Parameterdatei:

```
null
```

¹¹ Erstellt mit KVEC (<http://www.kvec.de>)

Mersenne-Primzahlen

```
Mersenne_Exponents_In_PhaseSpace.png  
-antialias 2 -dimension 1024 -format png -mode aniso -random 24 703
```

Man könnte darin symmetrische Strukturen erkennen. Wie mag das Bild wohl aussehen, wenn statt der ersten 51 Mersenne-Primzahlen die ersten 100 oder 1000 Mersenne-Primzahlen ausgewertet werden?

```
KVEC-Programm zum Erzeugen der ersten 51 Mersenne-Primzahlen:  
vnull  
(null).swf  
-debug plot -function imax 51; f1(i)=KV_MPRIMES[i]; endfunc
```

Und hier noch ein paar ‚Spielereien‘:

Lissajous-Figur, erzeugt mit allen Mersenne Primzahl Exponenten.

Das entsprechende KVEC-Programm lautet:

```
vnull  
plot_circles_MersennePrimes_Iteration.jpg  
-antialias 2 -xdim 847 -ydim 1025 -format jpeg  
-drcolor 50 0 24 -bkcolor 128 196 255 -nstep 500000 -grit 8 -scmode 2  
-paper user 600.000000 200.000000 -pattern outin 128 128 128 -function  
C1=0.9; x1=0.5; y1=0.25;  
object markfilledcircle;  
msize 0.1; imax 500000;  
x1()=(1.0-x1*y1*C1)*cos(log(KV_MPRIMES[II%49])+II);  
y1()=(x1-y1)*sin(log(KV_MPRIMES[II%49])-II);  
endfunc
```

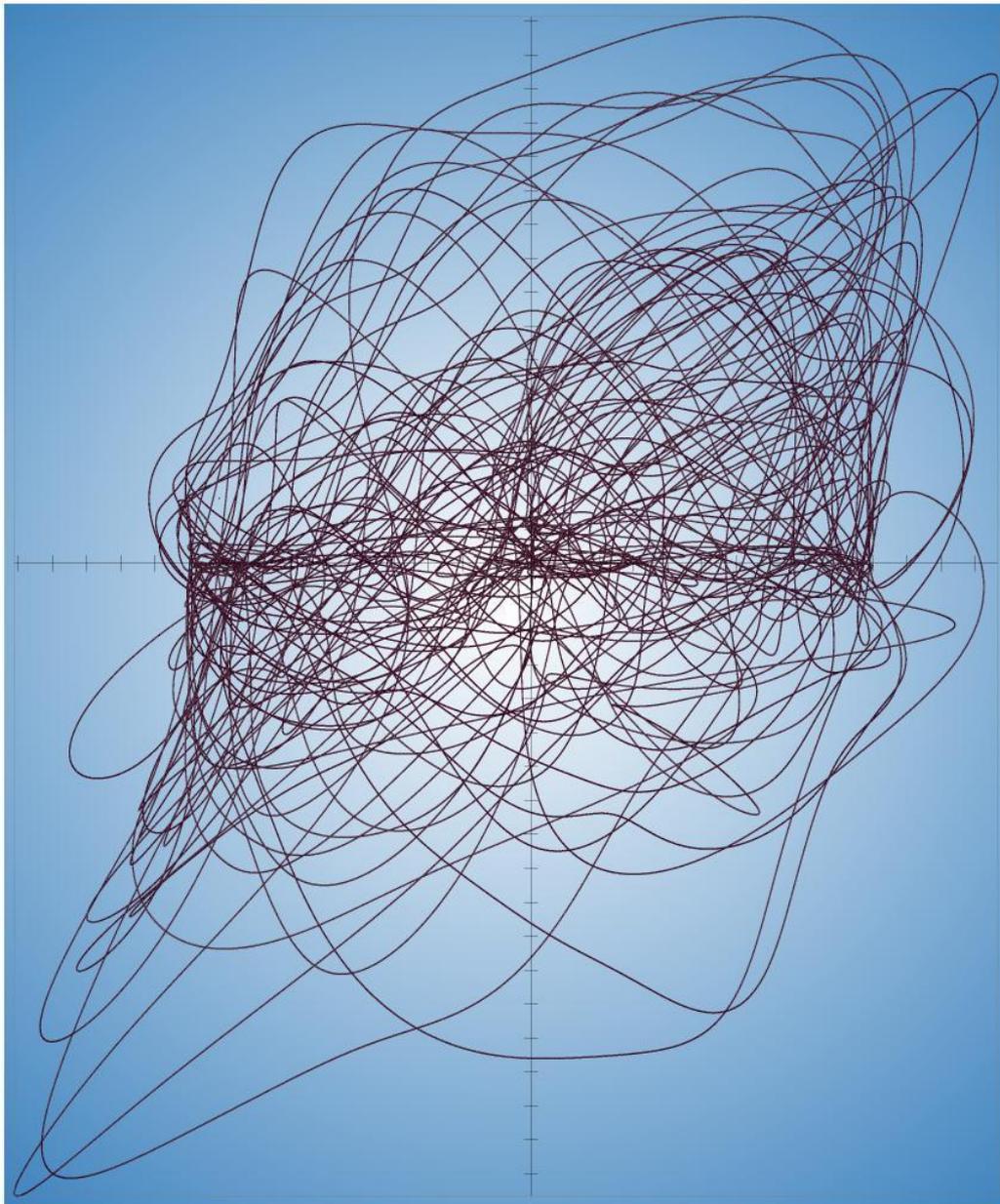


Abbildung 8: Lissajous-ähnliche Darstellung mit Hilfe von Mersenne-Primzahl-Exponenten

4.5.1 GIMPS – GREAT INTERNET MERSENNE PRIME SEARCH

GIMPS ist ein Projekt, bei dem die freiwillig zur Verfügung gestellte Rechnerleistung von Computernutzern dazu verwendet wird, um nach Mersenne-Primzahlen zu suchen.

In einem Buch über Primzahlen darf natürlich ein Hinweis auf dieses erfolgreiche Forschungsprojekt nicht fehlen!

Jeder, der an diesem Forschungsprojekt teilnehmen möchte kann, sich die entsprechende Software von der GIMPS-Seite herunterladen.¹²

¹² <http://www.mersenne.org>

Fermat-Primzahlen

Das Projekt war in den letzten Jahren sehr erfolgreich. Seit Bestehen des Projektes (1996) wurden mit GIMPS die 16 größten Mersenne-Primzahlen entdeckt.

Die gesamte Rechenleistung des Projektes beträgt zwischen 300 und 950 TFLOP/sec (Stand: Okt. 2016). Zur Erinnerung: Ein FLOP ist eine ‚Floating Point Operation‘ (eine Rechenoperation, an der Gleitkommazahlen beteiligt sind). Ein TFLOP/sec (=TeraFLOP/sec) bedeutet, dass 10^{12} Gleitkomma-Rechenoperationen pro Sekunde ausgewertet werden.

In Spitzenzeiten besitzt GIMPS somit eine Rechenleistung von fast einem PFLOP/sec (PetaFlop/sec = 10^{15} FLOP/sec = 1.000.000.000.000.000 FLOP/sec).

Hier die GIMPS-Statistik des Autors, der sich natürlich an diesem Projekt beteiligt (Stand: November 2016):

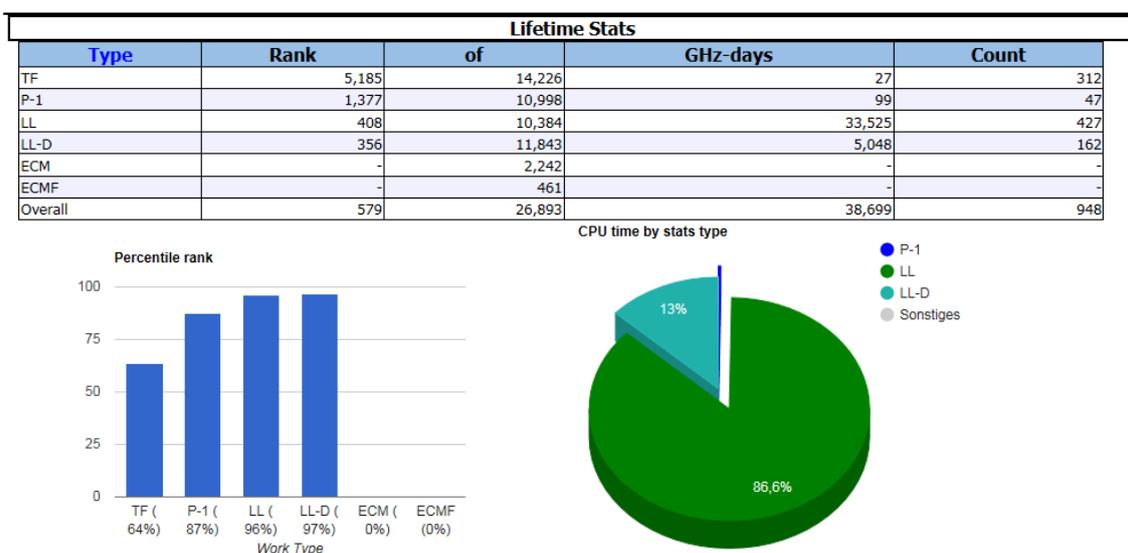


Abbildung 9: GIMPS-Statistik des Autors

4.6 FERMAT-PRIMZAHLEN

Zu den Fermat-Primzahlen gibt es ebenfalls eine unüberschaubare Menge an Literatur. Hier in Kürze das Wichtigste über die Fermat-Primzahlen. F_n :

Fermat-Primzahlen sind Primzahlen der Form

$$F_n = 2^{2^n} + 1 \quad (18)$$

Es ist seit Jahrhunderten bekannt, dass Zahlen der Form $2^m + 1$ nur dann Primzahlen sind, wenn m die Form 2^n hat. Leider ist jedoch nicht, wie Fermat glaubte, jede ‚Fermat-

Zahl $2^{2^n} + 1$ automatisch eine Fermat-Primzahl. Tatsächlich sind bis heute nur fünf dieser Zahlen bekannt, nämlich F_0 bis F_4 :

$$\mathbf{3, 5, 17, 257, 65537}$$

Alle weiteren Fermat-Zahlen sind zusammengesetzt.

F_5 bis F_{32} sind zusammengesetzte Zahlen. F_{33} ist die erste Fermat-Zahl von der man nicht weiß, ob sie zusammengesetzt oder prim ist. Stand: November 2016.

Bereits Euler zeigte 1732 dass jeder Faktor einer Fermat-Zahl F_n , ($n \geq 2$) die Form $k \cdot 2^{n+2} + 1$ haben muss.

Seit dem Jahr 1877 ist **Pepin's Primzahltest** für Fermat-Zahlen bekannt:

Folgende Bedingungen sind gleichwertig (meist wird für k der Wert 3 genommen):

- F_n ist eine Primzahl und $(k|F_n) = -1$ mit: $(k|F_n)$ Jacobi-Symbol¹⁴
- $k^{(F_n-1)/2} \equiv -1 \pmod{F_n}$

Ein paar grundlegende Eigenschaften von Fermat-Primzahlen:¹⁵

$$F_n = (F_{n-1} - 1)^2 + 1, \quad \text{für } n \geq 1$$

Sowie (für $n \geq 2$):

$$F_n = F_{n-1} + 2^{2^{n-1}} F_0 \cdots F_{n-2}$$

$$F_n = F_{n-1}^2 - 2(F_{n-2} - 1)^2$$

$$F_n = F_0 \cdots F_{n-1} + 2$$

Jede Fermat-Zahl endet (im Dezimalsystem) mit der Ziffer 7 (mit Ausnahme der ersten beiden).

Fermat-Primzahlen gehören nicht zu den sogenannten ‚Brasilianischen‘ Zahlen; das sind Primzahlen der Form:

$$P_k = 1 + n + n^2 + n^3 + \cdots + n^k, \quad n > 1, k > 1 \tag{19}$$

Hinweis: Es ist nicht bekannt, ob es unendlich viele Brasilianische Primzahlen gibt...

Mathematica-Programm zum Erzeugen Der Fermat-Primzahlen:
`Select[Table[2^(2^n) + 1, {n, 0, 4}], PrimeQ]`

Die Anzahl der Ziffern $D(n)$ einer Fermatzahl F_n (im Dezimalsystem) beträgt:

¹³ Sequenz <https://oeis.org/A000215>

¹⁴ <http://mathworld.wolfram.com/JacobiSymbol.html>

¹⁵ https://en.wikipedia.org/wiki/Fermat_number

$$D(n) = 1 + \lfloor 2^n \ln 2 \rfloor$$

Bemerkenswert ist auch der Zusammenhang zwischen der Konstruierbarkeit eines regelmäßigen n-Ecks mit Zirkel und Lineal und den Fermat-Primzahlen, den Gauß schon 18.ten Jahrhundert entdeckt hat:

Ein regelmäßiges n-Eck ist mit Zirkel und Lineal genau dann konstruierbar, wenn es das Produkt einer Potenz von 2 mit paarweise voneinander verschiedenen Fermatschen Primzahlen ist.

Merkwürdig ist, dass z. B. ein regelmäßiges 5-Eck oder 17-Eck konstruierbar ist, nicht jedoch ein 7-Eck oder 11-Eck...

4.7 GLÜCKLICHE PRIMZAHLEN

Glückliche Zahlen dürfen nicht mit den ‚fröhlichen‘ Zahlen (engl.: ‚Happy‘-numbers) verwechselt werden, die vollkommen anders definiert werden.¹⁶

Zunächst einmal sei hier die Definition der ‚Glücklichen‘ Zahlen vorgestellt¹⁷. ‚Glückliche‘ Zahlen entstehen nach einem Sieb-Verfahren, das dem ‚Sieb des Eratosthenes‘ ähnelt:

Wir betrachten die Liste der natürlichen Zahlen 1,2,3,4,5,6,...und streichen aus dieser Liste nach folgendem Prinzip:

- Die 1 ist per Definition ‚glücklich‘:
(01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...)
 - Die nächste Zahl ist 2, also streichen wir jede 2-te Zahl, es bleibt die 3:
(01, 03, 05, 07, 09, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, ...)
 - Die nächste Zahl ist 3, also streichen wir jede 3-te Zahl, es bleibt die 7:
(01, 03, 07, 09, 13, 15, 19, 21, 25, 27, 31, 33, 37, 39, 43, 45, 49, 51, 55, 57, ...)
 - Die nächste Zahl ist 7, also streichen wir jede 7-te Zahl, es bleibt die 9:
(01, 03, 07, 09, 13, 15, 21, 25, 27, 31, 33, 37, 43, 45, 49, 51, 55, 57, ...)
 - Die nächste Zahl ist 9, also streichen wir jede 9-te Zahl, es bleibt die 13:
(01, 03, 07, 09, 13, 15, 21, 25, 31, 33, 37, 43, 45, 49, 51, 55, ...)
- ... usw.

Es bleibt die Liste der glücklichen Zahlen übrig.

Die Menge der glücklichen Zahlen hat viel mit der Menge der Primzahlen gemeinsam.

Sie besitzen beide die gleiche Dichte, die proportional zu $\frac{1}{\ln(n)}$ ist. Primzahl-Zwillinge und ‚glückliche‘ Zwillinge scheinen die gleiche Dichte zu besitzen, wie folgende Tabelle vermuten lässt:

¹⁶ <http://mathworld.wolfram.com/HappyNumber.html>

¹⁷ <https://oeis.org/A000959>

Tabelle 7: Glückliche Zahlen bis 1E15

Bereich	Anzahl Glückl. Zahlen	Anzahl Primzahlen	Anzahl Glücks Zwillinge	Anzahl Primzahl Zwillinge
10^0	1	0	1	0
10^1	4	4	2	2
10^2	23	25	7	8
10^3	153	168	33	35
10^4	1118	1229	178	205
10^5	8772	9592	1162	1224
10^6	71918	78498	7669	8169
10^7	609237	664579	55548	58980
10^8	5286238	5761455	419174	440312
10^9	46697909	50847534	3274570	3424506
10^{10}	418348044	455052511	26298112	27412679
10^{11}	3790060378	4118054813	?	224376048
10^{12}	34652117969	37607912018	?	1870585220
10^{13}	319239995375	346065536839	?	15834664872
10^{14}	2960006060823	3204941750802	?	135780321665
10^{15}	27596305747873	29844570422669	?	1177209242304

Mathematica:

```

luckies=2*Range@500-1;
f[n_]:=Block[{k=luckies[[n]]},luckies=Delete[luckies,Table[{k},{k,k,Length@luckies,k}]]];Do[f@n,{n,2,30}];luckies
(*oder:*)
sieveMax = 10^6; luckies = Range[1, sieveMax, 2];
sieve[n_] := Module[{k = luckies[[n]]}, luckies = Delete[luckies,
Table[{i}, {i, k, Length[luckies], k}]]; n = 1; While[luckies[[n]] <
Length[luckies], n++; sieve[n]]; luckies

```

ergibt:

```

{1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63, 67, 69, 73, 75, 79, 87,
93, 99, 105, 111, 115, 127, 129, 133, 135, 141, 151, 159, 163, 169, 171, 18
9, 193, 195, 201, 205, 211, 219, 223, 231, 235, 237, 241, 259, 261, 267, 27
3, 283, 285, 289, 297, 303, 307, 319, 321, 327, 331, 339, 349, 357, 361, 36
7, 385, 391, 393, 399, 409, 415, 421, 427, 429, 433, 451, 463, 475, 477, 48
3, 487, 489, 495, 511, 517, 519, 529, 535, 537, 541, 553, 559, 577, 579, 58
3, 591, 601, 613, 615, 619, 621, 631, 639, 643, 645, 651, 655, 673, 679, 68
5, 693, 699, 717, 723, 727, 729, 735, 739, 741, 745, 769, 777, 781, 787, 80
1, 805, 819, 823, 831, 841, 855, 867, 873, 883, 885, 895, 897, 903, 907, 92
5, 927, 931, 933, 937, 957, 961, 975, 979, 981, 987, 991, 993, 997}

```

Hinweis: Dieser Typ von ‚glücklichen‘ Zahlen ist nicht zu verwechseln

mit den ‚glücklichen‘ Zahlen von Euler (Primzahlen n der Form $m^2 - m + n$ so dass $m^2 - m + n$ eine Primzahl ergibt, für $m = 0, 1, \dots, n - 1$).

Die Menge der ‚glücklichen‘ Primzahlen ist nun einfach die Menge der ‚glücklichen‘ Zahlen, die prim sind.¹⁸ Es ist unbekannt, ob es unendlich viele ‚glückliche‘ Primzahlen gibt (Stand Oktober 2015).

4.8 VOLLKOMMENE ZAHLEN

4.8.1 ALLGEMEINES UND DEFINITION

Vollkommene Zahlen sind eng mit den Mersenne- Primzahlen verknüpft (siehe Kapitel 4.5).

Definition:

Eine (positive ganze) Zahl ist vollkommen, wenn sie identisch mit der Summe ihrer Teiler ist (wobei die Zahl selbst als Teiler ausgenommen wird). Diese Teilersumme wird oft als Aliquot-Summe $S(n)$ bezeichnet, im Gegensatz zur vollständigen Teilersumme $\sigma_1(n)$, bei der die Zahl selbst zur Summe mitgezählt wird. Daraus folgt:

$$\text{Eine Zahl } n \text{ ist vollkommen, wenn } S(n) = n \text{ oder } \sigma_1(n) = 2n \quad (20)$$

Vollkommene Zahlen sind seit dem Altertum (Nicomachus¹⁹, Philo Judaeus²⁰) bekannt (genauer sind dies die vier Zahlen **6, 28, 496, 8128**) und wurden schon in den ‚Elementen‘ des griechischen Mathematikers Euklid erwähnt. Wahrscheinlich kommt die Bezeichnung von der Vorstellung, dass Gott die Welt in 6 Tagen erschuf, sowie ein Mondumlauf 28 Tage beträgt.

Es gibt so viele bekannte vollkommene Zahlen wie es bekannte Mersenne-Primzahlen gibt (Stand Dez. 2020). Die ersten 10 lauten²¹:

¹⁸ <https://oeis.org/A031157>

¹⁹ Nicomachus (60 – 120 n. Chr.), Antiker Philosoph, Musiktheoretiker und Mathematiker

²⁰ Philo Judaeus: (25-50 n. Chr.), griechisch-jüdischer Philosoph, lebte in Alexandria

²¹ https://en.wikipedia.org/wiki/Perfect_number

Tabelle 8: Die ersten 10 vollkommene Zahlen

n	Vollkommene Zahl
1	6
2	28
3	496
4	8128
5	33550336
6	8589869056
7	137438691328
8	2305843008139952128
9	2658455991569831744654692615953842176
10	191561942608236107294793378084303638130997321548169216

Alle bekannten vollkommenen Zahlen hängen mit den Mersenne Primzahlen zusammen (der Beweis stammt von L. Euler aus dem 18. Jahrhundert).

Wenn $2^p - 1$ prim ist, dann ist $2^{p-1}(2^p - 1)$ eine perfekte Zahl (21)

Alle derzeit bekannten vollkommenen Zahlen sind gerade. Es sind 51 vollkommene Zahlen bekannt (Stand: Dez. 2020). Es ist unbekannt, ob es ungerade vollkommene Zahlen gibt. Wahrscheinlich gibt es unendlich viele vollkommene Zahlen (so wie es wahrscheinlich unendlich viele Mersenne-Primzahlen gibt).

Vollkommene Zahlen spielen auch in der Zahlenmystik und Numerologie eine Rolle.

4.8.2 EIGENSCHAFTEN

Jede gerade vollkommene Zahl lässt sich wie folgt darstellen:

$$n = 1 + \frac{9}{2}k(k+1), \quad (\text{mit } k = 8j + 2, \quad j > 0 \text{ und } n > 6) \quad (22)$$

Die Umkehrung gilt nicht! Man erhält nicht für jedes j eine vollkommene Zahl....

Für $j = 1, 2, \dots$ erhalten wir: 28,496,1540,3160,5356,8128,11476,15400,19900,...

Nur für folgende j erhalten wir vollkommene Zahlen

Mathematica-Programm zur Berechnung der Indices, die vollkommene Zahlen liefern:

```
MPrimeExp={2,3,5,7,13,17,19,31,61,89,107,127,521,607,1279,2203,2281,3217,4253,4423,9689,9941,11213,19937,21701,23209,44497,86243,110503,132049,216091,756839,859433,1257787,1398269,2976221,3021377,6972593,13466917,20996011,24036583,25964951,30402457,32582657}
```

Vollkommene Zahlen

```
PerfectN=Table[2^(MPrimeExp[[k]]-1)(2^MPrimeExp[[k]]-1),{k,1,20}]
```

```
Table[Solve[1+9/2(8j+2)(8j+3)==PerfectN[[i]],j],{i,2,10}]
```

Ergibt:

```
{{j->-(5/8)}, {j->0}}, {{j->-(13/8)}, {j->1}},  
{{j->-(45/8)}, {j->5}},  
{{j->-(2733/8)}, {j->341}},  
{{j->-(43693/8)}, {j->5461}},  
{{j->-(174765/8)}, {j->21845}},  
{{j->-(715827885/8)}, {j->89478485}},  
{{j->-(768614336404564653/8)}, {j->96076792050570581}},  
{{j->-(206323339880896712483187373/8)}, {j->25790417485112089060398421}}
```

Die Folge $\{0, 1, 5, 341, 5461, \dots\}$ ist die Folge aller n , für die $24n + 7$ eine Mersenne-Primzahl ergibt.

Weitere Eigenschaften:

- Die Summe der Kehrwerte aller Teiler einer vollkommenen Zahl n ergibt 2:

$$\sum_{k|n} \frac{1}{k} = 2$$

- Jede vollkommene Zahl $n > 6$ lässt sich darstellen als Summe von 3-ten Potenzen:

$$n = \sum_{i=1}^{2^{\frac{p-1}{2}}} (2i-1)^3, \text{ wobei } n = 2^{p-1}(2^p-1)$$

- Jede vollkommene Zahl lässt sich mit einem geeigneten k auch darstellen als:

$$n = \sum_{i=1}^k i = \frac{k(k+1)}{2}$$

Beispiele: $6 = 1 + 2 + 3 = \frac{3 \cdot 4}{2}$, $28 = 1 + 2 + 3 + 4 + 5 + 6 + 7 = \frac{7 \cdot 8}{2}$

Es gibt zwei Arten der Verallgemeinerung des Begriffes ‚Vollkommene Zahl‘:

- 1) Ergibt die echte Teilersumme das n -fache der Zahl selbst, so nennt man diese Zahl ‚ n -vollkommen‘. Beispiel: 120 ist ‚2-vollkommen‘
- 2) Wenn p und $p^k - m - 1$ Primzahlen sind, dann hat die Gleichung

$$\sigma_1(x) = \frac{px + m}{p - 1}$$

die Lösung $x = p^{k-1}(p^k - m - 1)$.

Ungerade vollkommene Zahlen:

Es sind keine ungeraden vollkommenen Zahlen bekannt. Es ist auch nicht bekannt, ob irgendwelche von ihnen existieren.

Die Suche nach solchen Zahlen n hat schon große Fortschritte gemacht. Hier der aktuelle Stand der Ergebnisse (Oktober 2015). Falls solche Zahlen n existieren, müssen sie folgende Bedingungen erfüllen:

- $n > 10^{1500}$
- 105 ist kein Teiler von n
- n hat die Form $n \equiv 1 \pmod{12}$ oder $n \equiv 117 \pmod{468}$ oder $n \equiv 81 \pmod{324}$
- Der größte Primfaktor von n ist größer als 10^8
- n besitzt mindestens 101 Primfaktoren

Es ist somit sehr unwahrscheinlich, dass ungerade vollkommene Zahlen existieren.

4.9 SOPHIE-GERMAIN-PRIMZAHLEN

Eine Primzahl p heißt *Sophie-Germain-Primzahl* wenn $2p + 1$ auch eine Primzahl ist. Die Zahlen $2p + 1$ werden auch „*Sichere Primzahlen*“ genannt. Sie sind auch Lösungen der Gleichung (wobei $\varphi(n)$ die Eulersche Phi-Funktion, auch Totient-Funktion genannt ist):

$$\varphi(n) = 2p \quad (23)$$

Es gilt folgender Satz: Wenn p eine *Sophie-Germain-Primzahl* ist, dann gibt es keine ganzen Zahlen x, y und z aus \mathbb{Z} (ohne 0) so dass p kein Teiler von $x \cdot y \cdot z$ ist und die Gleichung $x^p + y^p = z^p$ erfüllt ist.

Hinweis: Bei den Lösungen der Fermatschen Gleichung $x^n + y^n = z^n$ unterscheidet man zwei Fälle. Im ersten Fall ist n kein Teiler von x, y oder z . D. h. für Primzahlen vom Typ *Sophie-Germain* ist der erste Fall von Fermats Satz richtig.²²

Die ersten *Sophie-Germain-Primzahlen* lauten:

2, 3, 5, 11, 23, 29, 41, 53, 83, 89, 113, 131, 173, 179,
191, 233, 239, 251, 281, 293, 359, 419, 431, 443, 491, 509,
593, 641, 653, 659, 683, 719, 743, 761, 809, 911, 953,

²² Fermats Satz: Es gibt keine ganzzahligen Lösungen von $x^n + y^n = z^n$ für $n > 2$.

Sophie-Germain-Primzahlen

1013, 1019, 1031, 1049, 1103, 1223, 1229, 1289, 1409, 1439,
1451, 1481, 1499, 1511, 1559

4.9.1 BERECHNUNG UND EIGENSCHAFTEN

Eigenschaften von *Sophie-Germain-Primzahlen*:

- 1) Wenn $p > 3$ eine *Sophie-Germain-Primzahl* ist und es gilt $p \equiv 3 \pmod{4}$, dann ist $2p + 1$ ein Teiler der p -ten Mersenne-Zahl.
- 2) Für alle *Sophie-Germain-Primzahlen* gilt: $p \equiv 3 \pmod{4}$
- 3) Im Dezimalsystem können *Sophie-Germain-Primzahlen* niemals die Endziffer 7 haben.
- 4) p und $2p + 1$ sind dann und nur dann *Sophie-Germain-Primzahlen*, wenn p eine Primzahl ist und $2^{2p} \equiv 1 \pmod{2p + 1}$

Asymptotische Abschätzung der Anzahl von SG-Primzahlen bis zu einer oberen Grenze N :

$$\text{Anzahl}_{SG} = 2C_2 \int_2^N \frac{1}{\ln(x) \ln(2x + 1)} dx \approx \frac{2C_2 N}{\ln^2(N)} \quad (24)$$

(mit der Primzahl-Zwillingskonstanten $C_2 = 0,6601618158$)

Berechnung mit Mathematica: (z. B. im Intervall von 1 bis 1000):

```
Select[Prime[Range[1000]], PrimeQ[2#+1]&]
```

Vermutungen:

- 1) Es gibt unendlich viele *Sophie-Germain-Primzahlen*
- 2) Zwischen n und $2n$ gibt es immer eine *Sophie-Germain-Primzahl*.

Rekord:

Die derzeit größte SG-Primzahl lautet:

18543637900515 · 2⁶⁶⁶⁶⁶⁷ - 1,

eine Zahl mit 200701 Stellen (Stand November 2015).

Hinweis: Folgen von SG-Primzahlen nennt man in der Literatur Cunningham-Ketten der ersten Art²³

²³ <https://de.wikipedia.org/wiki/Cunningham-Kette>

Anzahl $a(n)$ der SG-Primzahlen bis 10^n :

Tabelle 9: Anzahl der Sophie-Germain-Primzahlen bis $1E12$

n	a(n)
1	3
2	10
3	37
4	190
5	1171
6	7746
7	56032
8	423140
9	3308859
10	26569515
11	218116524
12	1822848478

Berechnung mit Mathematica (Beispiel):

```
Accumulate[Table[Boole[PrimeQ[n]&&PrimeQ[2n+1]], {n, 1, 200}]]
```

4.10 FIBONACCI-ZAHLEN UND ANDERE REKURSIVE FOLGEN

Zu den Fibonacci-Zahlen gibt es eine unüberschaubare Menge an Literatur und Veröffentlichungen. Auf eine Übersicht wird hier verzichtet. Interessantes findet man auf Michael Beckers Homepage.²⁴

An Kuriositäten seien hier nur folgende Formeln erwähnt:

$$\frac{1}{F_{11}} = \frac{1}{89} = 0.01123595 \text{ (Dezimalbruchentwicklung beginnt mit Fibonacci-Folge)}$$

Genauer müsste man hier schreiben:

$$\frac{1}{F_{11}} = \sum_{k=0}^{\infty} \frac{F_k}{10^{k+1}} \quad (25)$$

Der Quotient $\frac{F_{n+1}}{F_n}$ zweier aufeinander folgender Fibonacci-Zahlen ist der n -te Näherungsbruch des Kettenbruches:

²⁴ <http://www.ijon.de/mathe/fibonacci/node2.html#0002320>

$$1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \dots}}} \quad (26)$$

Der Grenzwert ist die Zahl des Goldenen Schnittes: $\Phi = 1.618 = \frac{1+\sqrt{5}}{2}$.

$$F_{12} = 144 = 12^2$$

F_{12} ist die einzige Quadratzahl unter den unendlich vielen Fibonacci-Zahlen. Man mag sich fragen, ob es dafür einen tieferen Grund gibt, denn Zufall kann es ja wohl nicht sein. Den Grund gibt es tatsächlich. Er erscheint als Nebenprodukt in E. Wiles Beweis des Fermatschen Satzes. Aber das ist eine andere Geschichte ☺

Eine Kuriosität ist auch die Formel

$$\left\lfloor e^{\frac{n-1}{2}} \right\rfloor, n = 0, 1, 2 \dots$$

Sie liefert hintereinander für $n = 1, \dots, 10$ die ersten 10 Fibonacci-Zahlen.

Mathematica: `Table[Floor[Exp[(n-1)/2]]+1, {n, 1, 25}]`
`{1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 91, 149, 245, 404, 666, 1097, 1809, 2981, 4915, 8104, ...}`

Für den Praktiker: Die n -te Fibonacci-Zahl läßt sich ganz einfach berechnen:

$$F_n = \text{Round} \left(\frac{\Phi^n}{\sqrt{5}} \right) \quad (27)$$

Schließlich zeigte J. P. Jones im Jahr 1975, dass es ein Polynom 5.ten Grades mit zwei Variablen gibt, dessen positive Wertemenge (bei nicht-negativen Argumenten) mit der Menge der Fibonacci-Zahlen identisch ist. Diese bemerkenswerte Tatsache ist jedoch eher von theoretischem Interesse. Immerhin lässt sich die Aussage mit folgendem Mathematica-Programm testen:

```
numbersOfInstances=6;
polynom[x_, y_] := 2x*y^4 + x^2*y^3 - 2x^3*y^2 - y^5 - x^4*y + 2*y;
list=FindInstance[Reduce[polynom[x, y]>0&& x>=0&& y>=0, {x, y}, Integers], {x, y}, Integers, numbersOfInstances, RandomSeed->112]
Table[polynom[list[[n]][[1]][[2]], list[[n]][[2]][[2]]], {n, 1, Length[list]}]
```

Folgende merkwürdige Eigenschaft hat der Autor noch nirgends gelesen, wahrscheinlich ist es jedoch schon bekannt:

Sei:

$$\text{poly}_{Fib}(x, y) = 2xy^4 + x^2y^3 - 2x^3y^2 - y^5 - x^4y + 2y \quad \text{mit } x, y \in \mathbb{N}_0$$

Dann ist:

$$\text{poly}_{Fib}(F_i, F_{i+1}) = F_{i+1} \quad (i \geq 0) \quad (28)$$

D.h. die Argumente, für die das Polynom die (positiven) Fibonacci-Zahlenwerte annimmt, sind gerade die Fibonacci-Zahlen selbst! Daraus ergibt sich dann folgender Zusammenhang:

$$F_{i+1} = 2F_i F_{i+1}^4 + F_i^2 F_{i+1}^3 - 2F_i^3 F_{i+1}^2 - F_{i+1}^5 - F_i^4 F_{i+1} + 2F_{i+1} \quad (29)$$

Oder anders formuliert, der ‚Nachfolger‘ F_{i+1} in der Fibonacci-Folge lässt sich aus dem Vorgänger F_i über die positive, ganzzahlige Lösung der Gleichung 4.ten Grades berechnen, und das ohne Kenntnis des Indexes i :

$$-y^4 + 2F_i y^3 + F_i^2 y^2 - 2F_i^3 y + 1 - F_i^4 = 0 \quad (30)$$

Mathematica-Beispiel: $f = F_i = 8$;

```
Solve[-y^4+2f y^3+f^2y^2-2f^3y+1-f^4==0, y, Integers]
{{y->-5}, {y->13}}
```

Folgende Vermutung ist sicher leicht zu beweisen (wenn sie wahr ist...):

(30) hat für positive F_i immer reelle Lösungen. Ganzzahlige Lösungen existieren nur wenn F_i eine Fibonacci-Zahl ist.

Hinweis: Natürlich lässt sich aus der expliziten Formel für F_n (z. B. mit Mathematica) der Wert von n ausrechnen und mit Einsetzen von $n + 1$ in die explizite Formel auch der Nachfolger F_{n+1} von F_n (ohne Kenntnis von n) bestimmen. Diese Prozedur ist jedoch sehr umständlich und nicht so elegant wie mit der Gleichung 4.ten Grades (30).

Für den Praktiker: Mit folgender Formel geht es noch schneller (für $n > 1$):

$$F_{n+1} = \text{Round}(F_n \Phi), \text{ mit } \Phi = 1.618 = \frac{1+\sqrt{5}}{2} \text{ und } \text{Round}(x) = \lfloor x + 0.5 \rfloor \quad (31)$$

Umgekehrt lässt sich n ganz leicht berechnen:

$$n = \text{Round}\left(\frac{\ln F_n + \frac{\ln 5}{2}}{\ln \Phi}\right) \quad (32)$$

Ein einfacher Test, um herauszufinden, ob eine Zahl eine Fibonacci-Zahl ist:

n ist dann und nur dann eine Fibonacci-Zahl, wenn $5n^2 + 4$ oder $5n^2 - 4$ eine Quadratzahl ist.

Fibonacci-Zahlen und andere rekursive Folgen

Schließlich sei noch erwähnt, dass die letzte Ziffer der Fibonacci-Zahlen sich mit einer Periode von 60 wiederholt (auch für die letzten n Ziffern gibt es Perioden, deren Länge pro zusätzliche Ziffer um einen Faktor 5 wächst).

Die Fibonacci-Folge wird erstmals im Jahr 450 v. Chr. im „*Chandah-shāstra*“, einer in Sanskrit verfassten Schrift erwähnt, wurde später jedoch bekannt seit dem Jahr 1202 durch Leonardo Fibonacci²⁵, der in seinem Buch „*Liber Abaci*“ (Buch der Rechenkunst) damit die Vermehrung der Population bei Kaninchen beschrieb:

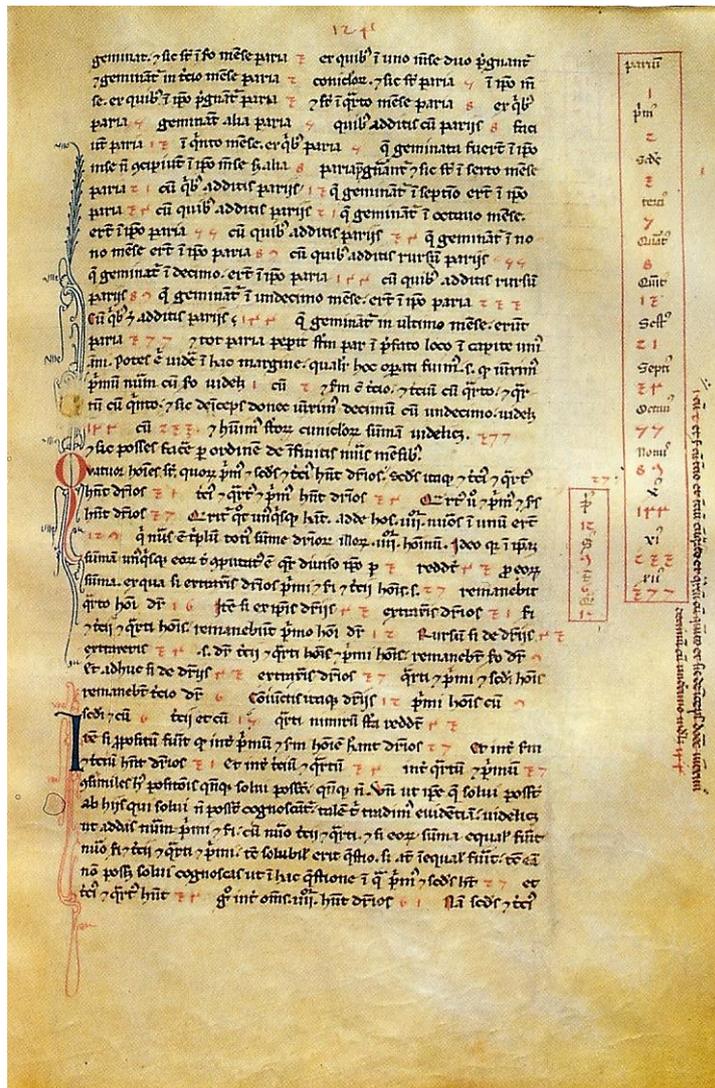


Abbildung 10: Seite aus „*Liber Abaci*“ von Leonardo Fibonacci

Sie ist bei Esoterikern, Weltverschwörern bestens bekannt. Sie taucht in etlichen Science-Fiction und Fantasie-Filmen auf (,Sakrileg“²⁶, 2004)

Die Fibonacci-Zahlen gehören zu dem Typ von rekursiv-definierten Folgen (genauer: linear rekursiv). Deshalb hier ein kurzer Einschub:

²⁵ https://de.wikipedia.org/wiki/Leonardo_Fibonacci

²⁶ [https://de.wikipedia.org/wiki/Sakrileg_\(Roman\)](https://de.wikipedia.org/wiki/Sakrileg_(Roman))

4.10.1 LINEARE REKURSION: EIN MÄCHTIGES INSTRUMENT

Die Methode der linearen Rekursion als Konstruktionsprinzip für arithmetische Folgen liefert viele interessante Folgen, die zum Gegenstand ausgedehnter mathematischer Untersuchungen gemacht wurden. Hier nur ein paar Beispiele: Die Fibonacci- und die verwandte Lucas-Folge, die Perrin-Folge (auch ‚Skiponacci‘-Folge genannt, siehe Kapitel 20.3), sowie die Pell-Folge.

Die Literatur über diese Folgen ist unüberschaubar. An dieser Stelle seien hier ein paar wichtige Ergebnisse genannt, die dem Autor interessant erscheinen.

Linear rekursiv definierte Folgen sind definiert durch eine lineare Beziehung ihrer Folgenglieder mit den vorausgegangenen Folgengliedern:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}, \text{ mit den Initialwerten } a_0, a_1, \dots, a_{k-1} \quad (33)$$

Tabelle 10: Einige linear rekursiv definierten Folgen

Rekursion	Initialwerte	Kernel	Expl. Formel	Name
$a_n = a_{n-1} + a_{n-2}$	$a_0 = 0, a_1 = 1$	$\{1, 1\}$	$\frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right]$	Fibonacci
$a_n = a_{n-1} + a_{n-2}$	$a_0 = 2, a_1 = 1$	$\{1, 1\}$	$\left(\frac{1+\sqrt{5}}{2} \right)^n + \left(\frac{1-\sqrt{5}}{2} \right)^n$	Lucas
$a_n = 2a_{n-1} + a_{n-2}$	$a_0 = 0, a_1 = 1$	$\{2, 1\}$	$\frac{(1+\sqrt{2})^n - (1-\sqrt{2})^n}{2\sqrt{2}}$	Pell
$a_n = 2a_{n-1} + a_{n-2}$	$a_0 = 2, a_1 = 2$	$\{2, 1\}$	$(1+\sqrt{2})^n + (1-\sqrt{2})^n$	Pell-Lucas
$a_n = a_{n-2} + a_{n-3}$	$a_0 = 1, a_1 = 1, a_2 = 1$	$\{0, 1, 1\}$	(kompliziert, siehe 20.6)	Padovan
$a_n = a_{n-2} + a_{n-3}$	$a_0 = 3, a_1 = 0, a_2 = 2$	$\{0, 1, 1\}$	(siehe Kapitel 20.6)	Perrin
$a_n = a_{n-1} + a_{n-2} + a_{n-3}$	$a_0 = 0, a_1 = 1, a_2 = 2$	$\{1, 1, 1\}$	(kompliziert)	‚Tribonacci‘
$a_n = a_{n-1} + a_{n-2} + a_{n-3} + a_{n-4}$	$a_0 = 0, a_1 = 1, a_2 = 2, a_3 = 4$	$\{1, 1, 1, 1\}$	(kompliziert)	‚Quadranacci‘
$a_n = a_{n-5} + a_{n-2}$	$a_0 = 5, a_1 = 0, a_2 = 2, a_3 = 0, a_4 = 2$	$\{0, 1, 0, 0, 1\}$?	‚5‘+Sloane A133394
$a_n = a_{n-5} + a_{n-2}$	$a_0 = 0, a_1 = 2, a_2 = 0, a_3 = 2, a_4 = 5$	$\{0, 1, 0, 0, 1\}$?	Reed Jameson
$a_n = a_{n-5} - a_{n-3}$	$a_0 = 5, a_1 = 0, a_2 = 0, a_3 = -3, a_4 = 0$	$\{0, 0, -1, 0, 1\}$?	Sloane A136598
$a_n = a_{n-7} + a_{n-4}$	$a_0 = 7, a_1 = a_2 = a_3 = 0, a_4 = 4, a_5 = a_6 = 0$	$\{0, 0, 0, 1, 0, 0, 1\}$?	Sloane A135435 Reed Jameson

Mathematica bietet die Funktionen `LinearRecurrence[kernel, init, n]`, `RecurrenceTable[]` und `FindLinearRecurrence[list]`, mit denen lineare rekursive Folgen untersucht werden können.

Mit `RSolve` und `RSolveValue` können rekursive Gleichungen gelöst werden, z.B.:

Fibonacci-Zahlen und andere rekursive Folgen

```
RSolve[{f[n]==f[n-1]+f[n-2], f[0]==0, f[1]==1}, f[n], n]
{{f[n]->Fibonacci[n]}} oder:
```

```
func=RSolveValue[{f[n]==f[n-1]+f[n-2], f[0]==0, f[1]==1}, f, n]
```

Hinweis: Die Folge ‚Sloane 136598‘ ist die negative Fortsetzung der Folge ‚Reed-Jameson‘.

Die Fibonacci-Folge und die Lucas-Folge hängen eng mit der Zahl $\Phi = 1.618 \dots$ des Goldenen Schnittes zusammen ($\Phi = \varphi$).

Die charakteristische Gleichung der Fibonacci-Folge und die expliziten Formeln für die Fibonacci- und die Lucas-Sequenz lauten:

$$x^2 - x - 1 = 0 \text{ mit den Lösungen } \varphi = \frac{1 + \sqrt{5}}{2} \text{ und } \psi = \frac{1 - \sqrt{5}}{2}$$
$$F_n = \frac{\varphi^n - \psi^n}{\varphi - \psi} = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right]$$
$$L_n = \left(\frac{1 + \sqrt{5}}{2} \right)^n + \left(\frac{1 - \sqrt{5}}{2} \right)^n \quad (34)$$

Zwischen den Lucas-Zahlen und den Fibonacci-Zahlen gibt es viele Zusammenhänge, auf die hier nicht eingegangen werden soll (z.B. $L_{2n} + 2(-1)^{n-1} = 5F_n^2$, oder $L_n = F_{n-1} + F_{n+1}$).

Bei den Reed Jameson-Folgen gibt es ähnlich interessante Zusammenhänge mit Primzahlen wie mit der Perrin-Folge (siehe Anhang).

Weitere Eigenschaften der Folgen aus Tabelle 10 :

4.10.1.1 MATRIX-DARSTELLUNGEN:

$$\text{Fibonacci } F_n: Q = \begin{bmatrix} F_2 & F_1 \\ F_1 & F_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad Q^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix}$$

$$\text{Lucas } L_n: Q = \begin{bmatrix} L_2 & L_1 \\ L_1 & L_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad Q^n = \begin{bmatrix} L_{n+1} & L_n \\ L_n & L_{n-1} \end{bmatrix}$$

$$\text{Padovan und Perrin } P_n: Q = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad Q^n = \begin{bmatrix} P_{n-5} & P_{n-3} & P_{n-4} \\ P_{n-4} & P_{n-2} & P_{n-3} \\ P_{n-3} & P_{n-1} & P_{n-2} \end{bmatrix}$$

Reed Jameson (,5# + Sloane A133394, sowie A136598) RSP_n, RSM_n :

$$Q = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}; RSP_n = Q^n \cdot \begin{pmatrix} 2 \\ 0 \\ 2 \\ 0 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}^n \cdot \begin{pmatrix} 2 \\ 0 \\ 2 \\ 0 \\ 5 \end{pmatrix}$$

$$RSM_n = Q^n \cdot \begin{pmatrix} 0 \\ -3 \\ 0 \\ 0 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}^n \cdot \begin{pmatrix} 0 \\ -3 \\ 0 \\ 0 \\ 5 \end{pmatrix}$$

Hinweis: RSP_n liefert die Folgenglieder in positiver Richtung, RSM_n in negativer Richtung.

4.10.1.2 MATHEMATICA-PROGRAMME ZUM ERZEUGEN VON LINEAR REKURSIVEN FOLGEN:

Fibonacci F_n :

```
LinearRecurrence[{1, 1}, {0, 1}, 30]
```

```
Table[Fibonacci[n], {n, 0, 30}]
```

```
{0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, ...}
```

Lucas L_n :

```
LinearRecurrence[{1, 1}, {2, 1}, 30]
```

```
Table[LucasL[n], {n, 0, 30}]
```

```
{2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778, ...}
```

Pell P_n :

```
LinearRecurrence[{2, 1}, {0, 1}, 30]
```

```
CoefficientList[Series[x/(1-2*x-x^2), {x, 0, 30}], x]
```

```
Expand[Table[((1+Sqrt[2])^n - (1-Sqrt[2])^n)/(2Sqrt[2]), {n, 0, 30}]]
```

```
a=1;b=0;c=0;lst={b};Do[c=a+b+c;AppendTo[lst,c];a=b;b=c,{n,30}];lst
```

```
{0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, 13860, 33461, 80782, 195025, ...}
```

Pell-Lucas Q_n :

```
LinearRecurrence[{2, 1}, {2, 2}, 30]
```

```
aa={};Do[k=Expand[((1+Sqrt[2])^n + (1-Sqrt[2])^n)];
```

```
AppendTo[aa,k],{n,0,30}]; aa
```

```
a=c=0; t={b=2}; Do[c=a+b+c; AppendTo[t,c]; a=b;b=c,{n,40}]; t
```

```
{2, 2, 6, 14, 34, 82, 198, 478, 1154, 2786, 6726, 16238, 39202, 94642, 228486, ...}
```

Fibonacci-Zahlen und andere rekursive Folgen

Padovan P_n :

```
LinearRecurrence[{0, 1, 1}, {1, 1, 1}, 30]
```

```
LinearRecurrence[{0, 1, 1}, {1, 0, 0}, 30]
```

```
a[0]=1; a[1]=a[2]=0; a[n_]:=a[n]=a[n-2]+a[n-3]; Table[a[n], {n, 0, 30}]
```

```
CoefficientList[Series[(1-x^2)/(1-x^2-x^3), {x, 0, 30}], x]
```

Weitere Mathematica-Programme: Bitte den Autor kontaktieren.

```
{1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, 265, 351, ...}  
bzw.  
{1, 0, 0, 1, 0, 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, ...}
```

Tribonacci T :

```
LinearRecurrence[{1, 1, 1}, {0, 1, 2}, 30]
```

```
{0, 1, 2, 3, 6, 11, 20, 37, 68, 125, 230, 423, 778, 1431, 2632, 4841, 8904, 16377, ...}
```

Quadranacci Q :

```
LinearRecurrence[{1, 1, 1, 1}, {0, 1, 2, 4}, 30]
```

```
{0, 1, 2, 4, 7, 14, 27, 52, 100, 193, 372, 717, 1382, 2664, 5135, 9898, 19079, 36776, ...}
```

Perrin P_n :

```
LinearRecurrence[{0, 1, 1}, {3, 0, 2}, 30]
```

```
CoefficientList[Series[(3-x^2)/(1-x^2-x^3), {x, 0, 30}], x]
```

```
explFunc=RSolve[{f[n]==f[n-2]+f[n-3], f[0]==3, f[1]==0, f[2]==2}, f[n], n]  
Round[Table[Evaluate[f[n]/.First[explFunc]], {n, 0, 30}]] (*fast*)
```

Weitere Mathematica-Programme: Bitte den Autor kontaktieren.

```
{3, 0, 2, 3, 2, 5, 5, 7, 10, 12, 17, 22, 29, 39, 51, 68, 90, 119, 158, 209, 277, ...}
```

,negative' Perrin P_n :

```
LinearRecurrence[{-1, 0, 1}, {3, -1, 1}, 30]
```

```
explFunc=RSolveValue[{f[n]==-f[n-1]+f[n-3],  
f[0]==3, f[1]==-1, f[2]==1}, f, n]  
Round[Table[Round[explFunc[n]], {n, 0, 30}]] (*fast*)
```

```
{3, -1, 1, 2, -3, 4, -2, -1, 5, -7, 6, -1, -6, 12, -13, 7, 5, -18, 25, -20, 2, 23, -43, 45, -  
22, -21, 66, -88, 67, -1, -87}
```

,5'+ Sloane 133394 (Reed Jameson) R_n :

```
LinearRecurrence[{0, 1, 0, 0, 1}, {5, 0, 2, 0, 2}, 30]
```

oder:

```

RecurrenceTable[{a[n]==a[n-2]
+a[n-5],a[1]==5,a[2]==0,a[3]==2,a[4]==0,a[5]==2},a,{n,1,30}]

reedJamesonMatrix={{0,1,0,0,1},{1,0,0,0,0},{0,1,0,0,0},{0,0,1,0,0},{0,
0,0,1,0}}; vect={{2},{0},{2},{0},{5}};
Flatten[Table[(MatrixPower[reedJamesonPlusMatrix,n].vect)[[1]],
{n,4-4,100-5}]]

```

Weitere Mathematica-Programme: Bitte den Autor kontaktieren.
{5,0,2,0,2,5,2,7,2,9,7,11,14,13,23,20,34,34,47,57,67,91,101,138,158,...}

Sloane A136598: R_n^* :

```
LinearRecurrence[{0,0,-1,0,1},{5,0,0,-3,0},30]
```

```

explFunc=RSolve[{f[n]==-f[n-3]+f[n-5],f[0]==5,f[1]==0,f[2]==0,f[3]==-
3,f[4]==0},f[n],n]//Simplify
Round[Table[Evaluate[f[n]/.First[explFunc]],{n,0,100}]]

```

{5,0,0,-3,0,5,3,0,-8,-3,5,11,3,-13,-14,2,24,17,-15,-38,-15,39,55,...}

Sloane A135435:

```
LinearRecurrence[{0,0,0,1,0,0,1},{7,0,0,0,4,0,0},30]
```

```

explFunc=RSolve[{f[n]==f[n-4]+f[n-7],f[0]==7,f[1]==f[2]==f[3]==0,
f[4]==4,f[5]==f[6]==0},f[n],n]//Simplify
Round[Table[Evaluate[f[n]/.First[explFunc]],{n,0,100}]]

```

{7,0,0,0,4,0,0,7,4,0,0,11,4,0,7,15,4,0,18,19,4,7,33,23,4,25,52,27,11,5
8,75,31,36,110,102,42,94,185,133,78...}

(unbekannt, negative A135435):

```
LinearRecurrence[{0,0,-1,0,0,0,1},{7,0,0,-3,0,0,3},30]
```

{7,0,0,-3,0,0,3,7,0,-3,-10,0,3,13,7,-3,-16,-17,3,19,30,4,-22,-46,-
21,25,65,51,-21,-87,-97,0,...}

4.10.1.3 VERGLEICH DER VERSCHIEDENEN BERECHNUNGSMETHODEN

Wie man sieht gibt es eine ganze Reihe von Berechnungsmethoden, die sich sehr hinsichtlich Speicherbedarf und Geschwindigkeiten unterscheiden.

- 1) **LinearRecurrence[...]**: Beste Methode für Situation 1).
(Alternativ: RecurrenceTable[{a[n]==a[n-2]+...}]; diese Funktion ist flexibler und hat mehr Optionen)
- 2) Matrix-Methode: $M^n * \text{initVec}$. Unschlagbar schnell für Situation 2). Evtl. auch für Situation 1) geeignet.

Fibonacci-Zahlen und andere rekursive Folgen

- 3) Berechnung mit Hilfe einer expliziten Funktion (`RSolveValue[]`): Hängt davon ab, wie kompliziert die explizite Lösung der rekursiven Gleichung ist. Für Situation 1) ungeeignet.
- 4) `CoefficientList[...]`
- 5) Berechnung mit Hilfe der Nullstellen des charakteristischen Polynoms: `Solve[...]`
- 6) Berechnung nach Definition (z.B. `Do[d=a+b+c;AppendTo[lst,d];a=b;b=c,...`): Langsam, aber noch einigermaßen praktikabel!
- 7) Rekursive Methode: (z. B: `a[0]=1; a[1]=a[2]=0; a[n_]:=a[n]=a[n-2]+a[n-3];`): Ungeeignet!

Wir unterscheiden folgende Situationen:

- 1) Berechnung der rekursiven Folge von Anfang an bis zu einer Grenze N .
- 2) Berechnung einzelner Folgenglieder ohne die Vorgänger zu kennen (in der Regel für sehr große Indices). Hier kommen vor allem die Matrix-Methode oder die Berechnung mit Hilfe expliziter Formeln zur Anwendung.

Hier ein Beispiel für den 2. Fall: Die Folge ‚5+Reed Jameson‘:

```
f[n]==f[n-2]+f[n-5], f[0]==5, f[1]==0, f[2]==2, f[3]==0, f[4]==2
```

$N=10000000$. Die Rechenzeiten für das N .te Folgenglied:

Methode 1: 476 Sekunden

Methode 2: 0.0156 Sekunden

Methode 3: 2964 Sekunden

(Die explizite Lösung ist kompliziert. Sie verlangt die Berechnung der Nullstellen von Polynomen 5.ten Grades. Die Lösung der rekursiven Folge, vereinfacht mit ‚Simplify‘ ist immer noch ca. 15 Mathematica Notebook-Seiten lang...)

Von Peter Danzeglocke stammt eine Methode, die statt der Funktion `MatrixPower[]` eine Funktion `MatrixPowerMod[]` benutzt und die nur mit Werten modulo n arbeitet. Diese Methode kann bei allen erwähnten rekursiven Folgen angewendet werden und liefert die Folgenwerte modulo n in unschlagbar kurzen Rechenzeiten.

4.10.1.4 ZUSAMMENHANG MIT PRIMZAHLEN

Aus manchen Folgengliedern dieser linear rekursiv definierten Folgen ergeben sich interessante Zusammenhänge zu Primzahlen. Lange Zeit glaubte man, die Glieder der Perrin-Folge würden für Primzahl-Indices stets durch diese Primzahl teilbar sein:

$$P_n \equiv 0 \pmod{n}, \text{ wenn } n \in \mathbb{P}$$

Das wäre eine Methode, um Primzahlen mit einer einzigen Modulus-Operation zu bestimmen. Die Berechnung von P_n verlangt nur Additionen, bzw. zusätzliche Potenzierungen und Multiplikationen, wenn eine Explizite Formel für das n .te Glied der rekursiv definierten Folge bekannt ist.

Perrin's Methode zur Primzahl-Bestimmung ist auf den ersten Blick perfekt. Aber nur auf den ersten Blick: Nämlich nur bis Indices < 271441 . Dieser Index ergibt beim Modulus-Test den Wert 0, sollte jedoch einen Wert > 0 ergeben. Dieser Index bezeichnet die erste Perrin-Pseudo-Primzahl. Inzwischen sind Hunderte von ihnen bekannt (siehe Kapitel 20.3). Es ist jedoch beeindruckend, dass diese Methode zur Primzahl-Bestimmung bis zur Zahl 271441 perfekt funktioniert! Hier zeigt sich wieder einmal, dass numerische Evidenz nicht immer auf eine Gesetzmäßigkeit schließen lässt!

Grundsätzlich ist die rekursive Berechnung der Folgenglieder wesentlich schneller für kleine Indices (bei vielen Folgen im Bereich bis 10^6 ca. 100 mal schneller als die explizite Berechnung. Für sehr große Indices ändert sich dies vermutlich jedoch zugunsten der expliziten Berechnung...

Von Reed Jameson stammt eine ähnliche Methode zur Primzahl-Bestimmung, die ebenfalls die Modulus-Werte von Folgengliedern mit Primzahl-Indices auswertet. Hier werden jedoch zwei Folgen benutzt. Es sind dies die Folgen F_n („**Sloane 133394**“) und deren ‚komplementäre‘ Folge F_n^* („**Sloane A136598**“). (Hinweis: Die ‚komplementäre Folge ergibt sich, wenn die ‚normale‘ Folge in die entgegengesetzte Richtung der negativen Indices verlängert wird). Es wird hierbei die Summenfolge $S_n = (F_n \bmod n) + (F_n^* \bmod n)$ gebildet. Es ergibt sich, dass $S_n = 0$ wenn (und nur dann wenn) n eine Primzahl ist (wir starten mit Index 0). [Diese Vermutung wurde verifiziert im Bereich bis \$n = 10^{10}\$. Peter Danzeglocke entdeckte jedoch zahlreiche Reed Jameson Pseudo-Primzahlen im Bereich \$n > 10^{15}\$.](#)²⁷

Noch unbekannt ist eine Methode, die ähnlich wie die von Reed Jameson funktioniert. Hierbei haben wir die zwei (zueinander komplementären) Folgen

F_n : ‚**Sloane 135435**‘ und deren Komplementärfolge F_n^* .

Wieder bilden wir die Summenfolge $S_n = (F_n \bmod n) + (F_n^* \bmod n)$. Es ergibt sich, dass $S_n = 0$ wenn n eine Primzahl ist. Diese Vermutung wurde von Peter Danzeglocke numerisch bis $n = 10^9$ überprüft. Bisher sind keine Pseudo-primzahlen zu dieser Folge bekannt (Stand: Dezember 2020). Allerdings gibt es etwa 10 Prozent ‚triviale‘ Pseudo-Primzahlen, die jedoch leicht separiert werden können, da für alle zu dieser Folge gehörenden Pseudo-Primzahlen gilt:

$$F_n \equiv 0 \pmod{2} \parallel F_n \equiv 0 \pmod{5}.$$

Hier ein paar Graphen der interessanten Summenfolgen der Modulus-Werte der obigen Folgen:

²⁷ Danzeglocke testete die Reed Jameson Vermutung für alle Fermat-Pseudoprimzahlen zur Basis 2 im Bereich bis $n < 2^{64}$. Siehe Anhang „Reed Jameson Pseudoprimzahlen“.

Fibonacci-Zahlen und andere rekursive Folgen

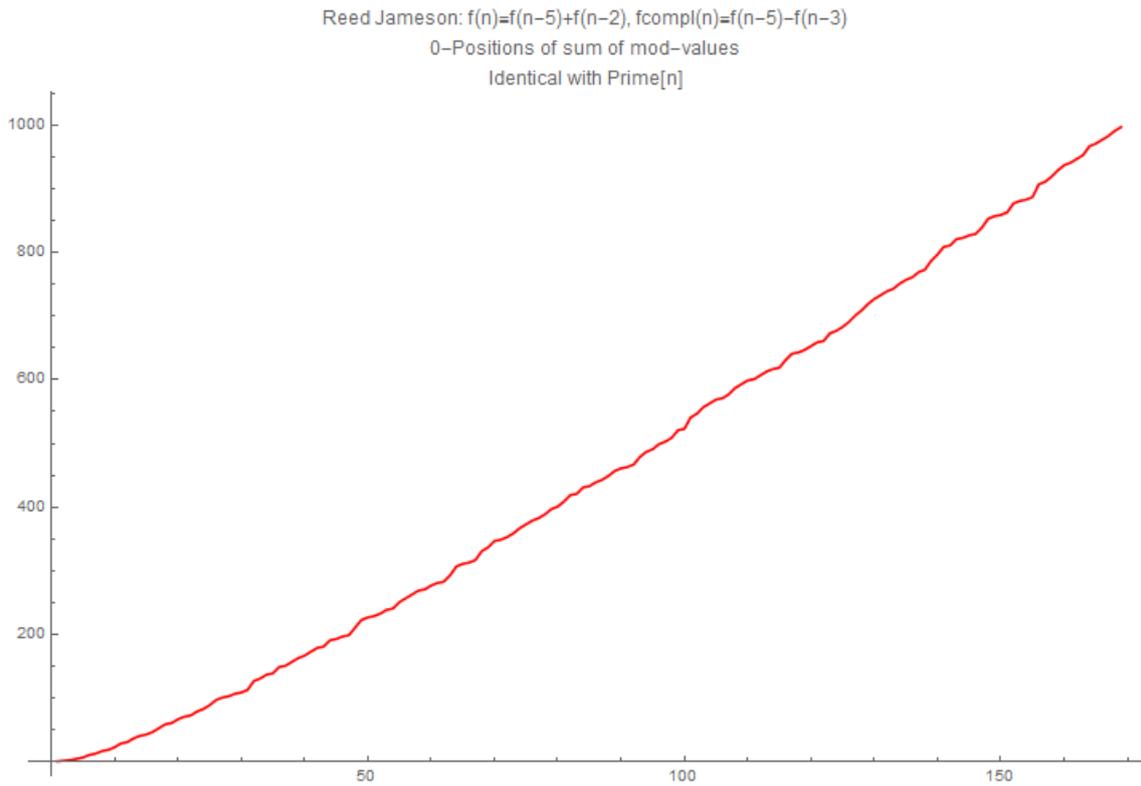


Abbildung 11: Reed Jameson Folge: Plot der 0-Positionen der Summe der Mod-Werte

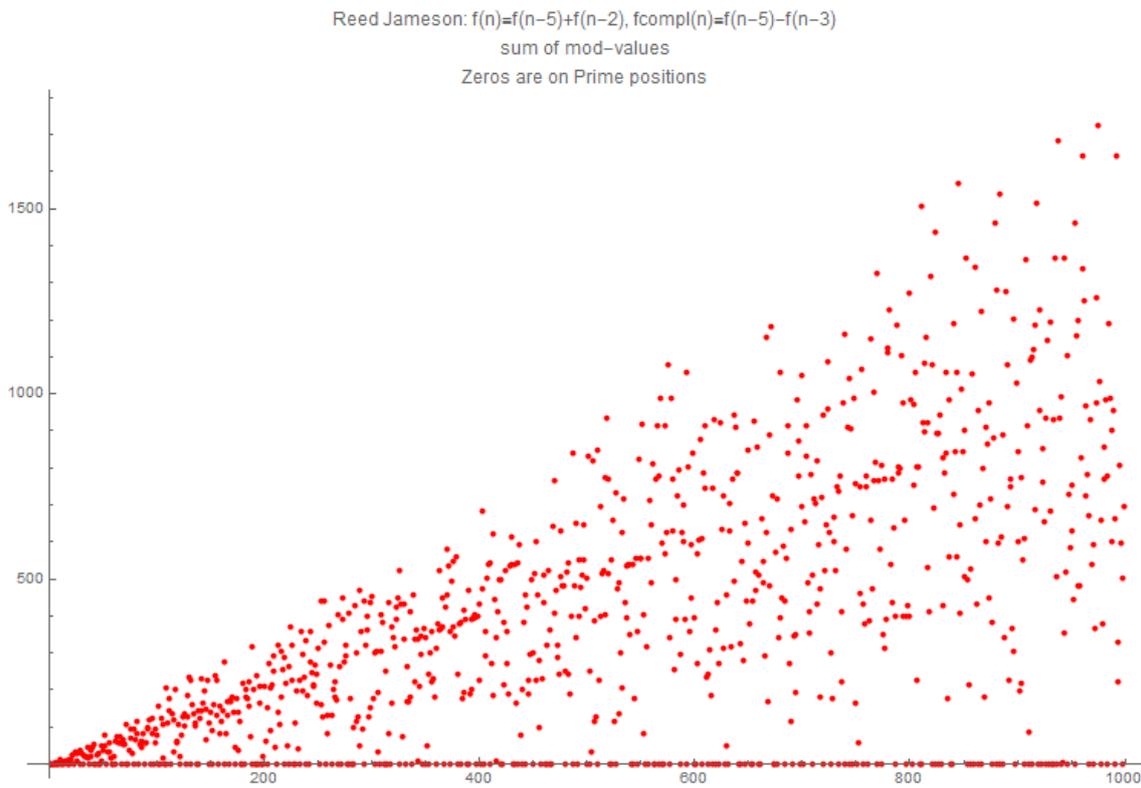


Abbildung 12: Reed Jameson Folge: Plot der Summe der Mod-Werte

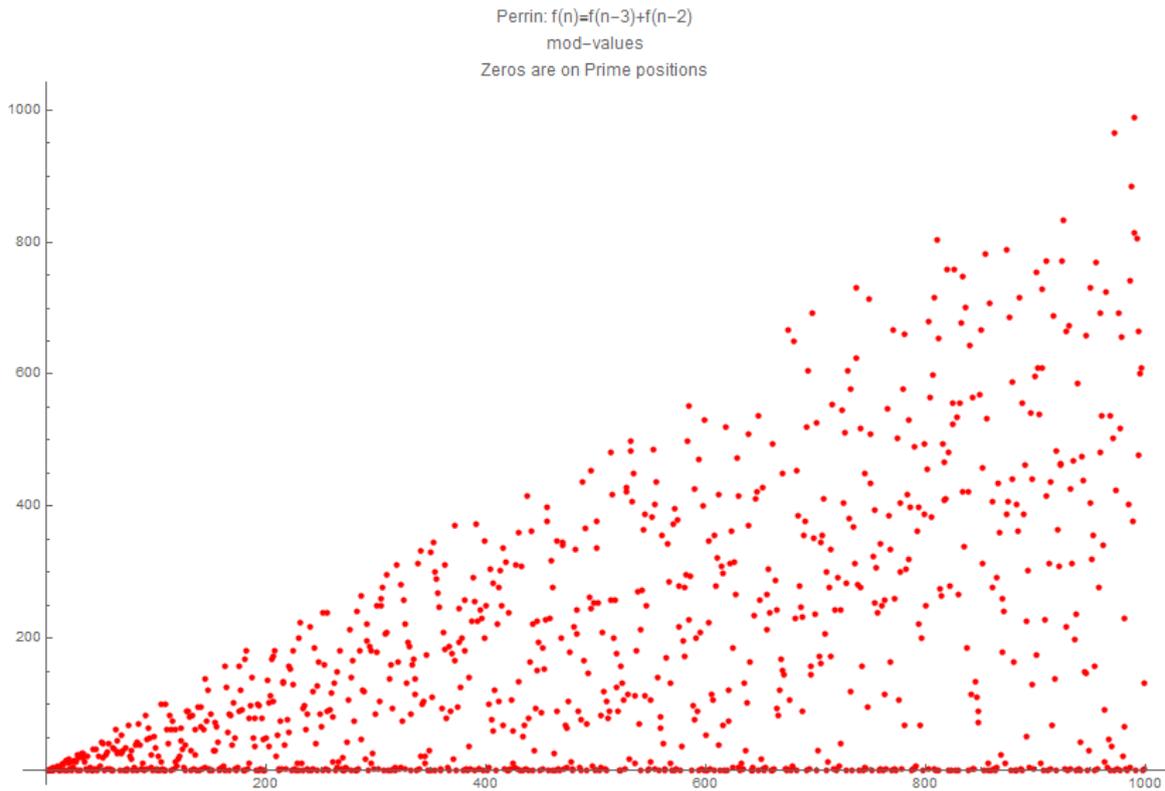


Abbildung 13: Perrin-Folge: Plot der Mod-Werte. Die Nullstellen sind (fast immer) bei Primzahlen

Hinweis: Die Mathematica-Programme zum Erzeugen der Plots finden sich im Anhang (Reed Jameson und Perrin-Folgen).

4.10.2 FIBONACCI-PRIMZAHLEN UND FIBONACCI PSEUDO-PRIMZAHLEN

Eine Fibonacci-Primzahl ist eine Primzahl, die gleichzeitig in der Fibonacci-Folge vorkommt.

Schauen wir uns die Fibonacci-Folge genauer an und markieren die Werte mit Primzahl-Indices:

0, 1, **1**, **2**, 3, **5**, 8, **13**, 21, 34, 55, **89**, 144, **233**, 377, 610, 987, **1597**, 2584, **4181**, 6765, 10946, 17711, **28657**, 46368, 75025, 121393, 196418, 317811, **514229**, 832040, **1346269**

So stellen wir fest, dass für viele Primzahl-Indices auch die dazugehörigen Fibonacci-Zahlen F_n Primzahlen (Treffer in rot, Gegenbeispiele in lila) sind. Es gilt zwar der Satz:

Wenn F_n eine Primzahl ($n \neq 2$ und $n \neq 4$) ist, dann ist auch n eine Primzahl, nicht jedoch seine Umkehrung. Fibonacci-Zahlen, die zu Primzahl-Indices gehören, selbst jedoch keine Primzahlen sind, sind eine Teilmenge der Fibonacci-Pseudo-Primzahlen. Fibonacci-Pseudo-Primzahlen sind definiert als zusammengesetzte Zahlen, für die die Kongruenz $V_n \equiv 1 \pmod{n}$ gilt.

Fibonacci-Zahlen und andere rekursive Folgen

Es gilt weiterhin der Satz:

Wenn n eine Primzahl ist, dann gilt:

$$\begin{array}{ll} F_n \equiv 0 \pmod{n} & \text{wenn } n \equiv 0 \pmod{5} \\ F_{n-1} \equiv 0 \pmod{n} & \text{wenn } n \equiv \pm 1 \pmod{5} \\ F_{n+1} \equiv 0 \pmod{n} & \text{wenn } n \equiv \pm 2 \pmod{5} \end{array}$$

Die erste Bedingung trifft nur auf F_5 zu: $F_5 \pmod{5} \equiv 5 \pmod{5} \equiv 0$

Ein Beispiel für die zweite Bedingung: $n = 11$; $n \equiv 1 \pmod{5} \Rightarrow F_{10} \equiv 0 \pmod{11}$

Ein Beispiel für die dritte Bedingung: $n = 13$; $n \equiv -2 \pmod{5} \Rightarrow F_{14} \equiv 0 \pmod{13}$

Die größte, derzeit bekannt Fibonacci-Primzahl lautet F_{104911} . Sie besitzt 21925 Dezimalzahlen. Es ist noch unbekannt (Stand Dez. 2015), ob es unendlich viele Fibonacci-Primzahlen gibt.

Berechnung mit Mathematica:
`Select[Fibonacci[Range[400]], PrimeQ]`

4.10.3 META-FIBONACCI-FOLGEN

In Meta-Fibonacci-Folgen wird das neue Folgenglied nicht direkt aus den Werten der beiden vorangegangenen Folgenglieder berechnet, sondern indirekt über Offsets oder Indices. Die erste, in der Literatur vorkommende Meta-Fibonacci Folge ist Hofstadters berühmte Q-Folge²⁸:

$$Q(n) = Q(n - Q(n - 1)) + Q(n - Q(n - 2)), n > 2$$

mit $Q(1) = Q(2) = 1$ (35)

Die Folge scheint auf den ersten Blick ziemlich chaotisch, zeigt jedoch generationenartige Strukturen. Die Folge ist noch relativ unerforscht. Die ersten Folgenglieder lauten:

1 1, 2, 3, 3, 4, 5, 5, 6, 6, 6, 8, 8, 8, 10, 9, 10, 11, 11, 12, 12,
12, 12, 16, 14, 14, 16, 16, 16, 16, 20, 17, 17, 20

Statt wie bei der Fibonacci-Folge die beiden vorangegangenen Werte zu addieren, sagen uns die beiden vorangegangenen Werte, wie weit wir in der Folge rückwärtsgehen müssen um, die Zahlen zu erhalten, die wir addieren müssen.

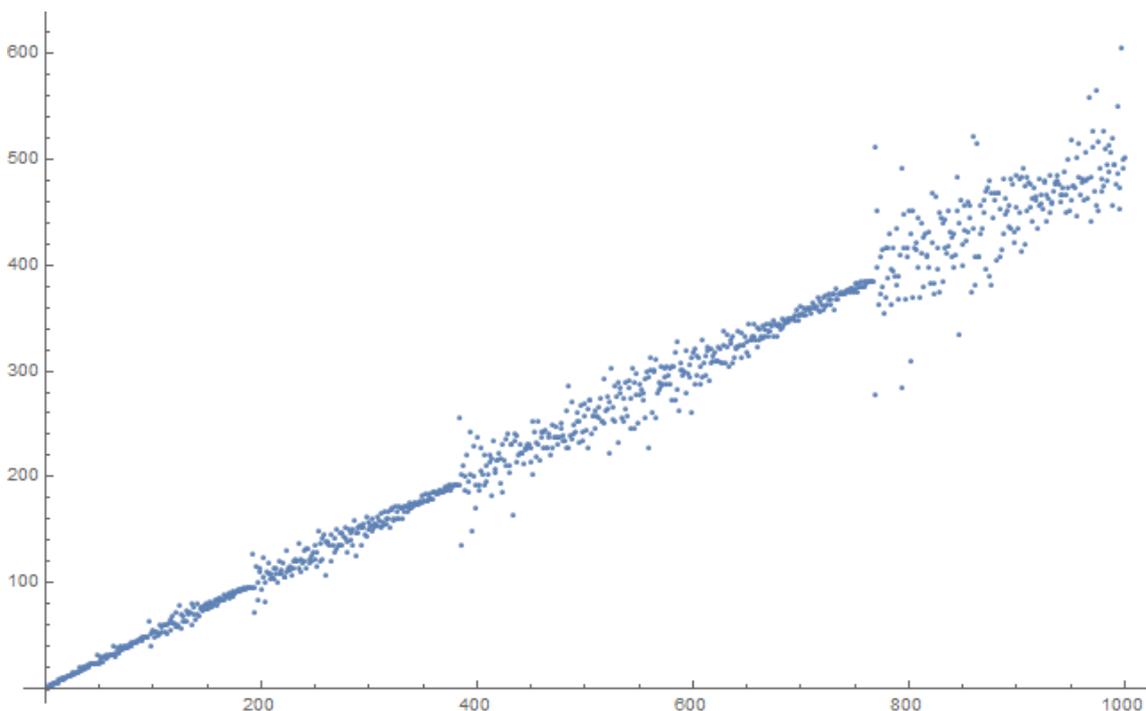


Abbildung 14: Hofstadters Q-Folge: Eine Meta-Fibonacci-Folge

Mathematica:

```
a[1] = a[2] = 1; a[n_] := a[n] = a[n - a[n - 1]] + a[n - a[n - 2]];
ListPlot[Table[{n, a[n]}, {n, 1, 1000}], PlotRange->Full]
```

²⁸ Hofstadter: Gödel, Escher, Bach S. 149

4.11 CARMICHAEL- UND KNÖDEL-ZAHLEN

Eine (zusammengesetzte) Zahl n heisst Carmichael-Zahl, wenn gilt:

$a^{n-1} \equiv 1 \pmod{n}$, und zwar für alle zu n teilerfremden $a < n$. Für die Teiler von n gilt die Kongruenz nicht. Die kleinste Carmichael-Zahl ist 561. Die Primfaktorenzerlegung von 561 lautet $561 = 3 \cdot 11 \cdot 17$

Es gibt unendlich viele Carmichael-Zahlen. Als Beispiel hier alle Carmichael-Zahlen bis 10000:

Carmichael-Zahl	Primfaktoren
561	3·11·17
1105	5·13·17
1729	7·13·19
2465	5·17·29
2821	7·13·31
6601	7·23·41
8911	7·19·67
10585	5·29·73
15841	7·31·73
29341	13·37·61
41041	7·11·13·41
46657	13·37·97
52633	7·73·103
62745	3·5·47·89
63973	7·13·19·37
75361	11·13·17·31

Abbildung 15: Carmichael-Zahlen bis 10000

Die größte bisher bekannte Carmichael-Zahl (Stand: Dez. 2015) kann hier nicht abgedruckt werden, denn sie besitzt mehr als 10 Milliarden Primfaktoren und ca. 300 Milliarden Dezimalstellen (d.h. es existiert lediglich ein Konstruktionsprinzip).²⁹ Es lässt sich einfach beweisen, dass jede Carmichael-Zahl mindestens drei verschiedene Primfaktoren enthalten muss und quadratfrei ist. Es gibt Konstruktionsmethoden, mit denen sich sehr große Carmichael-Zahlen konstruieren lassen. Umgekehrt ist es sehr schwierig und aufwendig, sehr große Zahlen auf ihre Carmichael-Eigenschaft hin zu testen, da man sie zu diesem Zweck faktorisieren muss.

Sei $C(n)$ die Anzahl von Carmichael-Zahlen bis zu einem gegebenen n . Dann existieren folgende Abschätzungen:

$$n^{\frac{1}{3}} < C(n) < ne^{\left(-\frac{\ln n \ln \ln \ln n}{\ln \ln n}\right)}$$

²⁹ <http://math.ucsd.edu/~kedlaya/ants10/poster-hayman.pdf>

Eine Verallgemeinerung der Carmichael-Zahlen führt zu den Knödel-Zahlen:

Mit K_n bezeichnet man die Menge der zusammengesetzten Zahlen $a^{m-n} \equiv 1 \pmod{m}$, und zwar für alle zu m teilerfremden $a < m$. Als Spezialfall mit $n = 1$ ergeben sich die Carmichael-Zahlen. Jede zusammengesetzte Zahl m ist eine Knödel-Zahl K_n mit $n=m-\varphi(n)$. Die ersten Knödel-Mengen K_n lauten:³⁰

n	K_n
1	561, 1105, 1729, 2465, 2821, 6601, ...
2	4, 6, 8, 10, 12, 14, 22, 24, 26, ...
3	9, 15, 21, 33, 39, 51, 57, 63, 69, ...
4	6, 8, 12, 16, 20, 24, 28, 40, 44, ...

Mathematica:

```
Cases[Range[1, 100000, 2], n_ /; Mod[n, CarmichaelLambda[n]] == 1 && !
PrimeQ[n]]
```

4.12 MIRP-PRIMZAHLEN

Eine Mirpzahl ist eine Primzahl, die eine andere Primzahl ergibt, wenn sie rückwärts gelesen wird.

Die größte bisher bekannte Mirpzahl ist (Stand Oktober 2015):

$$10^{10006} + 941992101 \cdot 10^{4999} + 1$$

Mathematica:

```
fQ[n_] := Block[{idn = FromDigits@ Reverse@ IntegerDigits@ n}, PrimeQ@
idn && n != idn]; Select[Prime@ Range@ 200, fQ]
```

Kurioses (die folgende Liste enthält 11 aufeinander folgende Primzahlen, die sämtlich MIRP-Primzahlen sind):

1477271183, 1477271249, 1477271251, 1477271269, 1477271291, 1477271311, 1477271317,
1477271351, 1477271357, 1477271381, 1477271387

4.13 WAGSTAFF-PRIMZAHLEN

Wagstaff Primzahlen sind Primzahlen der Form

$$p = \frac{2^q + 1}{3}, \text{ mit } q \text{ prim, ungerade} \quad (36)$$

³⁰<https://de.wikipedia.org/wiki/Knödel-Zahl>

Wagstaff-Primzahlen

Es sind z. Zt. (Stand Oktober 2015) 43 Wagstaff-Primzahlen p bekannt (in Rot: PRP-Primzahlen)³¹:

Tabelle 11: Wagstaff-Primzahlen: Exponent q

n	q	N	q
1	3	22	2617
2	5	23	3539
3	7	24	5807
4	11	25	10501
5	13	26	10691
6	17	27	11279
7	19	28	12391
8	23	29	14479
9	31	30	42737
10	43	31	83339
11	61	32	95369
12	79	33	117239
13	101	34	127031
14	127	35	138937
15	167	36	141079
16	191	37	267017
17	199	38	269987
18	313	39	374321
19	347	40	986191
20	701	41	4031399
21	1709	42	13347311
		43	13372531

Wagstaff-Primzahlen kann mit folgendem Mathematica-Programm berechnet werden:

```
Select[Array[(2^#+1)/3&,190],PrimeQ]
Output:
{3,11,43,683,2731,43691,174763,2796203,715827883,2932031007403,
768614336404564651,201487636602438195784363,
845100400152152934331135470251,
56713727820156410577229101238628035243,
62357403192785191176690552862561408838653121833643}
```

³¹ Pseudo Primzahltests liefern PRPs (Pseudoprimzahlen) und arbeiten mit probabilistischen Methoden, ergeben jedoch nach menschlichem Ermessen sichere Aussagen über Primalität

Und hier ein Mathematica-Programm für die Prim-Exponenten der Wagstaff-Primzahlen:
`a = {}; Do[c = 1 + Sum[2^(2n - 1), {n, 1, x}]; If[PrimeQ[c], AppendTo[a, c]], {x, 0, 100}]; a`

4.14 WIEFERICH-PRIMZAHLEN

Eine Primzahl, die die Kongruenz $2^{p-1} \equiv 1 \pmod{p^2}$ erfüllt, wird in der Literatur als ‚Wieferich-Primzahl‘ bezeichnet, da Wieferich Anfang des letzten Jahrhunderts der erste war, der sich mit diesen Zahlen beschäftigte. Zur Erinnerung sei hier noch einmal die Kongruenz-Beziehung des kleinen Fermatschen Satzes erwähnt: $2^{p-1} \equiv 1 \pmod{p}$.

Diese gilt für jede ungerade Primzahl. Im Gegensatz dazu gilt die obige Wieferich-Kongruenz-Beziehung nur für sehr wenige Primzahlen (genauer genommen nur für zwei Zahlen: **1093** und **3511**). Dies sind die einzigen Wieferich-Primzahlen, die bis heute bekannt sind (Stand: Dez. 2015). Es ist auch bekannt, dass bis $4.968543 \cdot 10^{17}$ keine weiteren Wieferich-Primzahlen existieren (Stand Nov. 2016).

Bemerkenswert sind die Muster in der Binärdarstellung des Wertes von Wieferich-Primzahlen (genauer: um 1 verringerter Wert):

$$1092 = 10001000100_2, \quad 3510 = 110110110_2$$

Es gibt einen engen Zusammenhang zu den sogenannten ‚mächtigen‘ Zahlen P_i („powerful numbers“), für die gilt: $p | P_i$ und $p^2 | P_i$.

Die ersten ‚mächtigen‘ Zahlen lauten: 1,4,8,9,16,25,27,32,36,49 ... Sie sind alle der Form $a^2 b^3$ mit $a, b \geq 1$. Die Summe der Kehrwerte aller mächtigen Zahlen ist endlich und hat den Wert:

$$\sum_i \frac{1}{P_i} = \frac{\zeta(2)\zeta(3)}{\zeta(6)} = 1.9435964 \quad (37)$$

Primzahlen, die die Kongruenzbedingung $2^p \not\equiv 1 \pmod{p^2}$ nicht erfüllen, werden allgemein als ‚Nicht-Wieferich‘-Primzahlen bezeichnet. Beide sind zueinander komplementäre Mengen, d.h. wenn eine von ihnen endlich ist, dann muss die andere unendlich groß sein. Es gibt interessante Querverbindungen zur abc-Vermutung (Kapitel 11.1). Die Literatur zu Wieferich-Primzahlen ist sehr groß. Es gibt viele weitere überraschende interessante Zusammenhänge zu Mersenne- und Fermat-Primzahlen, als auch zu anderen Gebieten der Zahlentheorie.³²

Wieferich-Primzahlen können zu einer beliebigen positiven ganzzahligen Basis a verallgemeinert werden, wenn sie folgende Kongruenz erfüllen:

$$a^{p-1} \equiv 1 \pmod{p^2} \quad (38)$$

³² https://en.wikipedia.org/wiki/Wieferich_prime

Wieferich-Primzahlen

Es gilt die Vermutung, dass es zu jeder natürlichen Basis a unendlich viele ‚Basis- a -Wieferich-Primzahlen‘ gibt. So sind z.B. für $a = 5$ folgende Basis-5-Wieferich-Primzahlen bekannt (Stand: Oktober 2016):

2, 20771, 40487, 53471161, 1645333507, 6692367337, 188748146801

Der Quotient $q_p(a) = \frac{a^{p-1}-1}{p}$ wird allgemein als Fermat-Quotient von p zur Basis a bezeichnet. Der Modulo-Rest des Fermat-Quotienten $q_p(a)$ hat logarithmische Eigenschaften. Wenn p kein Teiler von ab ist, gilt:

$$q_p(ab) \equiv q_p(a) + q_p(b) \pmod{p} \quad (39)$$

Weitere Eigenschaften:

$$q_p(p-1) \equiv 1 \text{ und } q_p(p+1) \equiv -1 \pmod{p} \quad (40)$$

$$q_p(2) = \frac{1}{p} \left(1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots - \frac{1}{p-1} \right) \pmod{p} \quad (41)$$

Mathematica-Programme zum Berechnen von Wieferich-Primzahlen:

```
Select[Prime[Range[50000]], Divisible[2^(#-1)-1, #^2]&]  
Select[Prime[Range[50000]], PowerMod[2, #-1, #^2]==1&]
```

Es existieren weitere interessante Querverbindungen zu anderen zahlentheoretischen Gebieten:

Zu Fermats Satz:

Sei $k^p + l^p + m^p = 0$ (mit $k, l, m \in \mathbb{Z}$ und $p \in \mathbb{P}$). Weiter soll p kein Teiler des Produktes klm sein. Dann ist p eine Wieferich-Primzahl.

Zu Mersenne-Primzahlen:

Sei M_q eine Mersennesche Zahl mit primem Index q und p eine Primzahl, die ein Teiler von M_q ist. Wenn auch p^2 ein Teiler von M_q ist, dann und nur dann ist p eine Wieferich-Primzahl.

Doppelte Wieferich-Primzahlen:

Ein Paar von verallgemeinerten Wieferich-Primzahlen, für die gilt:

$$q^{p-1} \equiv 1 \pmod{p^2} \text{ und } p^{q-1} \equiv 1 \pmod{q^2} \quad (42)$$

wird als ‚Doppelte Wieferich-Primzahl‘ bezeichnet. Hier ein Beispiel:

(83,4871) sind doppelte Wieferich-Primzahlen.

Hinweis: Mit Hilfe der doppelten Wieferich-Primzahlen ist die Catalansche Vermutung bewiesen worden (siehe 20.1). Über Wieferich Primzahlen gibt es viele interessante Veröffentlichungen, die in den bekannten Internetseiten nachgelesen werden können.

Hier eine Tabelle einiger bekannter ‚Basis- a -Wieferich-Primzahlen‘ (Stand: Okt. 2016):

Tabelle 12: Verallgemeinerte Wieferich-Primzahlen

Basis	Wieferich-Primzahlen zur Basis
1	2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ... (Alle Primzahlen)
2	1093, 3511
3	11, 1006003
4	1093, 3511
5	2, 20771, 40487, 53471161, 1645333507, 6692367337, 188748146801
6	66161, 534851, 3152573
7	5, 491531
8	3, 1093, 3511
9	2, 11, 1006003
10	3, 487, 56598313
11	71
12	2693, 123653
13	2, 863, 1747591
14	29, 353, 7596952219
15	29131, 119327070011
16	1093, 3511
17	2, 3, 46021, 48947, 478225523351
18	5, 7, 37, 331, 33923, 1284043
19	3, 7, 13, 43, 137, 63061489
20	281, 46457, 9377747, 122959073
21	2
22	13, 673, 1595813, 492366587, 9809862296159
23	13, 2481757, 13703077, 15546404183, 2549536629329
24	5, 25633
25	2, 20771, 40487, 53471161, 1645333507, 6692367337, 188748146801
26	3, 5, 71, 486999673, 6695256707
27	11, 1006003,
28	3, 19, 23
29	2
30	7, 160541, 947270757
31	7, 79, 6451, 2806861
37	2, 3, 77867, 76407520781
41	2, 29, 1025273, 138200401
43	5, 103, 13368932516573
47	???
53	2, 3, 47, 59, 97
59	2777, 18088417183289
61	2
67	7, 47, 268573
71	3, 47, 331
73	2, 3
79	7, 263, 3037, 1012573, 60312841, 8206949094581
83	4871, 13691, 315746063
89	2, 3, 13
97	2, 7, 2914393, 76704103313

4.15 WILSON-PRIMZAHLEN

Eine Primzahl, die die Kongruenz $(p - 1)! \equiv -1 \pmod{p^2}$ erfüllt, wird in der Literatur als ‚Wilson-Primzahl‘ bezeichnet.

Zur Zeit (Nov. 2016) sind nur drei Wilson-Primzahlen bekannt. Diese lauten:

5, 13 und 563

Falls weitere Wilson-Primzahlen existieren, so müssen diese größer als $2 \cdot 10^{13}$ sein. (Stand: Okt. 2016). Es wird vermutet, dass es unendlich viele Wilson-Primzahlen gibt.

Der Quotient $W(p)$ heißt ‚Wilson-Quotient‘:

$$W(p) = \frac{(p - 1)! + 1}{p} \quad (43)$$

Wilsons Satz besagt, dass alle Primzahlen p die Kongruenz

$$(p - 1)! \equiv -1 \pmod{p}$$

erfüllen.

Hier ein Graph der Wilson-Quotienten der ersten 100 Primzahlen:

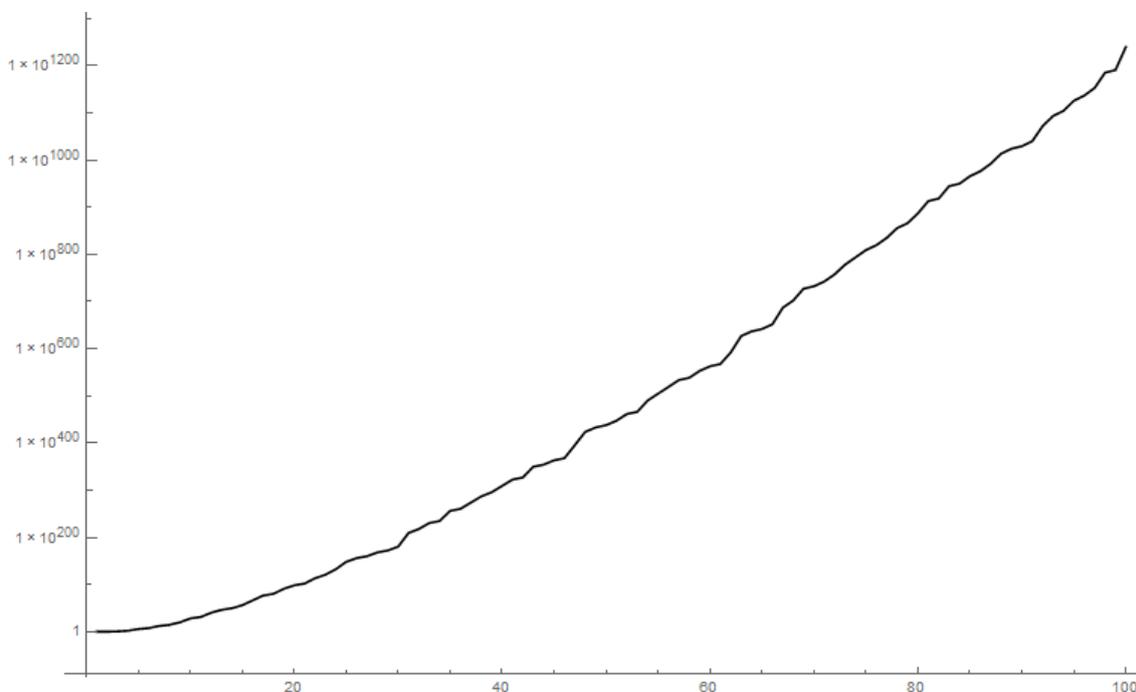


Abbildung 16: Wilson-Quotienten der ersten 100 Primzahlen(logarithmische Darstellung)

Mathematica:

```
wilsonQuotients=Table[ ((Prime[i]-1)!+1)/(Prime[i]), {i, 1, 100}];  
ListLogPlot[wilsonQuotients, Joined->True, PlotStyle->Black]
```

Auch bei den Wilson-Primzahlen gibt es Verallgemeinerungen, die in der Literatur beschrieben sind.

4.16 WOLSTENHOLME-PRIMZAHLEN

Eine Primzahl, die folgender Kongruenz genügt, heißt ‚Wolstenholme-Primzahl‘:

$$\binom{2p-1}{p-1} \equiv 1 \pmod{p^4} \quad (44)$$

Man beachte, dass nach dem Wolstenholme-Theorem für jede Primzahl $p > 3$ gilt:

$$\binom{2p-1}{p-1} \equiv 1 \pmod{p^3} \quad (45)$$

Die einzigen derzeit bekannten Wolstenholme-Primzahlen lauten (Stand: Okt. 2016):

16843 und **2124679**

Wenn es weitere Wolstenholme-Primzahlen gibt, so sind diese größer als 10^9 . Es wird vermutet, dass es unendlich viele Wolstenholme-Primzahlen gibt.

- 3) Jede nichtperiodische Folge besitzt ein ‚Zentrum‘, das in der Gegend des Minimums liegt. Es ist die jeweils einzige ungerade zusammengesetzte Zahl in dieser Folge. Daher können die Folgen nach dieser einzigen zentralen ungeraden zusammengesetzten Zahl benannt werden. Es sind dies die rot markierten Zahlenwerte.
- 4) In Richtung positiver Indices besteht eine starke Tendenz, den Grad der ‚Zusammengesetztheit‘ zu erhöhen, in Richtung negativer Indices existiert die Tendenz zu erhöhter ‚Primalität‘. Dies drückt sich darin aus, dass beim ‚Zentrums-Element‘ (das zusammengesetzt und ungerade ist) das ‚Umkippen‘ von ‚Prim‘ in ‚Zusammengesetzt‘ (oder umgekehrt, je nach dem aus welcher Richtung man kommt) erfolgt.
- 5) Folgende RG-Folgen sind periodisch (klassifiziert nach dem zentralen, ungeraden, zusammengesetzten Element):
3,5,6,7,10,12,13,20,31,61,97,250,457,41112

RG-Folgen von ‚Prime-OCRONS‘ (sofern sie nicht periodisch sind) kommen aus dem Unendlichen als Primzahlen, ‚tummeln‘ sich ein bisschen unter den ‚endlichen‘ Zahlen, verwandeln sich an genau einem Punkt in zusammengesetzte Zahlen, bleiben von da ab gerade und verschwinden dann wieder im Unendlichen.

Hier ein paar Plots von RG-Folgen („Umwandlungspunkt“ rot markiert, links davon: Primzahlen, rechts davon: zusammengesetzte, gerade Zahlen):

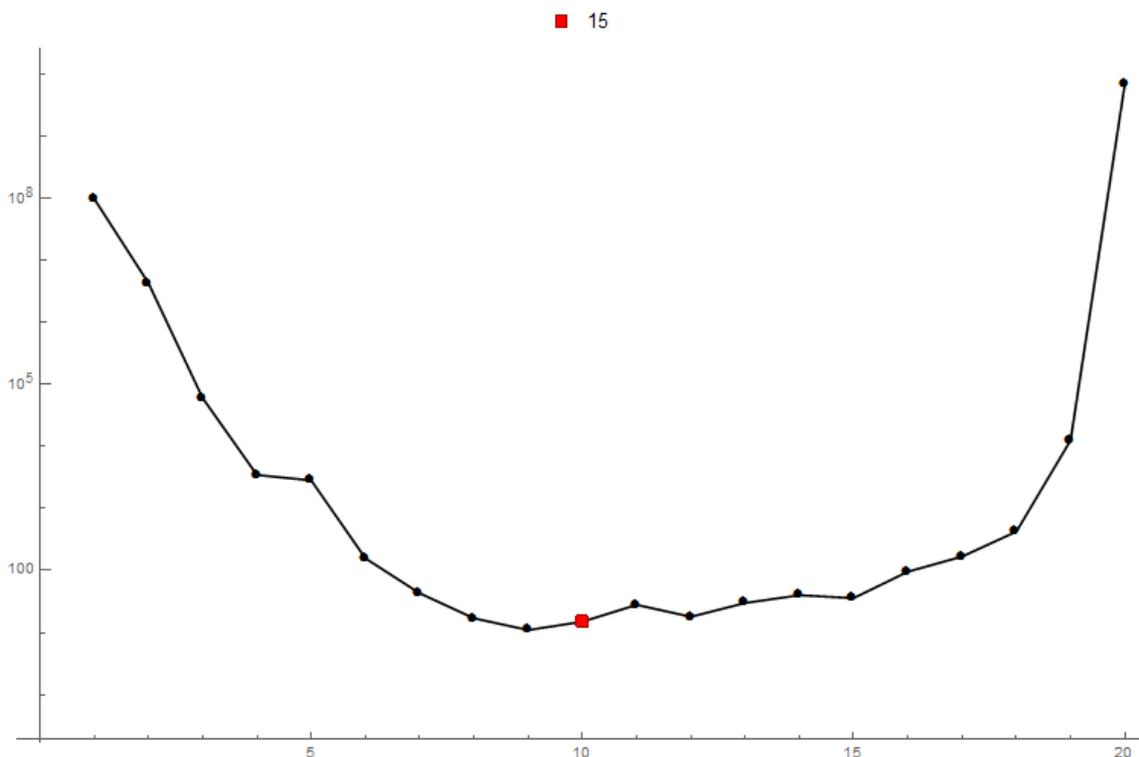


Abbildung 17: RG-Folge ‚15‘ (logarithm. Plot)

Zahlenfolge:

97152271, 4251697, 59723, 3463, 2833, 157, 43, 17, 11, **15**, 28, 18, 30, 40, 36, 96, 168, 424, 12544, 6845104128

RG-zahlen (= rekursiv gödelisiert)

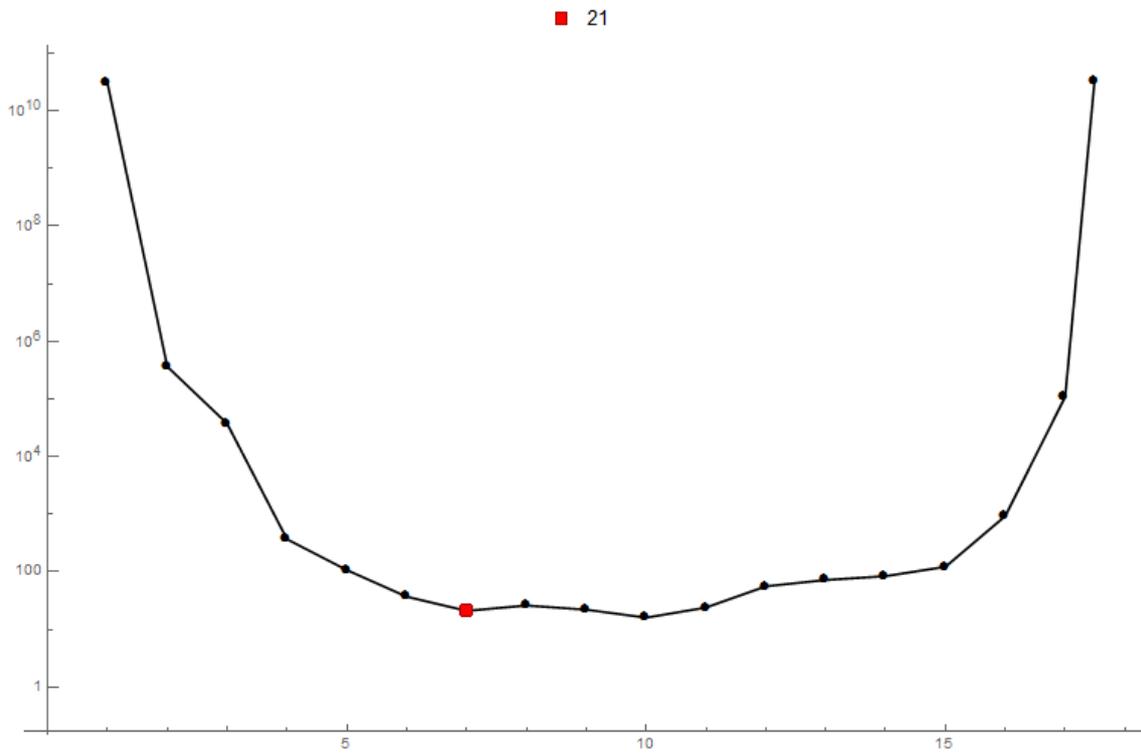


Abbildung 18: RG-Folge ,21' (logarithm. Plot)

30612065639, 363343, 37217, 367, 107, 37, 21, 26, 22, 16, 24, 56, 72, 84, 122, 928, 108544, 15903336184152064

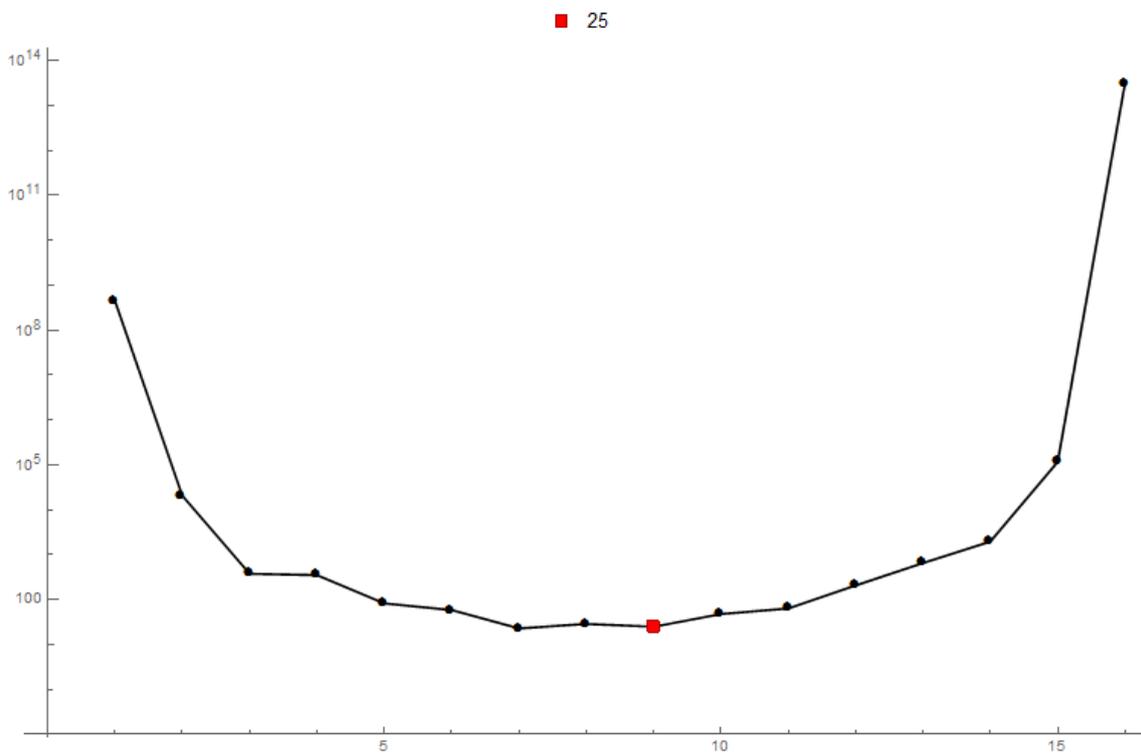


Abbildung 19: RG-Folge ,25' (logarithm. Plot)

448693727, 20719, 379, 353, 83, 59, 23, 29, 25, 48, 64, 208, 656, 1968, 116992, 30889404792832

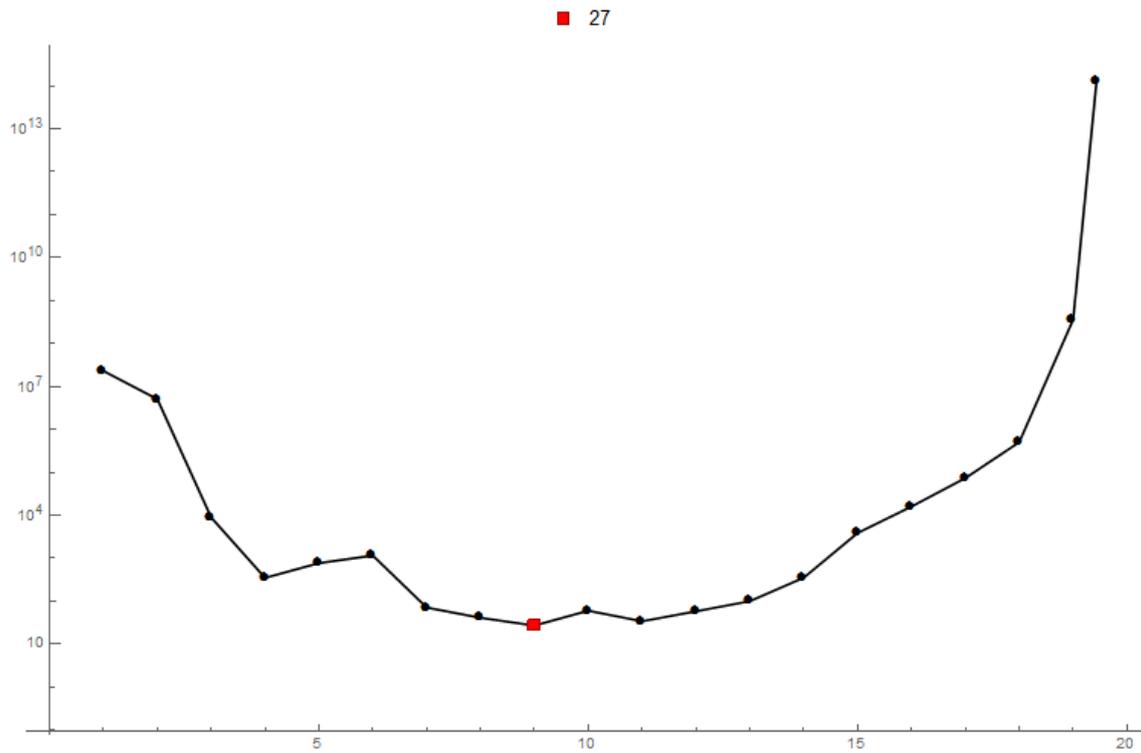


Abbildung 20: RG-Folge ,27' (logarithm. Plot)

22920311, 5065217, 8861, 349, 769, 1153, 71, 41, **27**, 60, 34, 58, 100, 352, 3872, 16016, 73764, 503296, 360710432, 2177877733799636238336

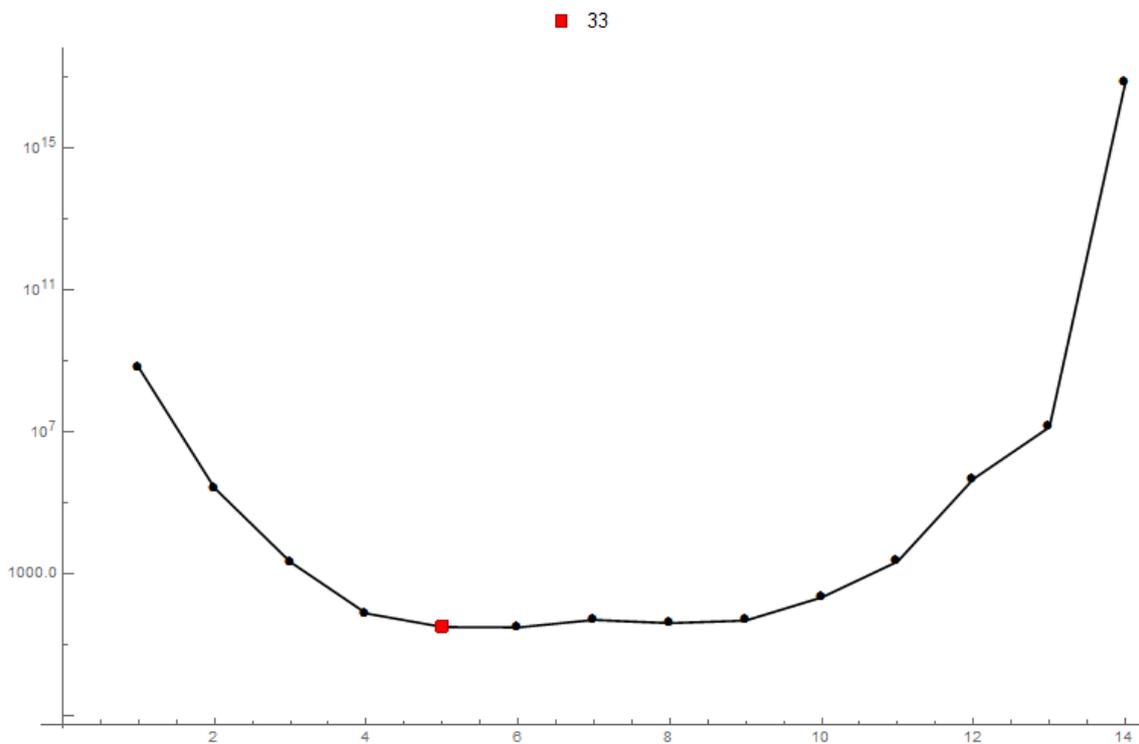


Abbildung 21: RG-Folge ,33' (logarithm. Plot)

680328533, 271003, 2221, 79, **33**, 32, 52, 42, 50, 224, 2304, 491776, 14160388, 70967016210563072

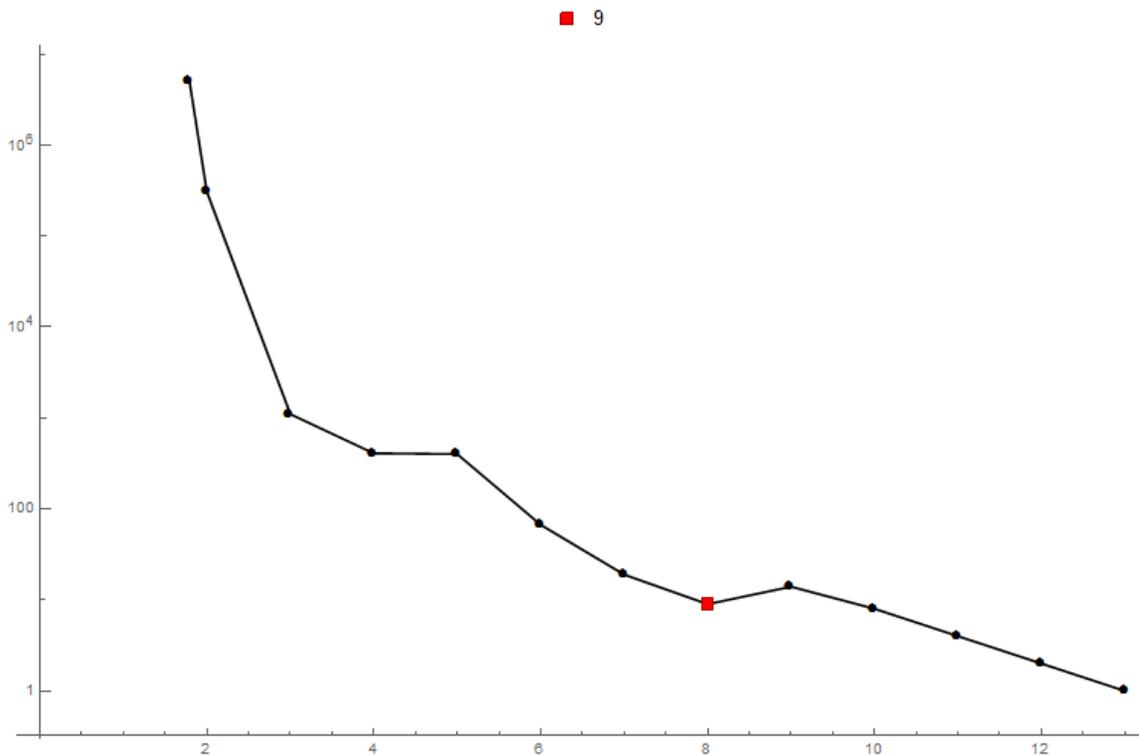


Abbildung 22: RG-Folge ,9' (logarithm. Plot)

210535619933, 305999, 1103, 409, 401, 67, 19, 9, 14, 8, 4, 2, 1, 0

Die Anwendung dieser rekursiven Vorschrift teilt die Menge der natürlichen Zahlen in Klassen ein. Alle Zahlen einer Klasse enden früher oder später auf die gleiche Art und Weise, wie oben beschrieben. Die Ähnlichkeit mit der Situation bei den Aliquot-Sequenzen ist auffallend. Man könnte hier wie bei den Aliquot-Sequenzen von ‚befeundenen‘ Zahlen, die eine Familie bilden, sprechen (siehe Kapitel 20.9.2.2).

4.17.2 GOCRON-TYP 4 (MIT DEN SYMBOLEN ‚2‘, ‚*‘, ‚P‘, ‚^‘)

Hierfür brauchen wir die Definition des GOCRON-Typs 4 (mit den Gödel-Codes ‚*‘=0, ‚P‘=1, ‚2‘=2 und ‚^‘=3, siehe Kapitel 10.2.2).

Eine **RG-Folge** sei in Richtung positiver Indices wie folgt rekursiv definiert:

$$a(0) = m \quad (\text{mit einer beliebigen ganzen Zahl } m \geq 1)$$

$$a(n + 1) = \text{EGOCRON4}(a(n))$$

Die Folge kann auch in Richtung negativer Indices fortgesetzt werden:

$$a(0) = m \quad (\text{mit einer beliebigen ganzen Zahl } m \geq 1)$$

$$a(n - 1) = \text{INVEGOCRON4}(a(n))$$

RG-Folgen der ‚Typ4-EGOCRONs‘ besitzen folgende Eigenschaften:

- 1) In Richtung negativer Indices ($n \rightarrow \text{EGOCRON4}$) wächst die RG-Folge schneller als exponentiell (siehe ABBILDUNG 23), und zwar für alle Startwerte > 2 .
- 2) In Richtung positiver Indices ($\text{EGOCRON4} \rightarrow n$) endet jede RG-Folge bei dem konstant bleibenden Wert ‚6‘. Zuvor kann die Folge jedoch astronomisch hohe

- Werte annehmen, bevor sie schließlich beim Wert 6 endet. Dies ist eine Vermutung, die auf den empirischen Daten beruht, und nicht bewiesen ist.
- 3) Es gibt einen ‚Umkehrwert‘, ab dem alle Folgenglieder gerade bleiben.

Hier ein paar Graphen von RG-Folgen in Richtung negativer Indices mit verschiedenen Startwerten:

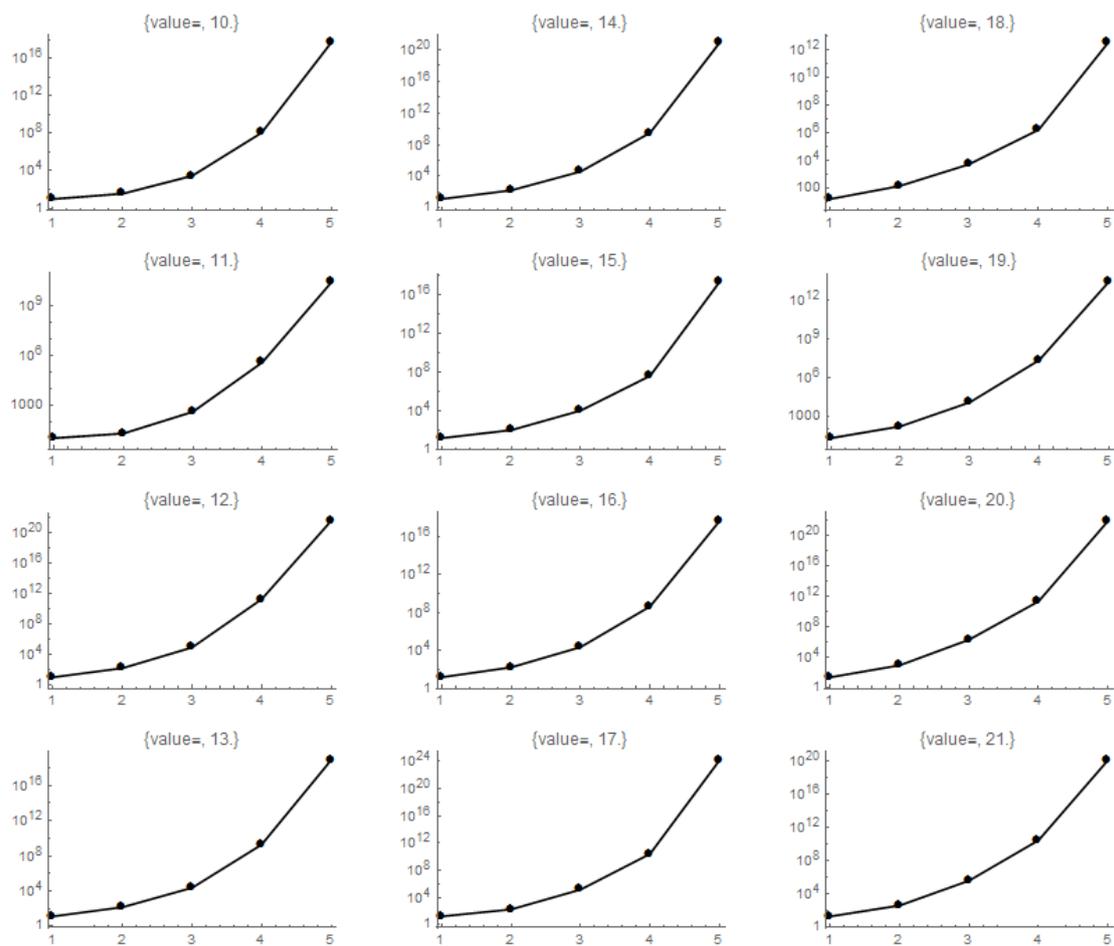


Abbildung 23: RG-Folgen in negativer Richtung (Typ EGOCRON4)

RG-zahlen (= rekursiv gödelisiert)

Hier ein paar Graphen von RG-Folgen in Richtung positiver Indices mit verschiedenen Startwerten (Umkehrwert, ab dem alle Werte gerade bleiben, ist mit angegeben):

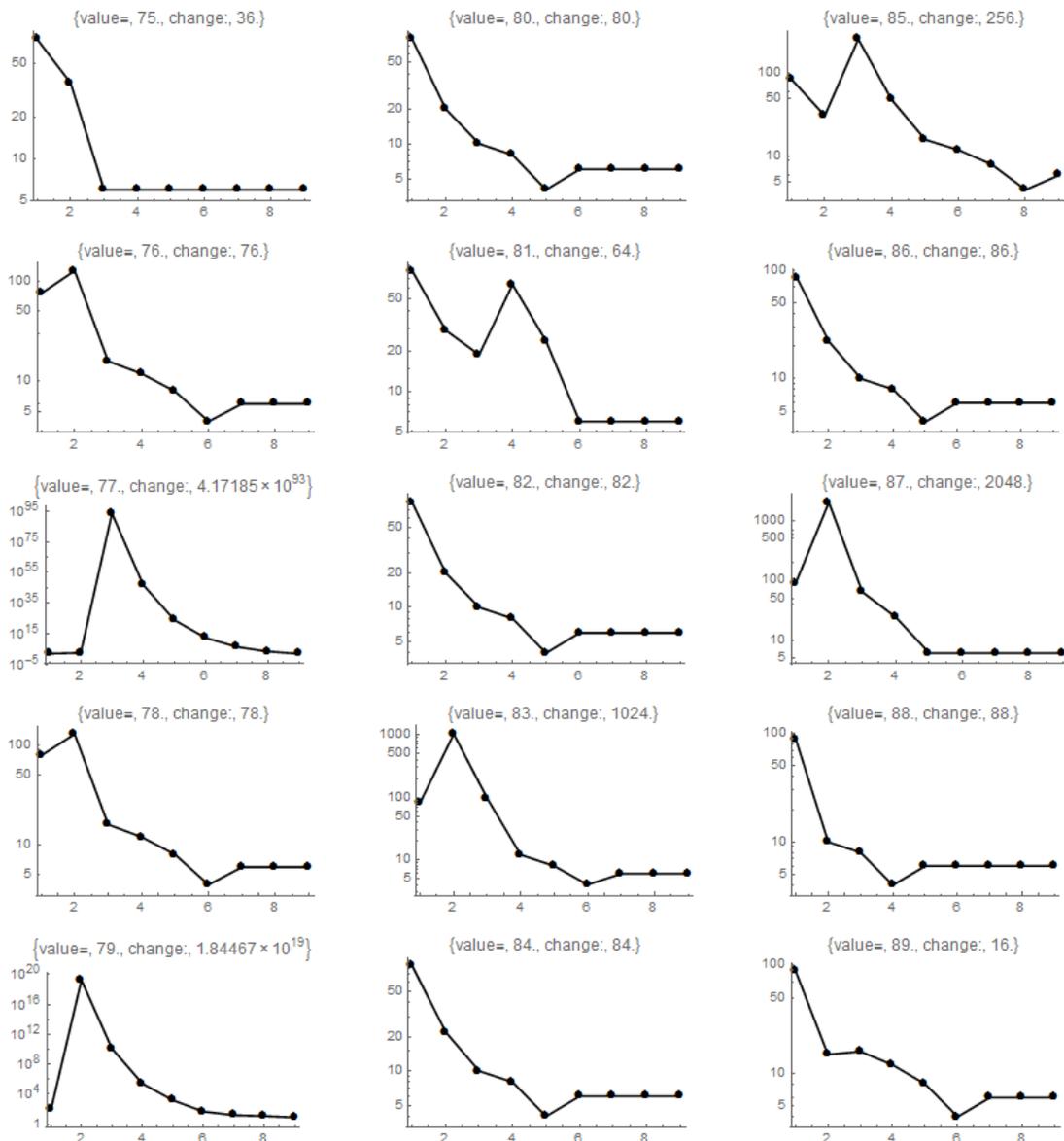


Abbildung 24: RG-Folgen in positiver Richtung (Typ EGOCRON4): Sie enden immer bei 6

Weitere Abbildungen zu diesem Thema finden sich im Anhang 20.7.

Mathematica-Programme, mit denen die Grafiken erstellt wurden, befinden sich im ebenfalls im Anhang.

Auch hier gilt das Gleiche wie im vorherigen Kapitel: Die Anwendung dieser rekursiven Vorschrift teilt die Menge der natürlichen Zahlen in Klassen ein. Alle Zahlen einer Klasse sind ‚befreundet‘ und enden auf die Gleiche Art und Weise.

5 EINSCHUB: DIE RIEMANNSCHE ZETA-FUNKTION $\zeta(s)$

5.1 ALLGEMEINES

Die Riemannschesche Zeta-Funktion ist eines der Mysterien in der Mathematik. Besonders rätselhaft sind ihre Nullstellen. Es gibt die sogenannten trivialen Nullstellen im reellen Bereich. Diese Nullstellen liegen bei den geraden, negativen ganzzahligen Werten $-2, -4, -6, \dots, (-2n)$. Es gibt jedoch auch im Komplexen unendlich viele Nullstellen, die alle auf der sogenannten ‚kritischen‘ Geraden $Re(s) = \frac{1}{2}$ liegen. Es gibt keine einfache Formel für die Lage dieser Nullstellen. Sie sind scheinbar chaotisch und zufällig verteilt und besitzen daher ähnlich geheimnisvolle Eigenschaften wie die Primzahlen. Tatsächlich ist die Lage der nicht-trivialen Nullstellen sehr eng mit der Verteilung der Primzahlen verbunden (siehe z. B. Kapitel 1.1, Formel (131)).

Die Behandlung der Eigenschaften der Zeta-Funktion füllt ganze Bücher, deshalb soll hier nicht weiter darauf eingegangen werden. Vereinfacht kann man sagen, dass aus der Kenntnis der nicht-trivialen Nullstellen die Lage (und Verteilung) der Primzahlen berechnet werden kann und umgekehrt. Diese Beziehung ist jedoch keine einfache ein-zu-eins-Beziehung zwischen Nullstellen und Primzahlen, sondern ähnelt eher einer Transformation (wie z. B. der Fourier-Transformation, die die Verbindung zwischen Zeitbereich und Frequenzbereich herstellt). So könnte man auch hier von verschiedenen Bereichen, wie z.B. ‚Primzahl-Bereich‘ und ‚Zeta-Bereich‘ sprechen.

Wir müssen hier präzisieren: Die Aussage, dass alle nicht-trivialen Nullstellen auf der kritischen Geraden $Re(s) = \frac{1}{2}$ liegen (das ist die berühmte ‚Riemannschesche Vermutung‘) ist bisher weder bewiesen noch widerlegt worden. Sie ist jedoch wahrscheinlich wahr. Hierfür gibt es wahrhaft überwältigende numerische Belege. Obwohl so einfach zu formulieren, hat die ‚Riemannschesche Vermutung‘ bisher allen Versuchen (auch der größten Mathematiker) widerstanden, bewiesen zu werden! So könnte es theoretisch sein, dass es in unvorstellbar hohen Zahlenbereichen Nullstellen gibt, die abseits der kritischen Geraden liegen... Die Geschichte der Mathematik hat uns gelehrt, dass wir der numerischen Evidenz nicht immer vertrauen dürfen (man denke nur an die Vermutung, dass die asymptotische Formel für die Primzahlzählfunktion $Li(x) \approx \pi(x)$ immer zu große Werte liefert³³ (ganz sicher bis 10^{26}). Denn: Es ist bewiesen worden (Skewes, 1933), dass $Li(x) < \pi(x)$ werden kann. Tatsächlich wechselt die Differenz $Li(x) - \pi(x)$ unendlich oft das Vorzeichen. Skewes bewies, dass der Punkt des ersten Vorzeichenwechsels kleiner ist als $10^{10^{34}}$! Diese obere Grenze wurde inzwischen erheblich reduziert auf $e^{727,95133}$.

Dennoch: Kein Mathematiker glaubt, dass die Riemannschesche Vermutung falsch ist! Dass sie jedoch trotz großer Anstrengungen bisher nicht bewiesen werden konnte, verleitet manchen zur Aussage, sie gehöre in die Kategorie der unbeweisbaren mathematischen Aussagen (spätestens seit Gödel wissen wir, dass es wahre, jedoch nicht beweisbare mathematische Sätze gibt). Die Aussage wiederum ist von Gödel bewiesen worden. Das bietet natürlich Stoff für philosophische Diskussionen: Wieso hat der liebe Gott bei der Erschaffung der Zahlen und der Mathematik Gesetzmäßigkeiten vorgegeben, so dass

³³ https://en.wikipedia.org/wiki/Prime-counting_function

Allgemeines

etwas ‚wahr‘ sein kann, ohne dass es einen zwingenden Grund dafür gibt? Gäbe es einen zwingenden Grund, dann existierte auch ein Beweis dafür...

Für Liebhaber: Ein paar besondere Werte der Zeta-Funktion, die durch explizite Formeln darstellbar sind:

$$\zeta(2) = \frac{\pi^2}{6}$$

$$\zeta(3) = \frac{5}{4} \text{Li}_3\left(\frac{1}{\tau^2}\right) + \frac{1}{6} \pi^2 \ln \tau - \frac{5}{6} \ln^3 \tau$$

$$\zeta(3) = \frac{6}{d(0) - \frac{1^6}{d(1) - \frac{2^6}{d(2) - \frac{3^6}{d(3) - \ddots}}}}$$

(mit $d(n) = 34n^3 + 51n^2 + 27n + 5$)

(mit dem Wert des goldenen Schnittes $\tau = \frac{1+\sqrt{5}}{2}$ sowie $\text{Li}_3(x)$: Der Polylogarithmus 3. Ordnung)³⁴

Die sogenannte Prim-Zeta-Funktion $P(s) = \sum_{p \text{ prime}} \frac{1}{p^s}$

kann aus der ‚normalen‘ Zeta-Funktion leicht berechnet werden:

$$P(s) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} \ln \zeta(ns) \quad (46)$$

Die ersten 20 ‚nicht-trivialen‘ Nullstellen der Zeta-Funktion entlang der kritischen Geraden lauten (mit einer Genauigkeit von 10 Dezimalstellen):

{14.13472514, 21.02203964, 25.01085758, 30.42487613,
32.93506159, 37.58617816, 40.91871901, 43.32707328,
48.00515088, 49.77383248, 52.97032148, 56.44624770,
59.34704400, 60.83177852, 65.11254405, 67.07981053,
69.54640171, 72.06715767, 75.70469070, 77.14484007}

Mathematica-Code:

```
Table[N[Im[ZetaZero[n]],10],{n,20}]
```

Entlang der kritischen Geraden ist es praktisch, die Zeta-Funktion wie folgt aufzusplitten:

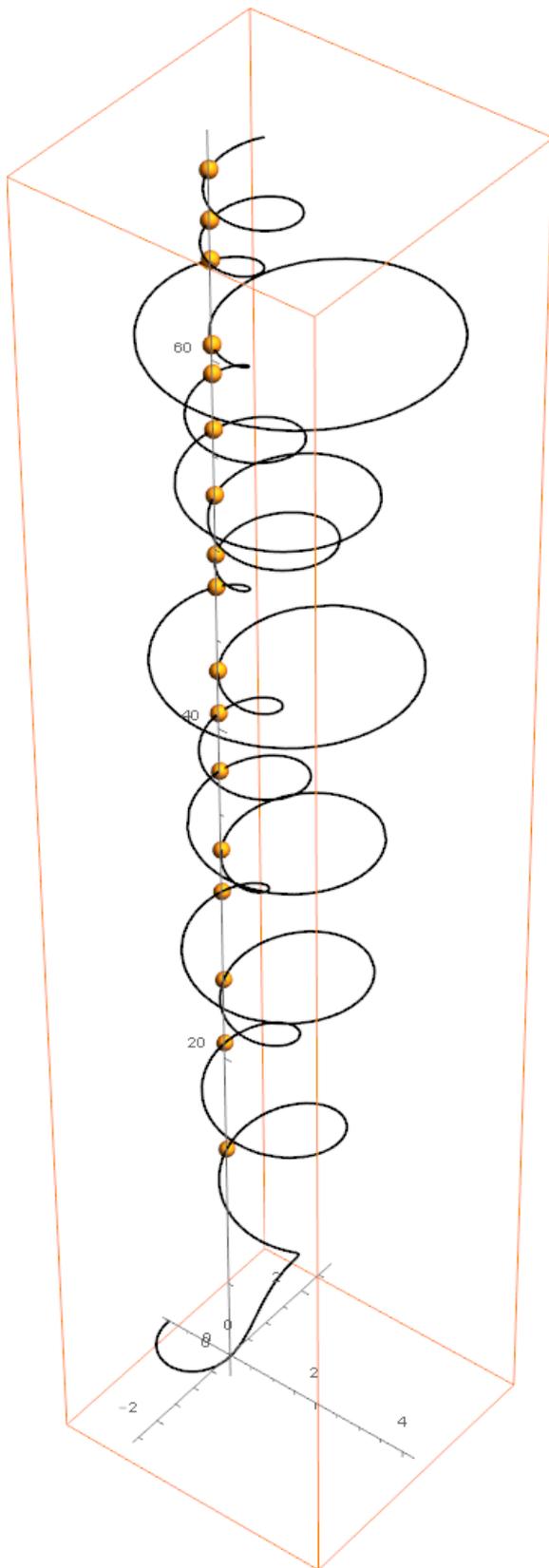
$$\zeta\left(\frac{1}{2} + it\right) = Z(t) e^{-i\theta(t)} \quad (47)$$

³⁴ Journal of Computational and Applied Mathematics 121 (2000) 247-296

wobei $Z(t)$ und $\vartheta(t)$ die Riemann-Siegel Funktionen sind.

Über die Zeta-Funktion sind viele Bücher geschrieben worden. Ein paar Beispiele aus dem Literaturverzeichnis: (Edwards, 1974) oder auch (Sautoy, 2004).

Hinweis: Die Riemannsche Zeta-Funktion besitzt Verallgemeinerungen (z.B. die ‚Hurwitzsche‘ bzw. die ‚Lerchsche‘ Zetafunktionen, die meistens nach ihren Entdeckern benannt wurden. Von diesen verallgemeinerten Zetafunktionen werden in der mathematischen Literatur mindestens zwölf Versionen erwähnt.



Die Zeta-Funktion mit Nullstellen als parametrischer 3D-Plot:

Die Zetafunktion entlang der kritischen Geraden ist komplexwertig. Die kritische Gerade geht nach oben und der komplexe Funktionswert bewegt sich in der x-y-Ebene. Die Nullstellen sind als kleine Kugeln markiert. Die Zetafunktion schraubt sich gegen den Urzeigersinn nach oben und schneidet an den Nullstellen die nach oben gehende z-Achse.

Das Mathematica-Programm zum Erzeugen der Graphik befindet sich im Anhang ‚Riemannsche Zeta-Funktion‘.

Abbildung 25: Parametrischer 3D-Plot (Re-und Im-Teil) der Zetafunktion mit Nullstellen

Allgemeines

Die nächste Grafik zeigt wie ABBILDUNG 26 die Zetafunktion entlang der kritischen Geraden (rot: Betrag, schwarz: Phase (Argument)), jedoch als 2D-Plot. Das Argument der Zetafunktion hängt eng mit der Riemann-Siegel-Funktion $\vartheta(t)$ zusammen (siehe (47)).

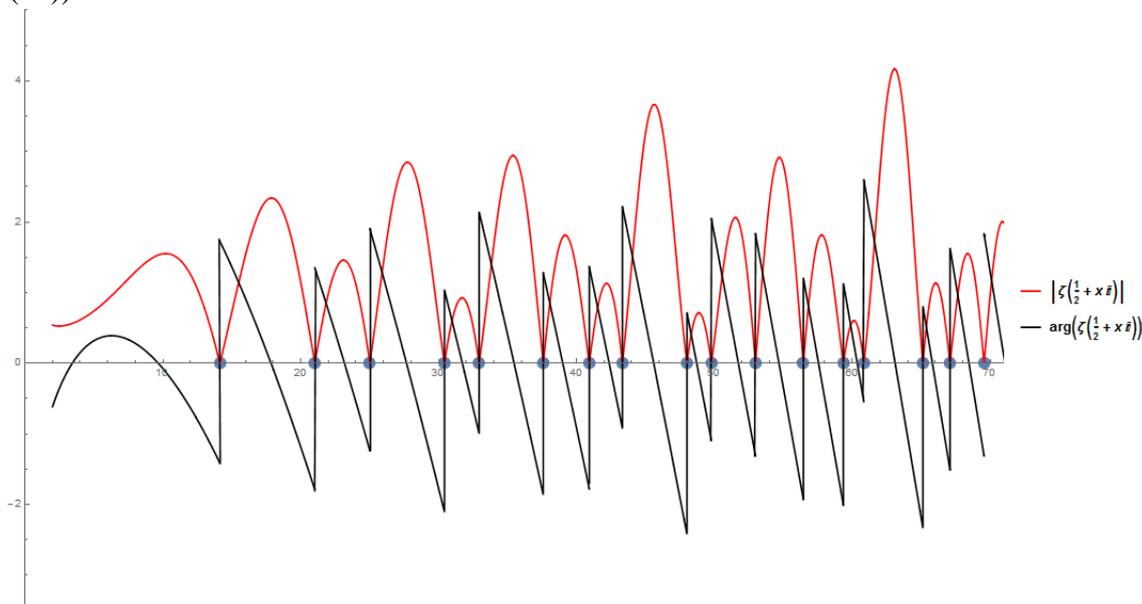


Abbildung 27: Betrag und Phase der Zetafunktion entlang der krit. Geraden (0-70)

Die Phase der Zetafunktion macht an den Nullstellen Sprünge um den Wert $+\pi$. Die Phase einer Funktion ist nur im Intervall $[-\pi, +\pi]$ definiert. Daher ist sie auf diesen Wertebereich beschränkt. Die Riemann-Siegel-Funktion $\vartheta(t)$ beschreibt jedoch eine kontinuierliche Phase. Wegen der Mehrdeutigkeit von $e^{-i\vartheta(t)}$ ergeben sich die gleichen Werte. Man kann sagen, dass sich die stetigen Stücke der Phase der Zetafunktion entlang der kritischen Geraden durch Verschieben mit der überall stetigen Riemann-Siegel-Funktion $-\vartheta(t)$ zur Deckung bringen lassen. Z. B. im folgenden Graph zwischen der ersten und zweiten Nullstelle (mit Werten von ca. 14 bis 21).

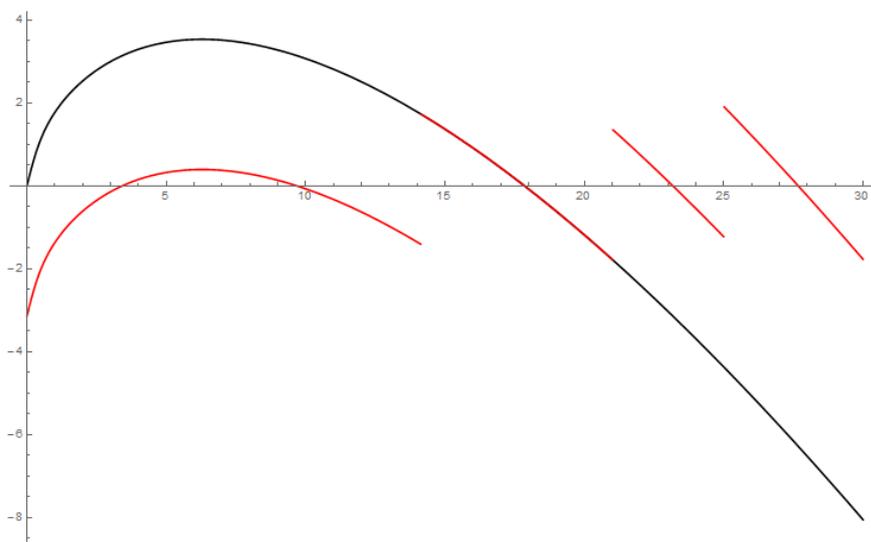


Abbildung 28: Vergleich der Phase der Zetafunktion mit der (negativen) Riemann-Siegel Funktion.

Die Riemann-Siegel-Funktion $\vartheta(t)$ lässt sich erstaunlicherweise ohne Kenntnis der Zeta-Funktion, nur mit Hilfe der Gamma-Funktion berechnen. Sie ist relativ 'unauffällig', hat jedoch Sprünge, da die Argumente nur Werte zwischen π und $-\pi$ annehmen können:

$$\vartheta(t) = \operatorname{Im} \left(\ln \left(\Gamma \left(\frac{1}{4} + \frac{it}{2} \right) \right) \right) - \frac{t}{2} \ln \pi = \operatorname{Arg} \left(\Gamma \left(\frac{1}{4} + \frac{it}{2} \right) \right) - \frac{t}{2} \ln \pi \quad (48)$$

Hinweis: Bei der Berechnung mit Mathematica sollte für den Term $\ln(\Gamma(\dots))$ nicht die normale Γ -Funktion hergenommen werden, sondern die Funktion 'LogGamma'. Der Grund dafür ist, dass die Zweigstruktur im Komplexen bei der normalen Gamma-Funktion komplizierter ist und man nur den Hauptwert des Logarithmus bekommen würde. Die Funktion 'LogGamma' umgeht dieses Problem.

Wenn wir die Phase von $\zeta\left(\frac{1}{2} + it\right)$ mit dem Faktor $\frac{1}{\pi}$ normieren bekommen wir an jeder Nullstelle einen Sprung von $+1$. Wenn wir das gleiche mit $-\vartheta(t)$ tun und diese beiden Funktionen von einander subtrahieren (und 1 addieren) bekommen wir eine Zählfunktion für die Nullstellen der Zetafunktion! Genauereres hierzu im Kapitel 5.5.

5.2 DIE VERSCHIEDENEN DARSTELLUNGEN VON $\zeta(s)$

In ein Buch über Primzahlen gehören die wichtigsten Darstellungen der Zeta-Funktion: Zunächst einmal ist die ursprüngliche Definition der Zeta-Funktion bestehend einfach, so dass man keine höhere Schulbildung braucht, um sie zu verstehen:

$$\zeta(s) = 1 + \frac{1}{2^s} + \frac{1}{3^s} + \dots = \sum_{n=1}^{\infty} \frac{1}{n^s} \quad (\operatorname{Re}(s) > 1) \quad (49)$$

Schon Euler bewies im 18. Jahrhundert, dass $\zeta(2) = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots = \frac{\pi^2}{6}$.

Für alle geradzahligen, positiven Argumente gibt es die einfache Formel:

$$\zeta(2n) = (-1)^{n-1} \frac{(2\pi)^{2n} B_{2n}}{2(2n)!} \quad (\text{mit } B_n: \text{Bernoulli-Zahlen, } n = 1, 2, \dots) \quad (50)$$

Für die positiven ungeradzahligen Argumente gibt es auch Formeln, die jedoch etwas komplizierter sind, z. B.:

$$\zeta(3) = \frac{7\pi^3}{180} - 2 \sum_{n=1}^{\infty} \frac{1}{n^3(e^{2\pi n} - 1)} \quad (51)$$

Für negative ganzzahlige Argumente:

$$\zeta(1-n) = -\frac{B_n}{n} \quad (52)$$

Die verschiedenen Darstellungen von $\zeta(s)$

So ist z. B. $\zeta(0) = -\frac{1}{2}$, $\zeta(-1) = -\frac{1}{12}$, $\zeta(-3) = -\frac{1}{120}$

Auch die Produktformel (aus der mit elementaren Methoden (49) hergeleitet werden kann) ist einfach zu verstehen:

$$\zeta(s) = \frac{1}{\underbrace{\left(1 - \frac{1}{2^s}\right)\left(1 - \frac{1}{3^s}\right)\left(1 - \frac{1}{5^s}\right) \dots}_{> 1}} = \prod_{p \in \mathbb{P}} \frac{1}{\left(1 - \frac{1}{p^s}\right)} \quad (\operatorname{Re}(s) > 1) \quad (53)$$

Äußerst bemerkenswert ist der von dem russischen Mathematiker Woronin bewiesene Satz, dass die Zetafunktion jede beliebige andere Funktion beliebig genau approximieren kann (genauer: jede holomorphe komplexe Funktion innerhalb eines Gebiets mit Radius $\frac{1}{4}$, ohne Nullstellen).

Anschaulich gesprochen: Jede komplexe Funktion, so chaotisch und kompliziert ihre Landschaft mit allen ‚Hügeln‘ und ‚Tälern‘ sein mag, taucht ‚irgendwann‘ auch in der Landschaft der komplexen Zeta-Funktion auf, wenn man nur weit genug in den unendlichen Landschaften der Zeta-Funktion sucht...

Die Ableitung der Zeta-Funktion hängt mit der Von-Mangoldt-Funktion $\Lambda(n)$ zusammen:

$$-\frac{\zeta'(s)}{\zeta(s)} = \sum_{n=1}^{\infty} \frac{\Lambda(n)}{n^s} \quad (54)$$

$$\zeta'(0) = -\frac{1}{2} \ln 2\pi \quad (55)$$

$$\zeta'(-2n) = (-1)^n \frac{\zeta(2n+1)(2n)!}{2^{2n+1}\pi^{2n}} \quad (56)$$

Weitere Formeln:

$$\frac{1}{\zeta(s)} = \sum_{n=1}^{\infty} \frac{\mu(n)}{n^s} \quad (\text{mit Möbiusfunktion } \mu(n)) \quad (57)$$

Integral-Darstellung:

$$\zeta(s) = \frac{1}{\Gamma(s)} \int_0^{\infty} x^{s-1} \sum_{n=1}^{\infty} e^{-nx} dx = \frac{1}{\Gamma(s)} \int_0^{\infty} \frac{x^{s-1}}{e^x - 1} dx \quad (58)$$

Weitere Produktdarstellungen (Hadamard-Produkt):

$$\zeta(s) = \frac{\pi^{\frac{s}{2}}}{2(s-1)\Gamma(1+\frac{s}{2})} \prod_{\rho} \left(1 - \frac{s}{\rho}\right) \quad (59)$$

Wegen der bedingten Konvergenz dieser Formel (die Produktbildung über die Terme mit den Nullstellen der Zetafunktion ρ muss paarweise erfolgen) kann man auch schreiben:

$$\zeta(s) = \frac{\pi^{\frac{s}{2}}}{2(s-1)\Gamma(1+\frac{s}{2})} \prod_{\text{Im}(\rho)>0} \left(1 - \frac{s}{\rho}\right) \left(1 - \frac{s}{1-\rho}\right) \quad (60)$$

5.3 DIE PRODUKTDARSTELLUNG VON $\zeta(s)$ IM KOMPLEXEN

Gleichung (53) beschreibt die Produktdarstellung der Zeta-Funktion. Sie gilt auch im Komplexen, konvergiert jedoch nur für $\text{Re}(s) > 1$. Deshalb sollte man lieber ‚die Finger davon lassen‘, diese Formel auch im Bereich der ‚kritischen Geraden‘, in der die nicht-trivialen Nullstellen liegen, anzuwenden, oder? Was passiert, wenn man das ‚Verbotene‘ trotzdem tut? Überschreiten wir dann eine Grenze, die der ‚seriöse‘, ernsthafte Mathematiker nicht überschreiten sollte? Riskiert man, nicht ‚ernst genommen‘ zu werden? Zeigen wir doch etwas mathematische Abenteuer-Lust und schauen uns an was passiert, wenn man das ‚Verbotene‘ trotzdem tut:

Wir verwenden Formel (53) und berechnen damit die Werte entlang der kritischen Geraden:

$$\zeta\left(\frac{1}{2} + t \cdot i\right) = \prod_{p \in \mathbb{P}} \frac{1}{\left(1 - p^{-\frac{1}{2}-t \cdot i}\right)} \quad \text{mit } t \geq 0$$

Als erstes fällt auf, dass der Term $\frac{1}{(\dots)}$ im unendlichen Produkt nie 0 werden kann.

Dennoch nimmt entlang der kritischen Geraden $\zeta\left(\frac{1}{2} + t \cdot i\right)$ unendlich oft den Wert 0 an. Wie soll das zusammen gehen? Schauen wir uns Realteil (schwarz) und Imaginärteil (rot), sowie die Nullstellen (blaue Kreise) im Bereich von 2 bis 70 an:

Die Produktdarstellung von $\zeta(s)$ im Komplexen

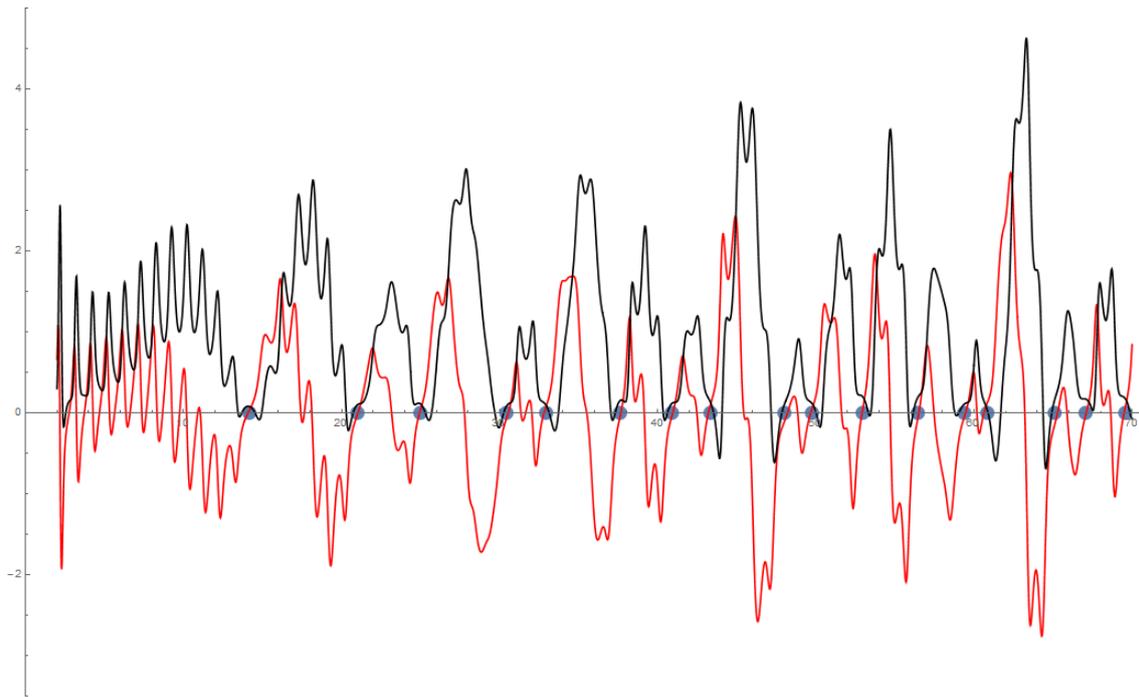


Abbildung 29: Zeta-Funktion (auf der krit. Linie, $t=0-70$, Produktformel mit 100 Primzahlen)

Mathematica:

```

cterm[n_, x_] := 1 / (1 - Prime[n] ^ (-1/2 - x*I));
myFunc[x_] := Product[cterm[n, x], {n, 1, 100}]
xmax=70;
Show[ListPlot[Table[{Im[ZetaZero[i]], 0}, {i, 1, 17}], PlotRange-
>{{0, 71}, {-3.5, 5}}, ImageSize-
>Large], Plot[{Im[myFunc[x]], Re[myFunc[x]]}, {x, 2, xmax}, PlotStyle-
>{Red, Black}, PlotRange->{{0, 71}, {-3.5, 5}},
PlotLegends->{TraditionalForm[Im[Product[1 / (1 - Prime[n] ^ (-1/2 -
x*I)), {n, 1, N}]]], TraditionalForm[Re[Product[1 / (1 - Prime[n] ^ (-1/2 -
x*I)), {n, 1, N}]]]}, ImageSize->Large]]

```

Betrachten wir den Betrag dieser Funktion, dann sehen wir klar, dass die mit der Produktformel berechneten Werte an den Nullstellen (blaue Kreise) der Zeta-Funktion klare, deutliche Minima haben, jedoch nie exakt 0 werden (was ja aufgrund der Formel klar ist). Irgendwie scheinen sich die unendlich vielen Faktoren so zusammenzufügen, dass das unendliche Produkt an den Nullstellen dennoch dem Wert 0 beliebig nahe kommt, wenn man das Produkt nur über genügend viele Faktoren laufen lässt:

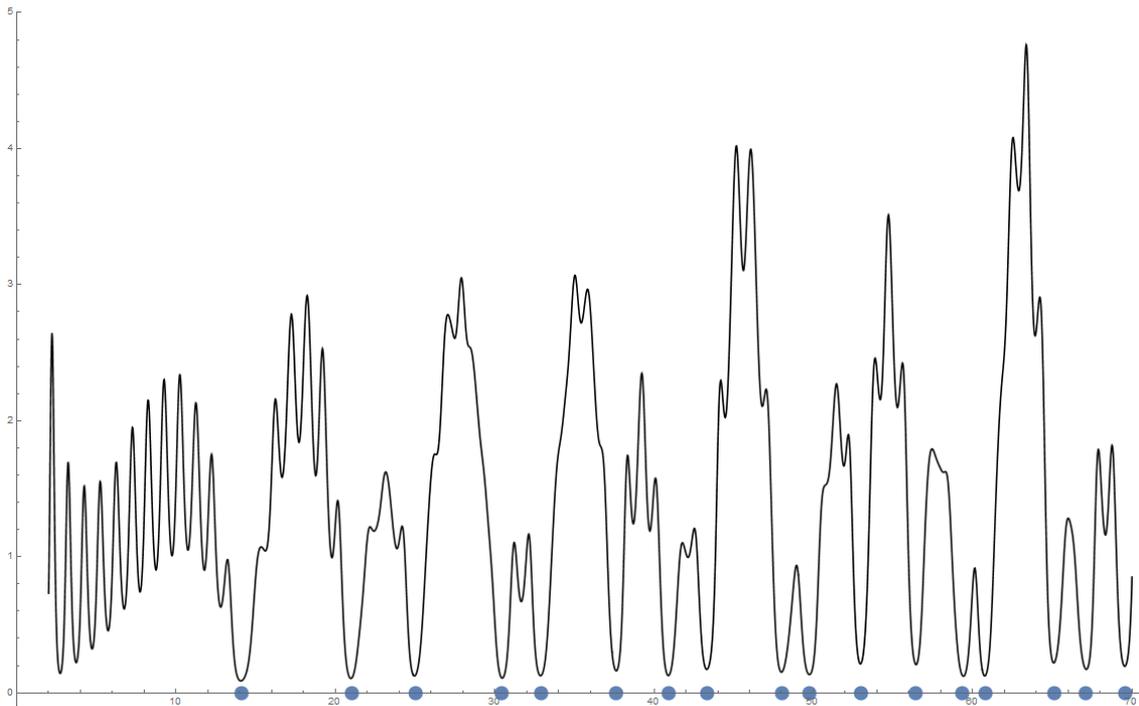


Abbildung 30: Zeta-Funktion (Betrag, auf krit. Linie, $t=0-70$, Produktformel mit 100 Primzahlen)

Mathematica:

```

cterm[n_, x_] := 1 / (1 - Prime[n]^(-1/2 - x*I));
myFunc[x_] := Product[cterm[n, x], {n, 1, 100}]
Show[ListPlot[Table[{Im[ZetaZero[i]], 0}, {i, 1, 17}], PlotRange -> {{0, 71}, {-0.1, 5}}, Plot[Abs[myFunc[x]], {x, 2, xmax}, PlotStyle -> Black], PlotRange -> {{0, 71}, {-0.1, 5}}]

```

Hier im Vergleich dazu die ‚exakte‘ Zeta-Funktion:

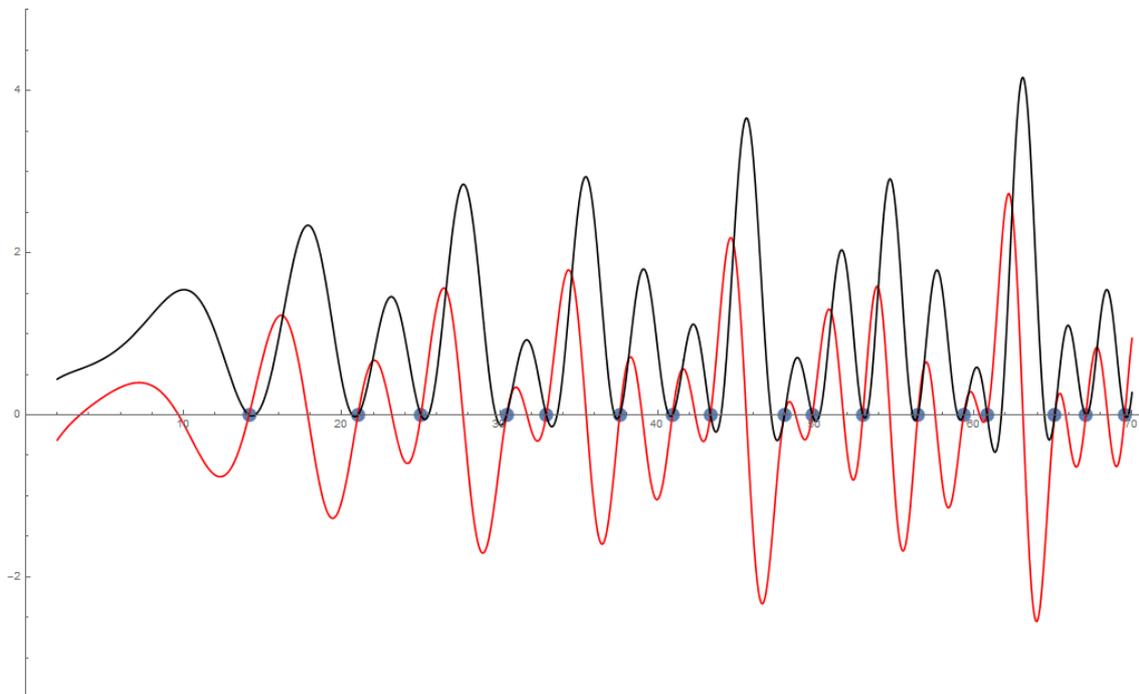


Abbildung 31: Zeta-Funktion (Real- und Imaginärteil, krit. Linie, $t=0-70$, exakte Formel)

Die Produktdarstellung von $\zeta(s)$ im Komplexen

Mathematica:

```
xmax=70;  
Show[ListPlot[Table[{Im[ZetaZero[i]],0},{i,1,17}],PlotRange->{{0,71},{-3.5,5}},ImageSize->Large],  
Plot[{Im[Zeta[1/2+x I]],Re[Zeta[1/2+x I]]},{x,2,xmax},  
PlotStyle->{Red,Black},PlotRange->{{0,71},{-3.5,5}},  
PlotLegends->"Expressions",ImageSize->Large]]
```

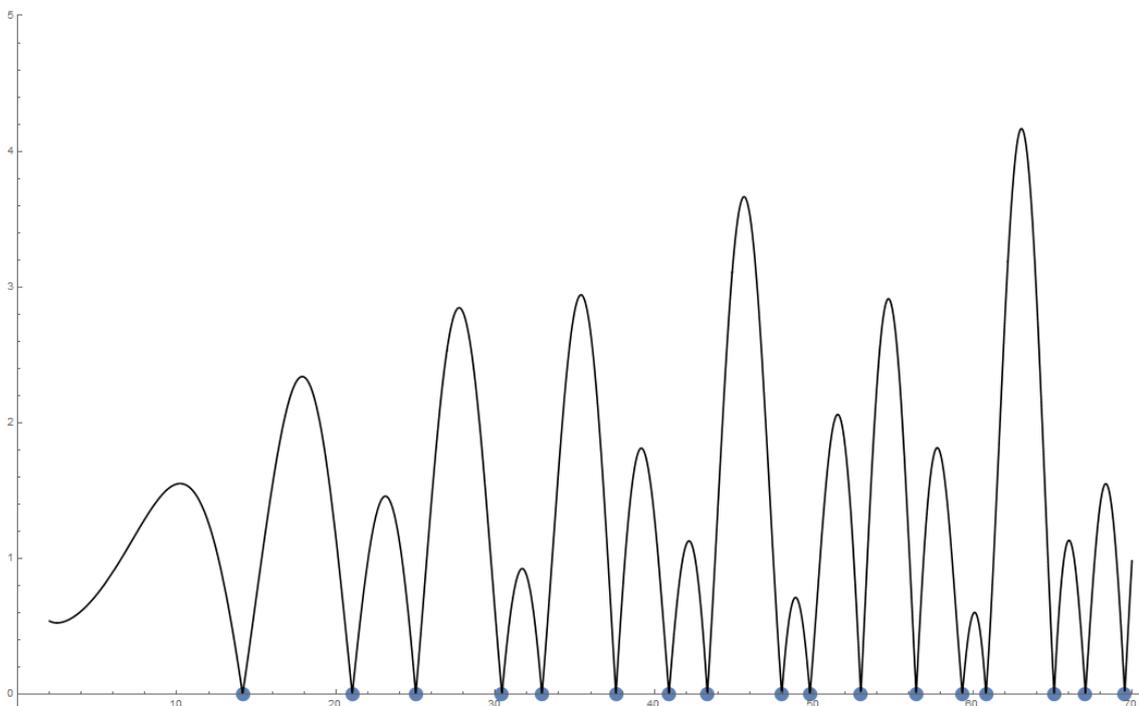


Abbildung 32: Zeta-Funktion (Betrag, krit. Linie, $t=0-70$, exakte Formel)

Vergleich der exakten Zeta-Funktion mit der aus der Produktformel berechneten Funktion:

Rein phänomenologisch lassen sich folgende Aussagen treffen, ohne Anspruch auf Exaktheit oder Gültigkeitsbereich. Nennen wir der Einfachheit halber die aus der Produktformel berechnete Zeta-Funktion die P-Zeta-Funktion.

Auf den ersten Blick sieht die P-Zeta-Funktion wie eine etwas ‚zerrupfte‘ Zeta-Funktion aus. Sie beginnt immer mehr zu ‚zappeln‘, je mehr Terme n_{max} in der Produktformel mitgenommen werden. Sieht sie für kleine n_{max} noch recht ‚anständig‘ aus, so wird sie für große n_{max} immer bizarrer und ähnelt immer mehr der pathologischen ‚Weierstraßschen‘ \wp -Funktion³⁵, die zwar überall stetig, aber nirgends differenzierbar ist und gelegentlich auch als ‚Monster-Funktion‘ bezeichnet wird. Jedoch lässt sich beobachten, dass die P-Zeta-Funktion in der Gegend der Nullstellen mit steigendem n_{max} tatsächlich gegen 0 geht. Man könnte sagen, dass die P-Zeta-Funktion in der Gegend der Nullstellen ‚lokal‘ konvergiert, wobei wahrscheinlich der Konvergenzbereich ε für $n_{max} \rightarrow \infty$ gegen 0 geht. Ob die P-Zeta-Funktion tatsächlich für alle Werte divergiert, lässt sich nicht sicher sagen. Es scheint im Vergleich zur exakten Zeta-Funktion für

³⁵ <https://de.wikipedia.org/wiki/Weierstraß-Funktion>

wachsendes n_{\max} ein (ebenfalls zunehmendes) ‚Rauschen‘ dazukommen. Das Anwachsen des ‚Rauschens‘ und das ‚Herumzappeln‘ der P-Zeta-Funktion kann in einer Animation betrachtet werden (als Video auf der beiliegenden CD) oder als Mathematica-Animation (siehe Anhang „Riemannsche Zeta-Funktion“).

Es lohnt sich, den Produkt-Term

$$\prod_{p \in \mathbb{P}} \frac{1}{\left(1 - p^{-\frac{1}{2}-t \cdot i}\right)} \quad (61)$$

etwas genauer anzuschauen. Durch Aufsplitten in Real- und Imaginärteil erhalten wir

$$\prod_{n=1}^{\infty} \frac{p_n - \sqrt{p_n} \cos(t \cdot \ln(p_n)) - i\sqrt{p_n} \sin(t \cdot \ln(p_n))}{-2\sqrt{p_n} \cos(t \cdot \ln(p_n)) + p_n + 1}$$

Betrachten wir die Produktbildung rekursiv:

$x_{n+1} + i y_{n+1} = (x_n + i y_n) \cdot \text{Produktterm}(n)$, dann bekommen wir:

$$x_{n+1} = \frac{\sqrt{p_n}(x_n(-\cos(t \cdot \ln(p_n))) + y_n \sin(t \cdot \ln(p_n))) + x_n \sqrt{p_n}}{-2\sqrt{p_n} \cos(t \cdot \ln(p_n)) + p_n + 1}$$

$$y_{n+1} = \frac{\sqrt{p_n}(-x_n \sin(t \cdot \ln(p_n)) + y_n(-\cos(t \cdot \ln(p_n)))) + y_n \sqrt{p_n}}{-2\sqrt{p_n} \cos(t \cdot \ln(p_n)) + p_n + 1}$$

Lassen wir jeweils im Zähler die *Cos*-Terme einfach weg, so erhalten wir eine Iteration, die (zumindest im Bereich der Nullstellen) sehr viel schneller konvergiert

$$x_{n+1} = \frac{x_n + \frac{y_n \sin(t \ln(p_n))}{\sqrt{p_n}}}{-\frac{2 \cos(t \cdot \ln(p_n))}{\sqrt{p_n}} + \frac{1}{p_n} + 1}$$

$$y_{n+1} = \frac{y_n - \frac{x_n \sin(t \cdot \ln(p_n))}{\sqrt{p_n}}}{-\frac{2 \cos(t \cdot \ln(p_n))}{\sqrt{p_n}} + \frac{1}{p_n} + 1} \quad (62)$$

Der absolute Betrag der Funktion zeigt klare, absolute Minima bei den Nullstellen, die sehr nahe bei 0 liegen:

Die Produktdarstellung von $\zeta(s)$ im Komplexen

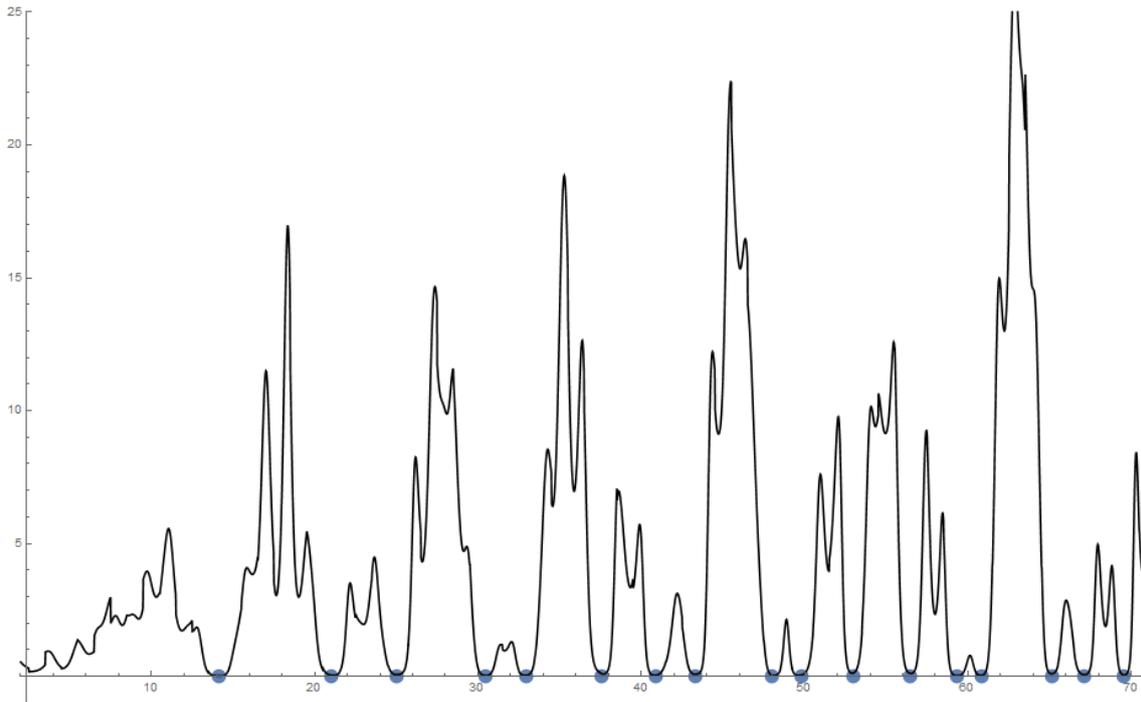


Abbildung 33: Graph nach Formel (62) mit $x_0 = 1, y_0 = 0$, Nullstellen: blaue Kreise

Das entsprechende Mathematica-Programm befindet sich im Anhang „Riemannsche Zeta-Funktion“.

Es ist interessant, dass für die Berechnung der Lage der Nullstellen der Zetafunktion (also die Lage der sich ergebenden absoluten Minima) die genaue Kenntnis aller Primzahlen gar nicht so wichtig ist. Schon das Berücksichtigen der ersten fünf Primzahlen (2,3,5,7,11) in Formel (62) liefert die ungefähre Lage der ersten 15 Nullstellen der Zetafunktion (wenn diese nicht zu dicht beisammen liegen):

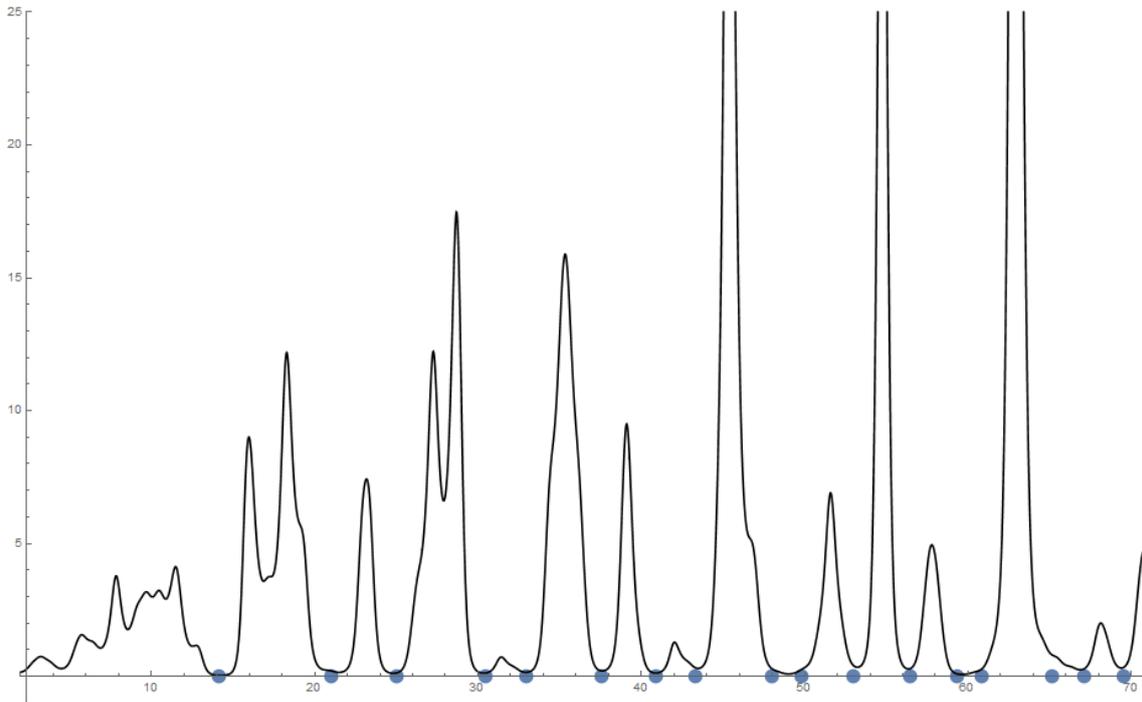


Abbildung 34: Zeta(s) nach (62) unter Verwendung der ersten 5 Primzahlen

5.4 EINE UNERWARTETE PRODUKTDARSTELLUNG EINER ETWAS ANDEREN $\zeta(s)$

Wir bleiben bei Formel (53) und zeigen noch mehr mathematische Abenteuer-Lust. Was passiert, wenn wir im unendlichen Produkt nicht t als Variable betrachten und das Produkt über alle Primzahlen laufen lassen, sondern p als Variable betrachten und das Produkt über alle Nullstellen der Zeta-Funktion laufen lassen?

Also statt:

$$\zeta\left(\frac{1}{2} + t \cdot i\right) = \prod_{n=1}^{\infty} \frac{1}{\left(1 - p_n^{-\frac{1}{2} - t \cdot i}\right)}$$

$$\mathfrak{Z}(p) = \prod_{n=1}^{\infty} \frac{1}{\left(1 - p^{-\frac{1}{2} - \rho_n \cdot i}\right)} \quad \text{mit } \rho_n: \text{Nullstellen von } \zeta(x), p \in \mathbb{R} \quad (63)$$

In Anlehnung an die Zetafunktion nennen wir diese Funktion die \mathfrak{Z} -Funktion.

Hier sind die Ergebnisse:

Betrachten wir den Betrag dieser Funktion, dann sehen wir klar, dass die mit der Produktformel berechneten Werte **bei den Primzahlen (blaue Kreise) klare, deutliche absolute Minima haben**, jedoch nie exakt 0 werden (was ja aufgrund der Formel klar ist).

Eine unerwartete Produktdarstellung einer etwas anderen $\zeta(s)$

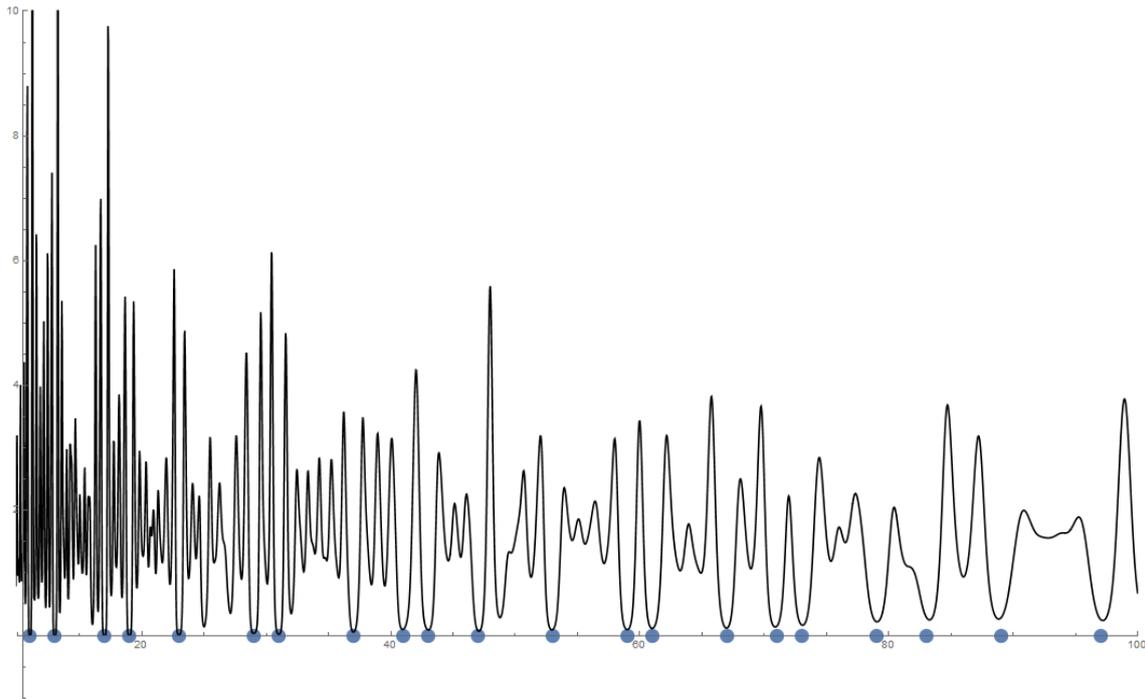


Abbildung 35: Formel (63) (Betrag, $x=10-100$, Produkt mit 100 Nullstellen der Zetafunktion)

```

cterm[n_,p_]:=1/(1-p^(-ZetaZero[n]));
myFunc[p_]:=Product[cterm[n,p],{n,1,100}]
xmin=10;xmax=100;
Show[ListPlot[Table[{Prime[i],0},{i,5,25}],PlotRange->{{xmin,xmax},{-1,10}}],Plot[Abs[myFunc[x]],{x,2,xmax},PlotStyle->Black,PlotRange->{{xmin,xmax},{-5,10}},MaxRecursion->6]]

```

Es wäre interessant, die explizite, exakte Formel für $\mathfrak{Z}(p)$ zu kennen!

Für die Konvergenzeigenschaften von $\mathfrak{Z}(p)$ gilt das Gleiche wie für das unendliche Produkt nach Formel (61). Wir haben auch hier nur ‚lokale‘ Konvergenz in der Gegend der Primzahlen. Je mehr Nullstellenterme das Produkt umfasst, umso mehr beginnt die Funktion zu ‚zappeln‘ und wird ‚verrauscht‘. Verwenden wir auch hier die gleiche Methode der Konvergenzbeschleunigung nach (62), dann bekommen wir:

$$\mathfrak{Z}(p) = \prod_{n=1}^{\infty} \frac{p - \sqrt{p} \cos(z_n \cdot \ln(p)) - i\sqrt{p} \cdot \sin(z_n \cdot \ln(p))}{-2\sqrt{p} \cdot \cos(z_n \cdot \ln(p)) + p + 1}$$

Wobei z_n die Imaginärteile der Nullstellen der Zetafunktion darstellen, und $p \in \mathbb{R}$.

Betrachten wir die Produktbildung rekursiv:

$x_{n+1} + i y_{n+1} = (x_n + i y_n) \cdot \text{Produktterm}(n)$, dann bekommen wir:

$$x_{n+1} = \frac{\sqrt{p}(x_n(-\cos(z_n \cdot \ln(p))) + y_n \sin(z_n \cdot \ln(p)) + x_n \sqrt{p})}{-2\sqrt{p} \cdot \cos(z_n \cdot \ln(p)) + p + 1}$$

$$y_{n+1} = \frac{\sqrt{p}(-x_n \sin(z_n \cdot \ln(p)) + y_n(-\cos(z_n \cdot \ln(p)))) + y_n \sqrt{p}}{-2\sqrt{p} \cdot \cos(z_n \cdot \ln(p)) + p + 1}$$

Lassen wir hier auch wie oben die *Cos*-Terme weg, dann erhalten wir wieder die schneller konvergierende Iteration:

$$\begin{aligned} x_{n+1} &= \frac{x_n + \frac{y_n \sin(z_n \ln(p))}{\sqrt{p}}}{-\frac{2\cos(z_n \cdot \ln(p))}{\sqrt{p}} + \frac{1}{p} + 1} \\ y_{n+1} &= \frac{y_n - \frac{x_n \sin(z_n \cdot \ln(p))}{\sqrt{p}}}{-\frac{2\cos(z_n \cdot \ln(p))}{\sqrt{p}} + \frac{1}{p} + 1} \end{aligned} \quad (64)$$

Der Graph der ‚konvergenzbeschleunigten‘ Funktion sieht wie folgt aus:

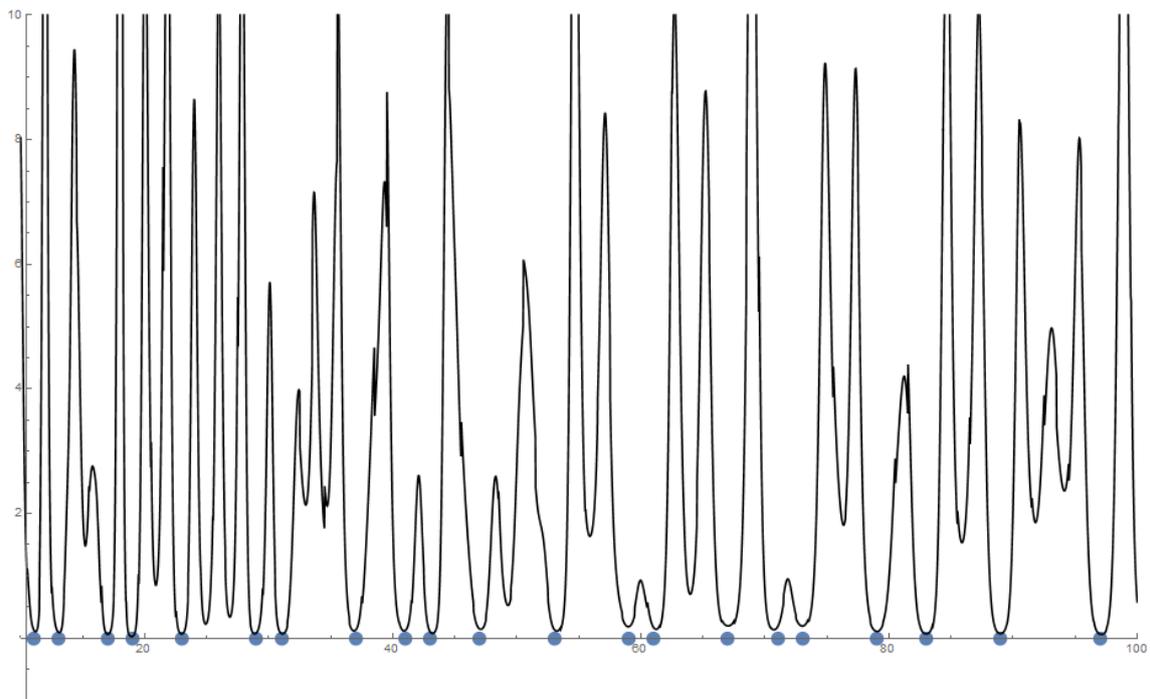


Abbildung 36 Graph mit absoluten Minima bei Primzahlen mit $x_0 = 1, y_0 = 0$

Das Mathematica-Programm, mit dem Graph erzeugt wurde findet sich im Anhang (Riemannsche Zeta-Funktion).

5.5 EINE ZÄHL-FUNKTION FÜR DIE ANZAHL DER NULLSTELLEN

Für die Anzahl der Primzahlen bis zu einer vorgegebenen Grenze N gibt es asymptotische und exakte Formeln, z.B. (133):

$$\pi(x) = R(x) - \sum_{\rho} R(x^{\rho})$$

Hierbei läuft die Summe über alle nicht-trivialen Nullstellen der Zeta-Funktion. Wir haben die heuristische Vermutung, dass sich die Anzahl der Zeta-Nullstellen bis zu einer Grenze auf ähnliche Weise durch eine unendliche Summe (diesmal über einen Primzahl-Term) darstellen lässt.

Die bekannte asymptotische Näherung lautet:

$$N(t) = \frac{t}{2\pi} \left(\ln \frac{t}{2\pi} - 1 \right) \quad (65)$$

Zu Beginn erstmal eine graphische Darstellung der Funktion $N(t)$, die die Anzahl der Nullstellen bis zu einer Obergrenze t angibt (Vergleich mit der exakten Werte):

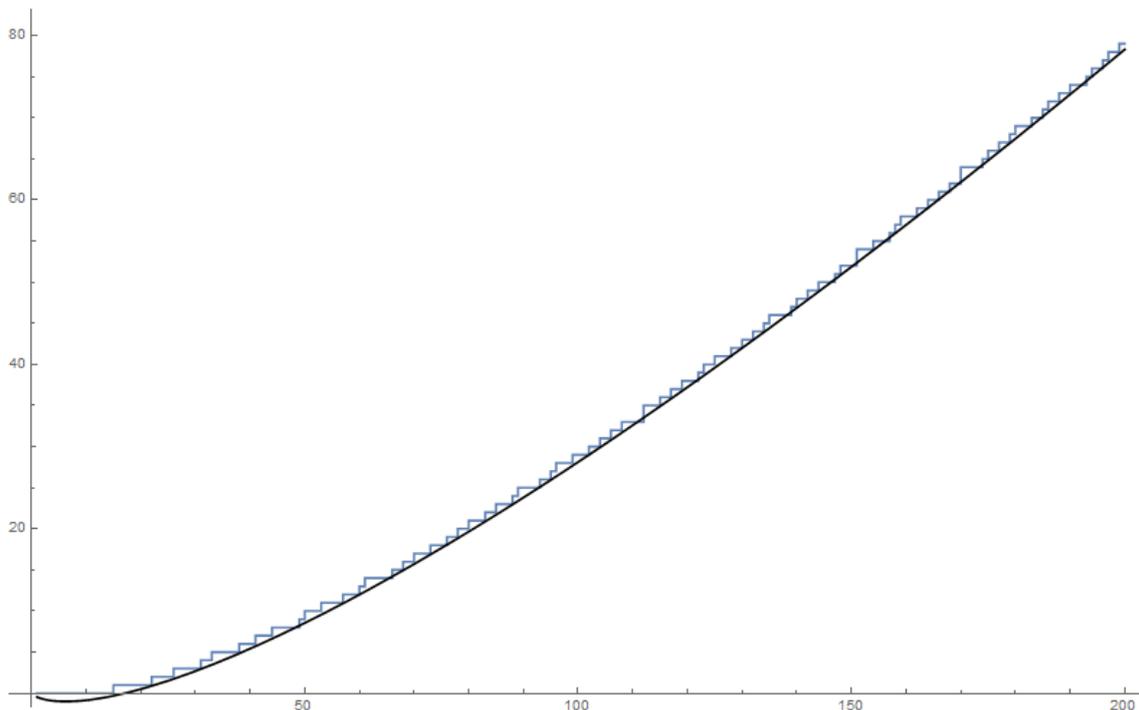


Abbildung 37: Anzahl der Nullstellen der Zetafunktion von 0-200, (exakt und Näherung)

```

Mathematica:
nn=200;temp=Table[0,{nn}];k=1;While[z=Im[ZetaZero[k]];
z<nn,k++;temp[[Ceiling[z];;nn]]++]
NExact[t_]:=temp[[Round[t]]]
NApprox[t_]:=t/(2*Pi)*(Log[t/(2Pi)]-1)
    
```

```
Show[ListLinePlot[Table[NExact[x], {x, 1, nn}], InterpolationOrder->0, ImageSize->Large], Plot[NApprox[x], {x, 1, nn}, PlotStyle->Black, ImageSize->Large]]
```

Es gibt in der Literatur jedoch auch exakte Formeln (Herleitung s. oben):

$$N(t) = \frac{1}{\pi} \operatorname{Im}(\ln(\Gamma(\frac{1}{4} + \frac{it}{2}))) - \frac{t}{2\pi} \ln(\pi) + \frac{1}{\pi} \operatorname{Im}(\ln(\zeta(\frac{1}{2} + it))) + 1 \quad (66)$$

Als Graph:

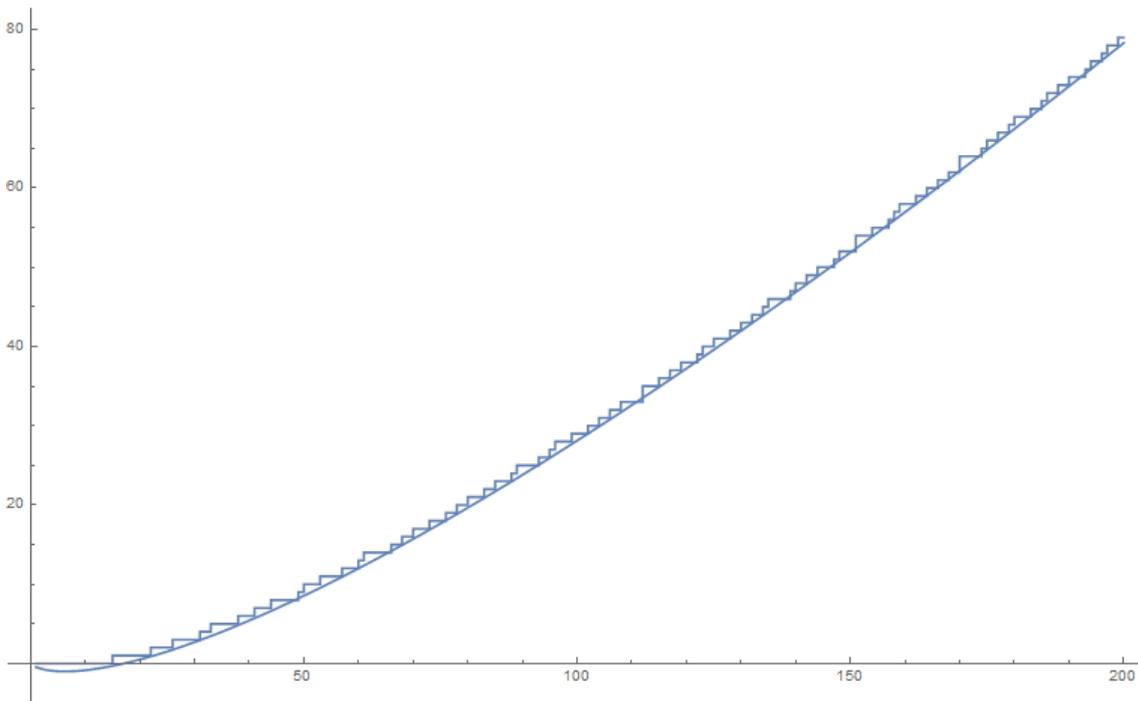


Abbildung 38: Anzahl der Nullstellen der Zetafunktion von 0-200, (Formel (66) und Näherung)

```
Mathematica:
countZero[t_]:=1/Pi*Im[LogGamma[1/4+I*t/2]]-
t/(2*Pi)Log[Pi]+1/Pi*Im[Log[Zeta[1/2+I*t]]]+1
NApprox[t_]:=t/(2*Pi)*(Log[t/(2Pi)]-1)
Show[ListLinePlot[Table[NApprox[n], {n, 1, 200}],
PlotRange->All], ListLinePlot[Table[countZero[n], {n, 1, 200}],
InterpolationOrder->0, PlotRange->All]]
```

Formel (66) scheint die Anzahl der Nullstellen exakt wiederzugeben. (Überprüft vom Autor bis 100000).

Die Dualität zwischen Primzahlen und Nullstellen der Zetafunktion ist ein wesentliches Thema dieses Buches. Die Versuchung ist natürlich sehr groß, eine Darstellung für die Anzahl der Nullstellen zu finden, die aus einem Näherungsterm besteht (der die asymptotische Entwicklung beschreibt), so wie einem zusätzlichen Summenterm, der umso mehr Details und lokale Feinheiten berücksichtigt, je mehr Terme wir mitnehmen.

Eine Zähl-Funktion für die Anzahl der Nullstellen

Damit wäre die Dualität zwischen Primzahlen und Nullstellen der Zeta-Funktion perfekt. Wir denken da z. B. an Formel (133), die die exakte Anzahl der Primzahlen liefert:

$$\pi(x) = R(x) - \frac{1}{\ln(x)} + \frac{1}{\pi} \arctan\left(\frac{\pi}{\ln(x)}\right) + \sum_{\rho} R(x^{\rho})$$

Ersetzen wir in (66) den Zeta-Term durch die Produktdarstellung mit Primzahlen, dann haben wir so eine Formel gefunden:

$$N_c(t) = \frac{1}{\pi} \ln\left(\Gamma\left(\frac{1}{4} + \frac{it}{2}\right)\right) - \frac{it}{2\pi} \ln \pi - \frac{1}{\pi} \sum_n \ln\left(1 - p_n^{-\frac{1}{2}-it}\right) \quad (67)$$

$$N(t) = \text{Im } N_c(t)$$

Hier ein Plot dieser Funktion von $t = 0$ bis 60 (mit den ersten 1000 Primzahlen beim Summenterm, der asymptotische Teil in blau):

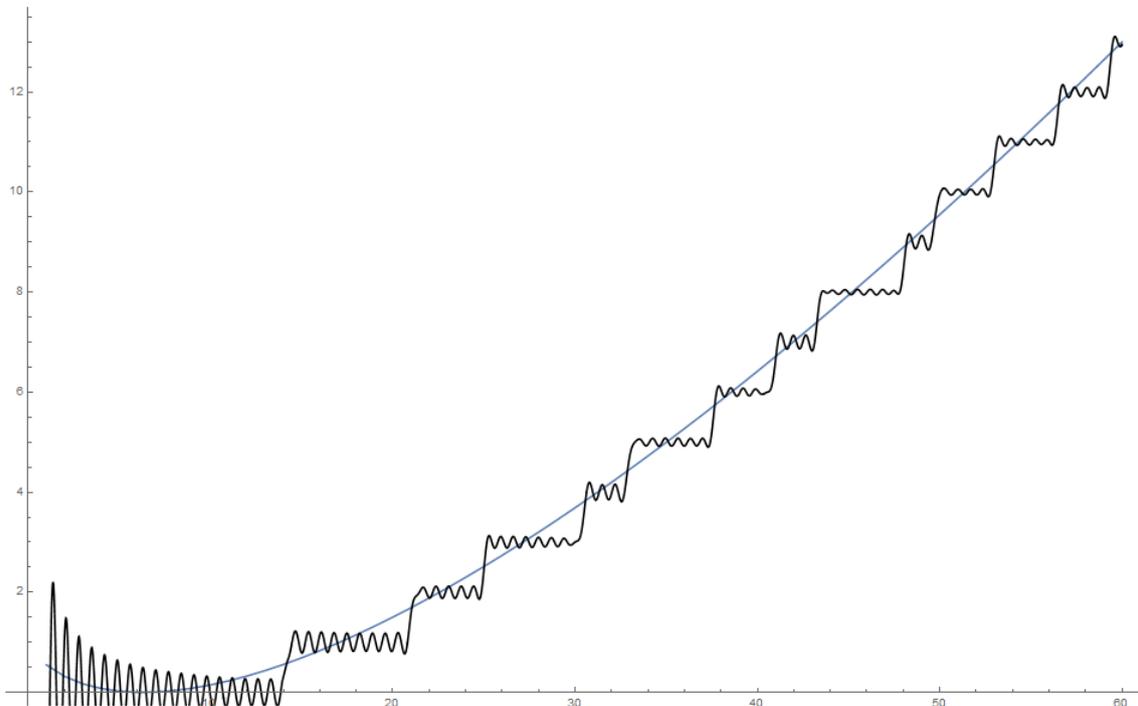


Abbildung 39: Nullstellen-Zählfunktion der Zetafunktion mit Primzahl-Summenterm

Mathematica:

```
NApprox[t_]:=t/(2*Pi)*(Log[t/(2*Pi)]-1)+1
countZeroComplex[t_]:=1/Pi*LogGamma[1/4+I*t/2]-I*t/(2*Pi)Log[Pi]-
1/Pi*Sum[Log[1-Prime[n]^(-1/2-I*t)],{n,1,1000}]+I
Show[ListLinePlot[Table[NApprox[n],{n,1,60}],PlotRange->All],
Plot[Im[countZeroComplex[n]},{n,1,60},PlotStyle->Black,PlotRange->All]]
```

Hinweis: Leider konvergiert diese Darstellung nicht absolut. Die Amplituden der Schwingungen werden umso größer, je mehr Primzahl-Terme hinzugefügt werden.

5.6 DIE ZETA-FUNKTION UND DAS QUANTENCHAOS: EINE BRÜCKE ZUR PHYSIK

Gelegentlich hilft der Zufall etwas bei mathematischen Entdeckungen nach. So geschah dies vermutlich irgendwann in den 70-er Jahren des letzten Jahrhunderts. Durch Zufall trafen sich der Mathematiker H. Montgomery und der Physiker F. Dyson und erzählten sich beiläufig von ihren jeweiligen aktuellen Forschungsprojekten, vermutlich wurden auch ein paar Blicke auf Diagramme bzw. Formeln geworfen.

Montgomery war Zahlentheoretiker und Spezialist auf dem Gebiet der Zeta-Funktion und untersuchte den Zusammenhang zwischen den komplexen Nullstellen und Primzahlen. Dyson war einer der führenden Kern-Physiker auf dem Gebiet der sogenannten Zufalls-Matrizen (einem mathematischen Spezialgebiet, das bei der Behandlung der Eigenschaften von schweren Atomkernen Anwendung findet).

Vielleicht hat Montgomery Dyson einen Zettel mit einem Diagramm der Lage der Nullstellen gezeigt. Dyson erkannte, dass zwischen der Verteilung der Nullstellen der Zetafunktion und gewissen physikalischen Spektren eine frappierende Ähnlichkeit besteht. Physikalisch beschreiben diese Spektren Energiezustände in schweren Atomkernen, mathematisch berechnen sich solche Spektren aus den Eigenwerten von sogenannten ‚Gaußschen Zufallsmatrizen‘ (das sind Matrizen, die mit Zufallswerten besetzt sind, die einer Gaußschen Normalverteilung entsprechen). In der Sprache der Physiker ausgedrückt, bedeutet dies, dass die Spektralwerte die ‚Eigenwerte‘ eines ‚Hermiteschen‘ Operators sind.³⁶ Hermitesche (auch ‚selbst-adjungiert‘ bezeichnet) Operatoren spielen in der Quantenmechanik eine große Rolle. Sie besitzen aufgrund ihrer Symmetrieeigenschaften stets reelle Eigenwerte.

Die Vermutung lautet nun, dass die komplexen Nullstellen der Zeta-Funktion nichts anderes als die (reellen) Eigenwerte eines geheimnisvollen hermiteschen Operators sind. Diese Hypothese wird in der Literatur als die ‚GUE‘-Hypothese bezeichnet (‚GUE‘: Gaussian Unitary Ensemble). Leider wurde dieser Operator bisher noch nicht gefunden. Es gibt jedoch überwältigende numerische Hinweise, dass ein solcher Operator tatsächlich existiert (s. unten).

Eigentlich geht diese Vermutung zeitlich noch viel weiter zurück: Der ungarische Mathematiker George Pólya äußerte diese Vermutung vor über 100 Jahren (Hilbert-Pólya-Vermutung).³⁷

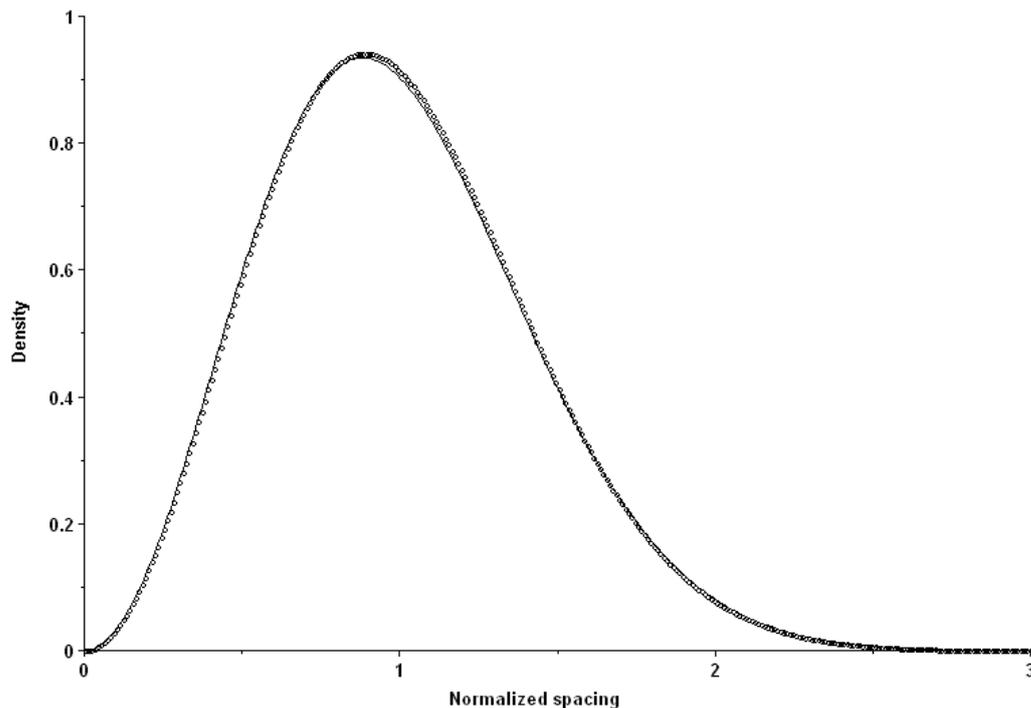
Betrachten wir die auftretenden Abstände der komplexen Nullstellen z_n entlang der kritischen Geraden und normieren diese Abstände:

$$\delta_n = \frac{z_{n+1} - z_n}{2\pi} \ln \frac{z_n}{2\pi} \quad (68)$$

Wir wissen sicher aus der Theorie, dass die δ_n den Mittelwert 1 haben. Der kleinste bisher bekannte Wert von δ_n liegt bei 1034741742903.353 (das ist die

³⁶ https://de.wikipedia.org/wiki/Hermitescher_Operator

³⁷ https://en.wikipedia.org/wiki/Hilbert-Pólya_conjecture



4.088.664.936.217-te Nullstelle!) und hat einen normalisierten Wert von 0.00007025. Dies entspricht einem tatsächlichen Abstand von 0.00001709! (Stand: Jan. 2016)

Diese kleinen Nullstellenabstände sind jedoch sehr selten. Im Allgemeinen zeigen die Nullstellen eher ein ‚abstoßendes‘ Verhalten und vermeiden es, sich zu nahe zu kommen. Dieses Verhalten kennt man von den Energieniveaus quantenmechanischer Systeme.

Hier ein Vergleich der statistischen Verteilung der Vorhersage aus der Theorie der Zufallsmatrizen (durchgezogene Linie) und der tatsächlichen Werte der Nullstellen der Zeta-Funktion. Für die Berechnung der statistischen Verteilung wurden 2 Milliarden Nullstellen im Bereich bis 10^{13} ausgewertet. Die Grafik wurde entnommen aus:

<http://numbers.computation.free.fr/Constants/Miscellaneous/zetazeroscompute.html>

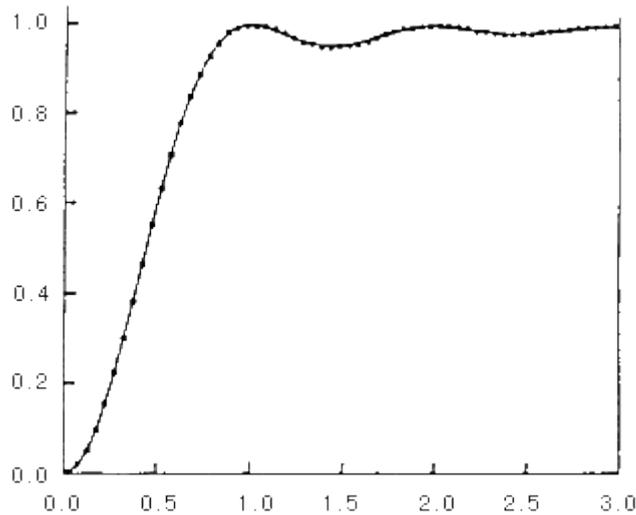
Die meisten Ergebnisse und Berechnungen haben wir Andrew Odlyzko zu verdanken:
<http://www.dtc.umn.edu/~odlyzko/>

Wie man sieht, sind die Hinweise darauf, dass den Nullstellen der Zeta-Funktion tatsächlich ein (noch unbekannter) Operator zugrunde liegt, offensichtlich.

Weitergehende Informationen über dieses aktuelle Forschungsgebiet findet man unter:

http://www.dartmouth.edu/~chance/chance_news/recent_news/primes_part3/part3.html

Auch die Paar-Korrelation zwischen zwei beliebigen Nullstellen scheint der theoretischen Vorhersage aus der GUE-Theorie zu folgen:



Es gibt jedoch auch noch Unklarheiten über die Natur dieses unbekanntes Operators. Untersuchungen von Odlyzko (Fourieranalysen der kritischen Nullstellen) zeigen ein etwas anderes Verhalten als bei physikalischen GUE-Eigenwerten (z. B Peaks bei Primzahl-Potenzen) zu erwarten ist. Es gibt daher auch die Vermutung, dass der zugrundeliegende Operator nicht von den Eigenwerten eines GUE-Operators kommt, sondern von den Eigenwerten eines allgemeineren chaotischen Systems. Der interessierte Leser möge im Internet unter den Stichworten „quantum chaology“ recherchieren.

Eine relativ unbekannt Methode stellt eine Verbindung zwischen Quanten-Oszillatoren und der Zeta-Funktion her. Ohne zu sehr ins Detail zu gehen sei hier die Methode von Crandall (Richard Crandall, 2001) beschrieben:

Es gibt eine durch die Schrödinger-Gleichung beschriebene zeitliche Entwicklung einer ‚glatten‘ Wellenfunktion $\psi(x, t)$, von der wir wissen, dass sie zum Zeitpunkt $t = 0$ keine Nullstellen auf der X-Achse hat. Nach einer Zeit t , in der sich die Wellenfunktion gemäß der Schrödingergleichung zeitlich entwickelt, wird diese Wellenfunktion jedoch ‚verrauscht‘ und ‚zappelig‘ und bekommt unendlich viele Nullstellen auf der X-Achse, die mit den kritischen Nullstellen der Zeta-Funktion identisch sind. Diese Wellenfunktion kann wie folgt dargestellt werden:

$$\psi(x, t) = f\left(\frac{1}{2} + ix\right) \zeta\left(\frac{1}{2} + ix\right) = e^{\frac{x^2}{2a^2}} \sum_{n=0}^{\infty} c_n (-1)^n H_{2n}\left(\frac{x}{a}\right) \quad (69)$$

Wobei a reell ist und c_n noch zu bestimmende (von a abhängige) Konstanten sind. H_n ist dabei das Hermitesche Polynom der n -ten Ordnung. $f(s)$ sei eine analytische Funktion, die keine Nullstellen besitzt. Beschränken wir die unendliche Summe auf endlich viele Terme, so kann man mit numerischen Methoden daraus die endlich viele Nullstellen berechnen. Borwein (Borwein, 2000) konnte mit dieser Methode unter Verwendung der ersten 27 Summen-Terme die ersten sieben kritischen Nullstellen der Zeta-Funktion mit einer Genauigkeit von 10 Dezimalstellen berechnen! Im Prinzip lassen sich mit dieser Methode alle kritischen Nullstellen berechnen. Die Berechnung läuft auf die Berechnung der Eigenwerte einer Hessenberg-Matrix hinaus, was wiederum letztlich auf die Berechnung der Nullstellen eines charakteristischen Polynoms hinausläuft.

Die Zeta-Funktion und das Quantenchaos: Eine Brücke zur Physik

Hier tun sich weitere, höchst interessante Zusammenhänge auf, die in Bezug zur Riemanschen Vermutung stehen.

6 EINSCHUB: DIE RIEMANN-FUNKTION $R(s)$

Die Riemann-Funktion $R(x)$ (nicht zu verwechseln mit der Riemannschen Zeta-Funktion $\zeta(s)$) spielt eine wichtige Rolle in der Theorie der Primzahlen. Sie liefert die beste einfache Näherung zur Berechnung von $\pi(x)$, der Anzahl von Primzahlen bis zur Grenze x . Im Folgenden seien die wichtigsten Eigenschaften und Berechnungsmethoden kurz vorgestellt. Die am häufigsten benutzten Darstellungen sind die Summation mit Hilfe der Möbius-Funktion $\mu(n)$ und dem Integrallogarithmus $\text{li}(x)$,

$$R(x) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} \text{li}\left(\frac{x}{n}\right) \quad \text{für } x > 1 \quad (70)$$

sowie die (sehr schnell konvergierende) Summation mit Hilfe von Potenzen von $\ln(x)$ und Werten von $\zeta(n)$ mit ganzzahligen Argumenten n , die in der Literatur allgemein als ‚Gram‘-Funktion oder –Reihenentwicklung bezeichnet wird:

$$R(x) = 1 + \sum_{n=1}^{\infty} \frac{(\ln x)^n}{n! n \zeta(n+1)} \quad \text{für } x > 0 \quad (71)$$

Der folgende Plot zeigt, wie gut die Riemann-Funktion $R(x)$ die Funktion $\pi(x)$ approximiert (siehe auch TABELLE 29):

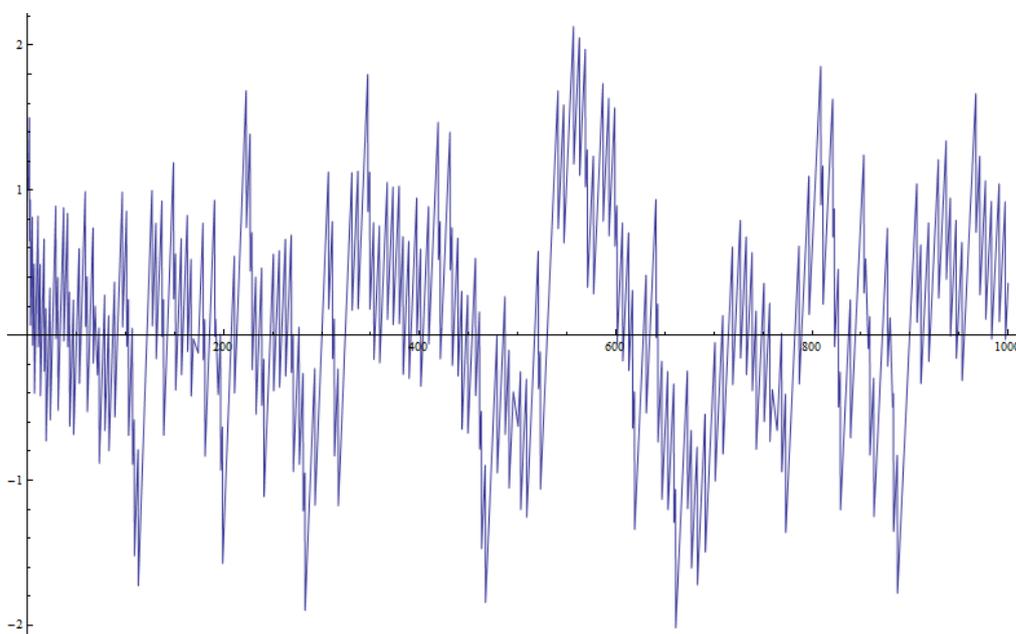


Abbildung 40: $R(x) - \pi(x)$, für Werte von $x=1$ bis 1000

7 EIN PAAR WICHTIGE ZAHLENTHEORETISCHE FUNKTIONEN

7.1 DIE OMEGA-FUNKTIONEN: ANZAHL DER PRIMFAKTOREN

$\omega(n)$ bezeichnet die Anzahl unterschiedlicher Primfaktoren einer positiven ganzen Zahl n . $\omega(n)$ wird somit klar durch die Faktorisierung einer ganzen Zahl als:

$$n = p_1^{e_1} p_2^{e_2} \dots p_{\omega(n)}^{e_{\omega(n)}}$$

Im Gegensatz dazu bezeichnet $\Omega(n)$ die totale Anzahl von Primfaktoren einer ganzen Zahl n :

$$\Omega(n) = \sum_{k=1}^{e_{\omega(n)}} e_k \quad (72)$$

Anschaulich ist $\Omega(n)$ einfach die Summe der Primzahlexponenten von n .

In Mathematica ist $\omega(n)$ durch `PrimeNu[n]` und $\Omega(n)$ durch `PrimeOmega[n]` definiert.

Zahlen, die nur aus verschiedenen Primfaktoren bestehen, sind identisch mit den quadratfreien Zahlen.

Das asymptotische Verhalten von $\omega(n)$ ist gegeben durch:

$$\omega(n) \sim \ln \ln n + B_1 + \sum_{k=1}^{\infty} \left(-1 + \sum_{j=0}^{k-1} \frac{\gamma_j}{j!} \right) \frac{(k-1)!}{(\ln n)^k}$$

wobei B_1 die Mertens-Konstante ist und γ_j die Stieltjes-Konstanten.

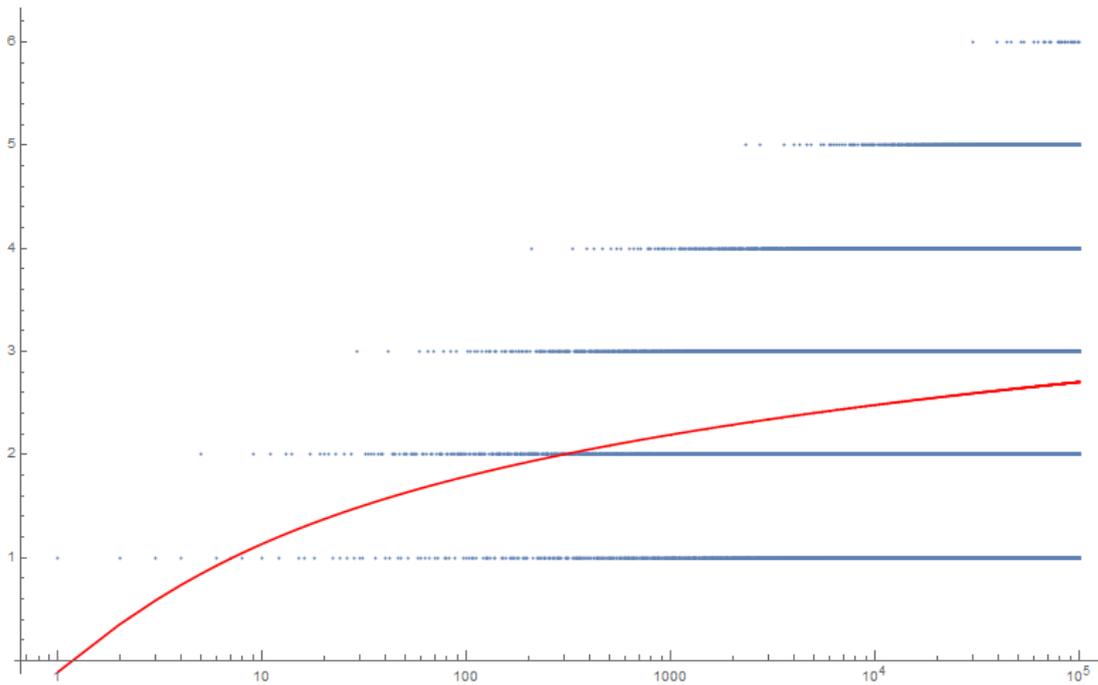


Abbildung 41: Funktion $\omega(n)$, Anzahl verschiedener Primfaktoren (rot: asymptotisch)

Mathematica:

```
mertensB1=0.2614972128;
```

```
nmax=100000;
```

```
Show[ListLogLinearPlot[Table[PrimeNu[n], {n, 2, nmax}],
```

```
PlotRange->All], ListLogLinearPlot[Table[Log[Log[n]]+mertensB1,
```

```
{n, 2, nmax}], PlotRange->All, Joined->True, PlotStyle->Red]]
```

Das asymptotische Verhalten von $\Omega(n)$ ist ebenfalls näherungsweise:

$$\Omega(n) \sim \ln \ln n + B_2, \text{ mit } B_2 = 0.494906$$

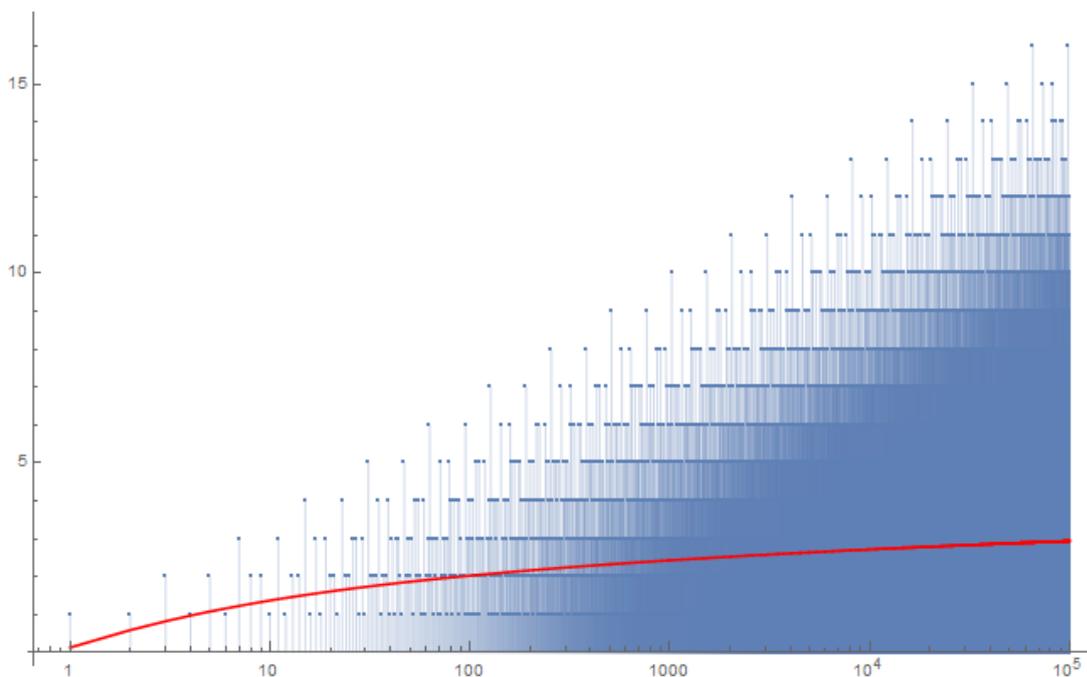


Abbildung 42: Funktion $\Omega(n)$, totale Anzahl Primfaktoren (rot: asymptotisch)

Die Liouville-Funktion

```
Mathematica:  
mertensB2=0.494906;  
nmax=100000;  
Show[ListLogLinearPlot[Table[PrimeOmega[n], {n, 2, nmax}],  
PlotRange->All, Filling->Axis], ListLogLinearPlot[Table[Log[Log[n]] +  
mertensB2, {n, 2, nmax}], PlotRange->All, Joined->True, PlotStyle->Red]]
```

Hinweis: $\Omega(n)$ steht in engem Zusammenhang mit dem ‚Gradus Suavitatis‘ von Leonhard Euler (siehe Kapitel 15.1)

Es gelten folgende Beziehungen zu anderen zahlentheoretischen Funktionen:

Liouville's Funktion:

$$\lambda(n) = (-1)^{\Omega(n)}$$

Statt die Anzahl aller Primfaktoren ($\Omega(n)$) bzw. die Anzahl unterschiedlicher Primzahlen $\omega(n)$ der Primfaktorenzerlegung einer Zahl zu berechnen, kann man auch die Summe aller Primfaktoren ($\text{sopfr}(n)$) bzw. die Summe aller unterschiedlichen Primzahlen ($\text{sopf}(n)$) berechnen. Die erstere wird auch als ‚ganzzahliger Logarithmus‘ (engl.: Integer Logarithm) bezeichnet.

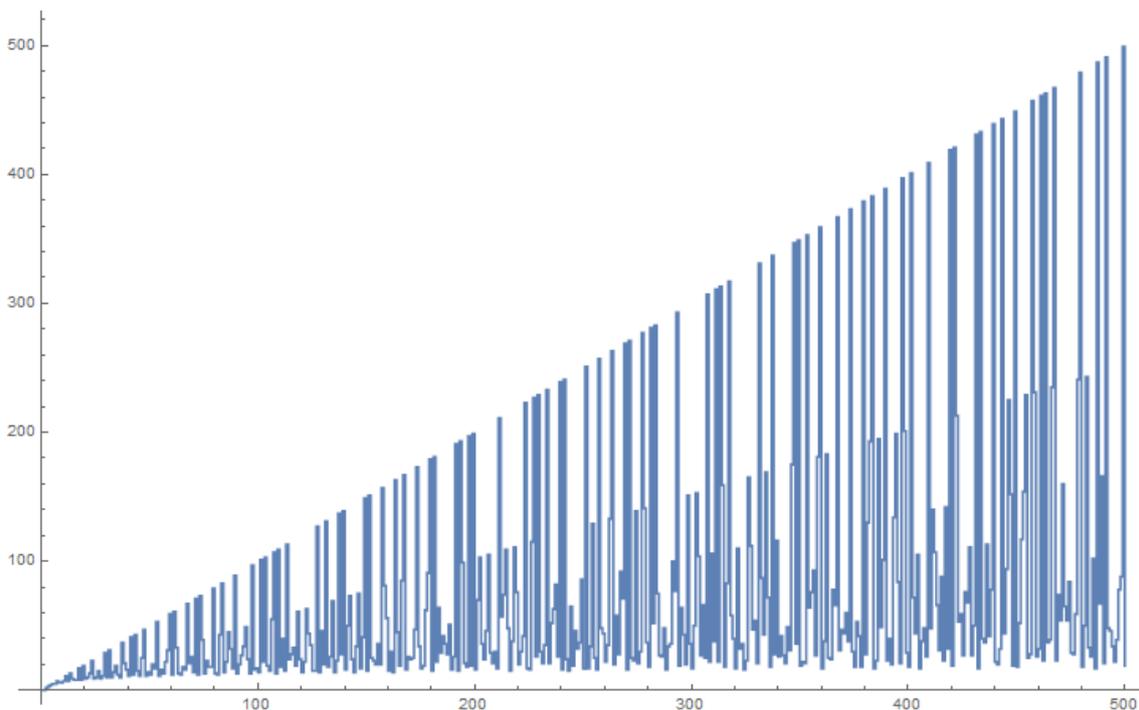


Abbildung 43: Integer-Logarithmus: Summe aller Primzahlen der Zerlegung für n:sopfr(n)

```
Mathematica:  
f[n_] := Plus@@Times@@@FactorInteger@n; f[1]=0;  
ListLinePlot[Table[f[n], {n, 1, 500}], InterpolationOrder->0, PlotRange->All]
```

7.2 DIE LIOUVILLE-FUNKTION

Die Liouville-Funktionen λ und L sind definiert als:

$$\lambda(n) = (-1)^{\Omega(n)}, \quad L(n) = \sum_{k=1}^n \lambda(k) \quad (73)$$

$\lambda(n)$ ist -1 , wenn n eine ungerade Anzahl von Primfaktoren besitzt und $+1$ wenn n eine gerade Anzahl besitzt.

$\lambda(n)$ hängt mit der Riemannschen ζ -Funktion zusammen:

$$\frac{\zeta(2s)}{\zeta(s)} = \sum_{k=1}^{\infty} \frac{\lambda(k)}{k^s} \quad (74)$$

$L(n)$ ist die summatorische Funktion über $\lambda(n)$.

Für $L(n)$ existieren folgende Formeln:³⁸

$$L(x) = \sum_{m=1}^{\frac{x}{w}} \mu(m) \left\{ \left\lfloor \sqrt{\frac{x}{m}} \right\rfloor - \sum_{k=1}^{v-1} \lambda(k) \left(\left\lfloor \frac{x}{km} \right\rfloor - \left\lfloor \frac{x}{m(v-k)} \right\rfloor \right) \right\} - \sum_{l=\frac{x}{w}-1}^{\frac{x}{v}} L\left(\frac{x}{l}\right) \sum_{m=1}^{\frac{x}{w}} \mu(m) \quad (75)$$

Hier ein Graph von $L(x)$:

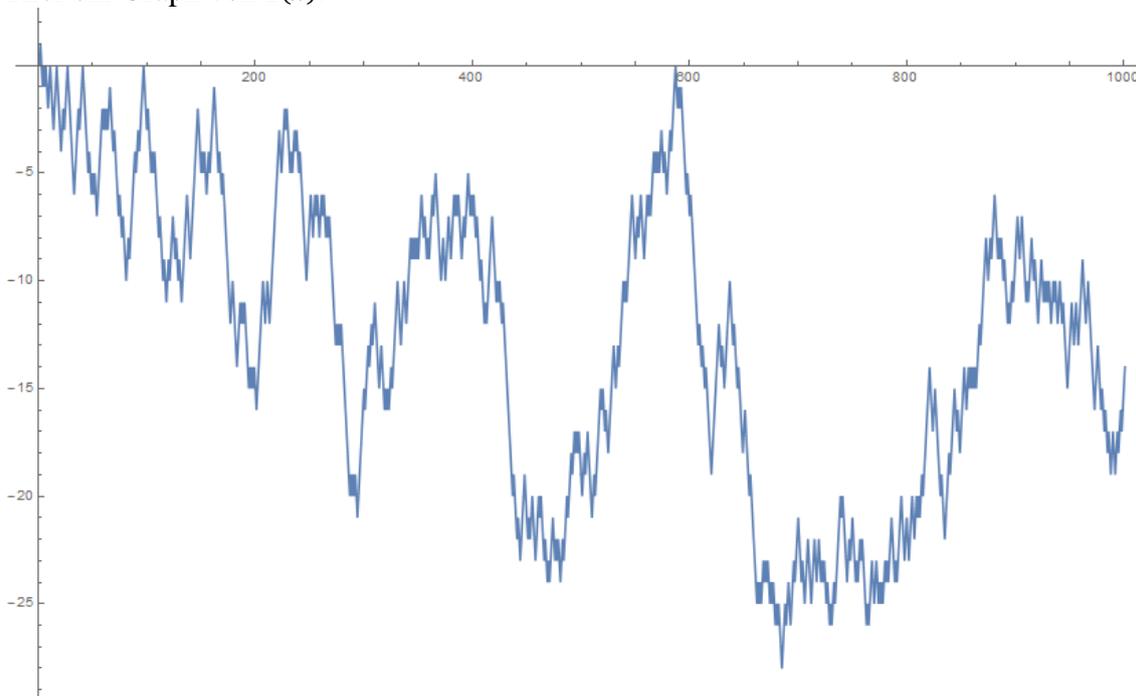


Abbildung 44: Liouville Lambda-Funktion, von 1 bis 1000

Mathematica:

```
lTab=Accumulate[Join[{0}, LiouvilleLambda[Range[1000]]]];
ListLinePlot[lTab]
```

$L(x)$ hat eine starke Tendenz, negativ zu sein. Man vermutete bis in die 50er Jahre des letzten Jahrhunderts, dass $L(x) \leq 0$ immer gilt. Tatsächlich wurde jedoch 1962 das erste Gegenbeispiel gefunden: $L(906180359) = 1$. Das kleinste Gegenbeispiel ist $L(906150257)$. Es ist noch ungeklärt, ob es nur endlich viele Gegenbeispiele oder unendlich viele davon gibt (Stand: Jan. 2016)

³⁸ <http://mathworld.wolfram.com/LiouvilleFunction.html>

Die Tschebyschew-Funktion

Wie viele zahlentheoretische Funktionen kann auch $L(x)$ analytisch mit Hilfe einer Summe über die komplexen Nullstellen der Zeta-Funktion sehr gut approximiert werden (hier über die ersten N Nullstellen):

$$L(x) = 1 + \frac{\sqrt{x}}{\zeta\left(\frac{1}{2}\right)} + 2\operatorname{Re}\left(\sum_{k=1}^N \frac{x^{\rho_k} \zeta(2\rho_k)}{\rho_k \zeta'(\rho_k)}\right) \quad (76)$$

Mathematica-Programm: Bitte den Autor kontaktieren.

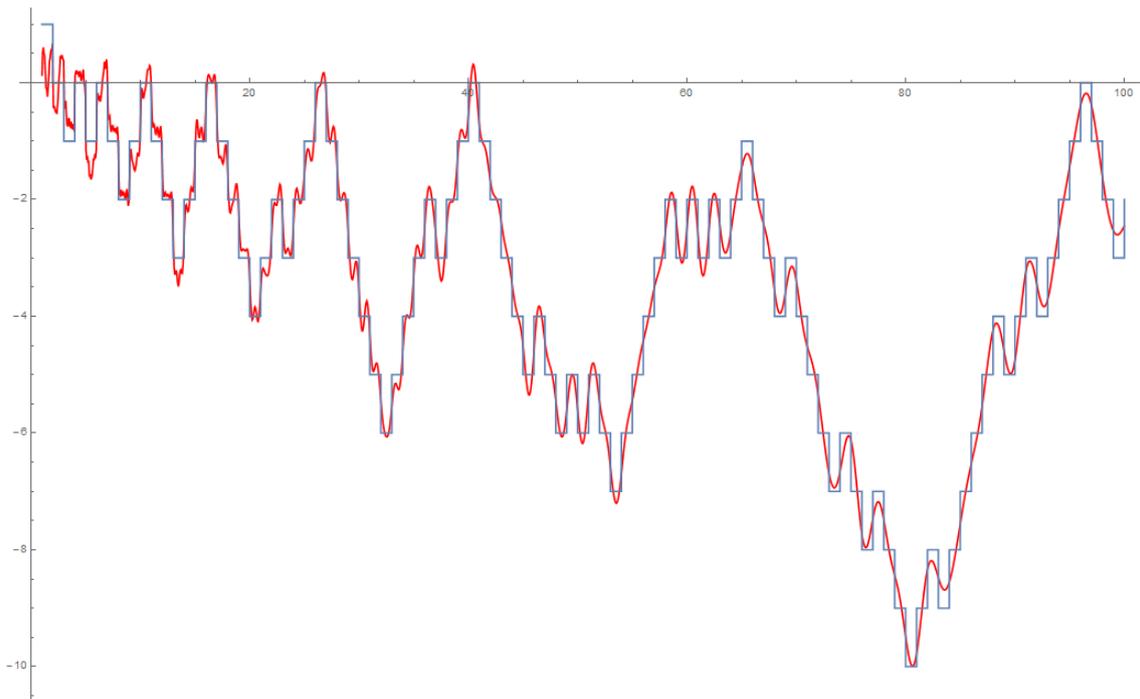


Abbildung 45: Liouville-Funktion, von 1 bis 100: Exakt und analytisch berechnet

7.3 DIE TSCHEBYSCHEW-FUNKTION

Die Summenfunktion der Mangoldt-Funktion $\Lambda(n)$ nennt man Funktion Tschebyschew-Funktion der 2.ten Art (Psi-Funktion):

$$\psi(x) = \sum_{p^k \leq x} \ln(p) = \sum_{n \leq x} \Lambda(n)$$

Mathematica:

```
ListLinePlot[Table[{n, Sum[MangoldtLambda[k], {k, 1, n}]}, {n, 1, 100}], InterpolationOrder->0]
```

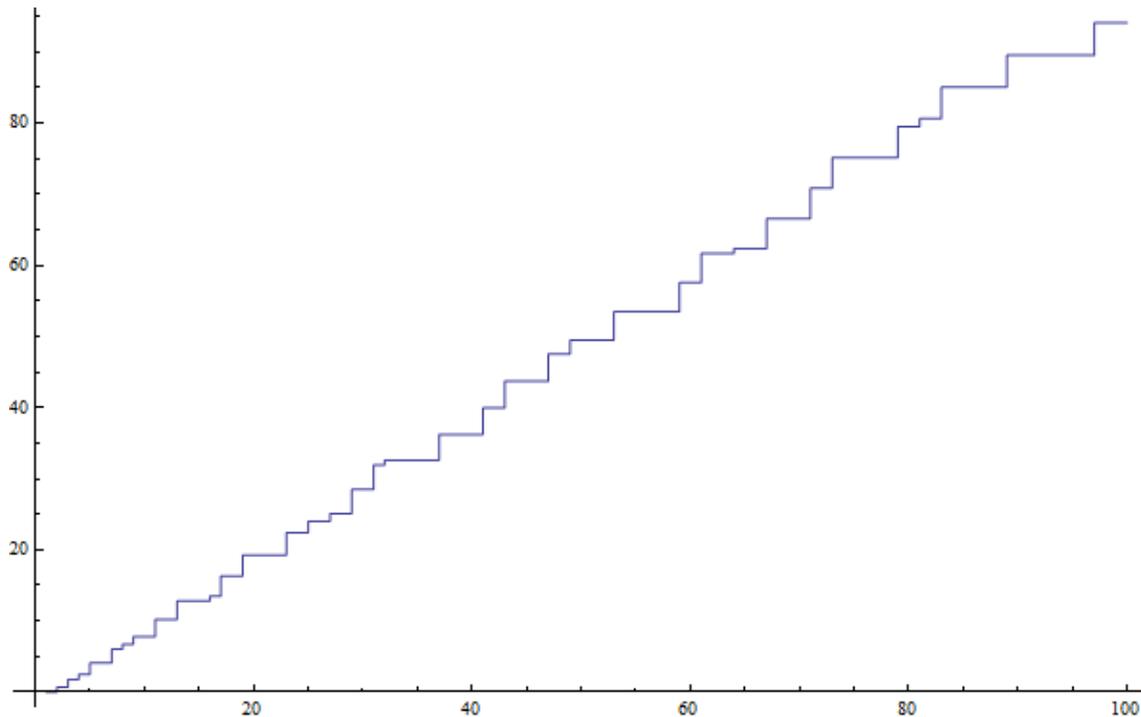


Abbildung 46: Tschebyschew Psi Funktion, dargestellt von 0 bis 100

Die Tschebyschew Psi-Funktion kann exakt analytisch als explizite Funktion geschrieben werden:

$$\psi_0(x) = x - \sum_{\rho} \frac{x^{\rho}}{\rho} - \ln(2\pi) - \frac{1}{2} \ln(1 - x^{-2}) \quad (77)$$

Dabei läuft die Summation über die nicht-trivialen Nullstellen der Zeta-Funktion (hier über die ersten 50 Nullstellenpaare).

Mathematica:

$$\text{myPsi}(x, y) := - \sum_{i=-50}^{50} \text{If}[i \neq 0, \frac{(x + iy)^{\rho_i}}{\rho_i}, 0] - 0.5 \log\left(1 - \frac{1}{x^2}\right) + x - \log(2\pi)$$

`Plot[Re[myPsi[x, 0]], {x, 1, 100}] (*Realteil*)`

Die Tschebyschew-Funktion

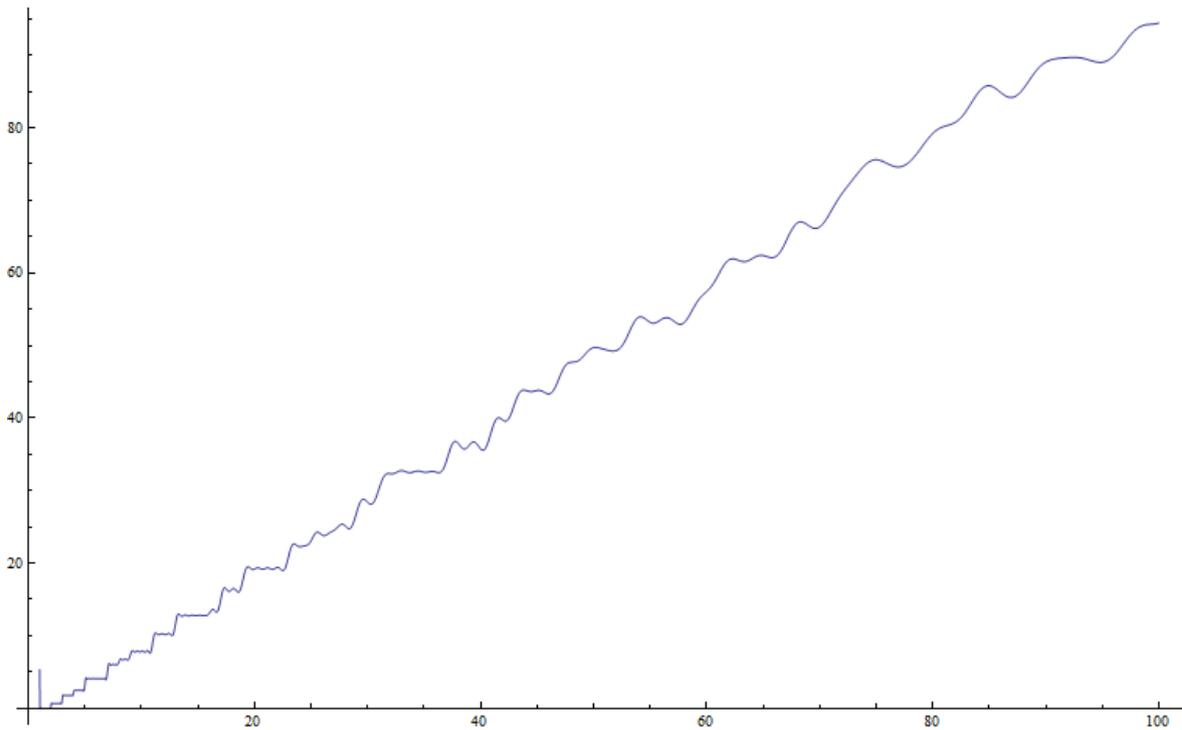


Abbildung 47: Tschebyschew Psi-Funktion analytisch berechnet, dargestellt von 0 bis 100

Beide Funktionen im Vergleich (Summe über die ersten 75 Nullstellenpaare):

Mathematica-Programm: Bitte den Autor kontaktieren.

```
Show[Plot[Re[myPsi[x, 0]], {x, 1, 100}, PlotPoints->400,  
PlotStyle->Red], ListLinePlot[Table[{n, Sum[MangoldtLambda[k], {k, 1,  
n}]}, {n, 1, 100}], InterpolationOrder->0]]
```

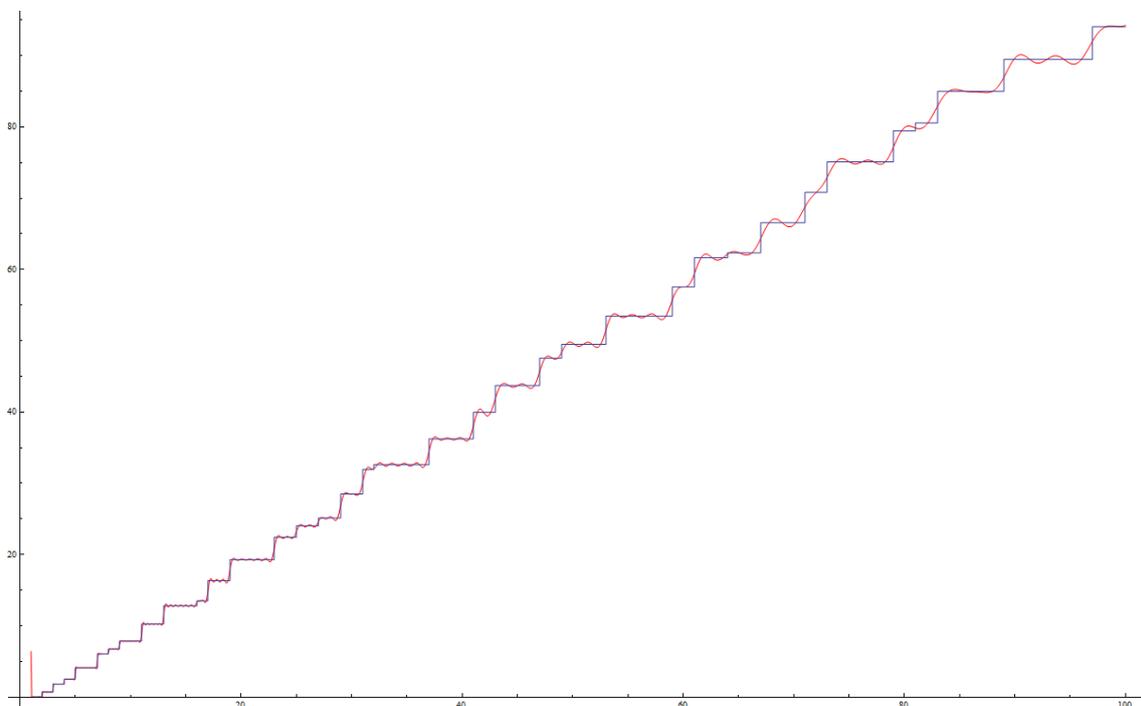


Abbildung 48 Tschebyschew Psi Funktion, Vergleich analytische und zahlentheoretische Berechnungsmethode

7.4 DIE EULERSCHE PHI-FUNKTION

Die Eulersche Phi-Funktion $\varphi(n)$, im Englischen Totient function) gibt an, wieviele zu n teilerfremde Zahlen es gibt, die kleiner oder gleich n sind.

Definition von $\varphi(n)$:

$$\varphi(n) = |\{k \in \mathbb{N} | 1 \leq k \leq n \wedge \text{ggT}(k, n) = 1\}| \quad (78)$$

7.4.1 BERECHNUNG UND GRAPHISCHE DARSTELLUNGEN DER PHI-FUNKTION

Mathematica:

```
ListLinePlot[Table[{n, EulerPhi[n]}, {n, 1, 100}], InterpolationOrder->0]
```

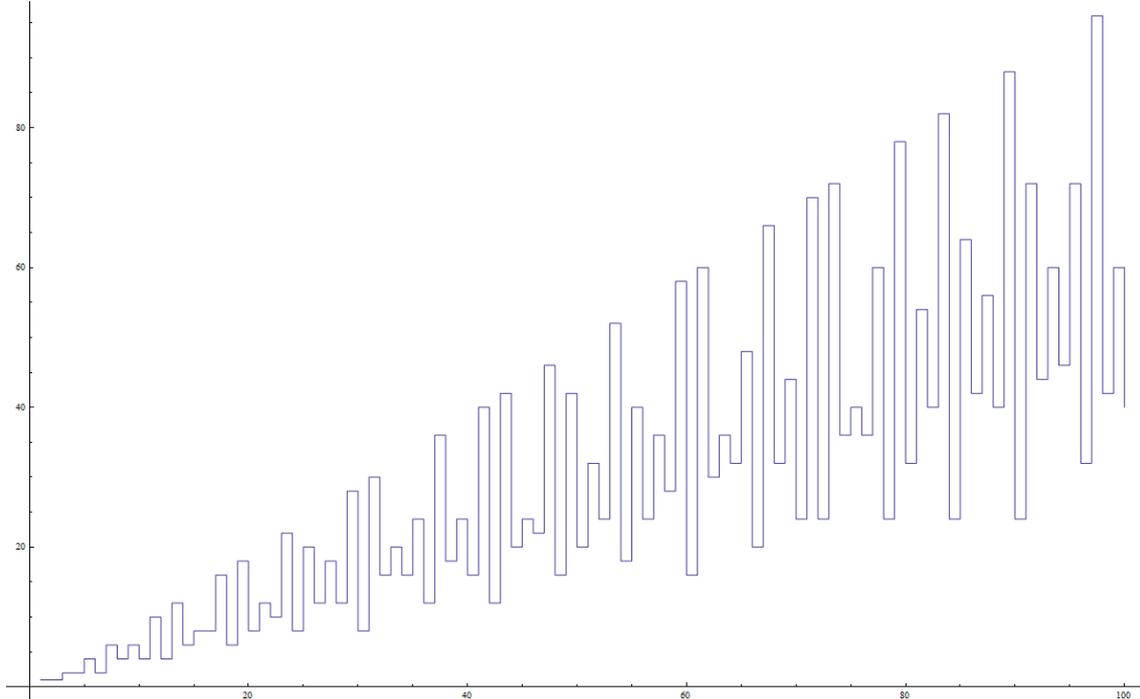


Abbildung 49: Eulersche Phi-Funktion, dargestellt von 1 bis 100

Berechnung von $\varphi(n)$:

(a_i seien die Exponenten der Primfaktorenzerlegung $n = \prod_{i=1}^r p_i^{a_i}$)

$$\varphi(n) = \prod_{p|n} p^{a_i-1}(p-1) = n \prod_{p|n} \left(1 - \frac{1}{p}\right) \quad (79)$$

$$\varphi(n) = n \lim_{s \rightarrow 1} \zeta(s) \sum_{d|n} \mu(d) (e^{\frac{1}{d}})^{(s-1)} \quad (80)$$

Ihre Summationsfunktion $\Phi(n)$ berechnet die Summe bis n :

$$\Phi(n) = \sum_{k=1}^n \varphi(k) \quad (81)$$

Die Eulersche Phi-Funktion

Mathematica:

```
PhiSum[n_] := Sum[EulerPhi[k], {k, 1, n}]
```

Wenden wir Perron's Formel an, bekommt man eine analytische Formel für $\phi(x)$:

$$\Phi(x) \sim \frac{1}{6} + \frac{3x^2}{\pi^2} + \operatorname{Re} \left(\sum_{k=1}^N \frac{x^{\rho_k} \zeta(\rho_k - 1)}{\rho_k \zeta'(\rho_k)} \right) + \sum_{k=1}^N \frac{x^{-2k} \zeta(-2k - 1)}{(-2k) \zeta'(-2k)} \quad (82)$$

Mathematica-Programm: Bitte den Autor kontaktieren.

Weitere nützliche Mathematica-Befehle:

```
DirichletTransform[EulerPhi[n], n, s]
```

Vergleich der analytischen $\Phi(n)$ mit der zahlentheoretischen Funktion $\Phi(x)$ (mit Summation über die ersten 50 nichttrivialen und 50 trivialen Nullstellen):

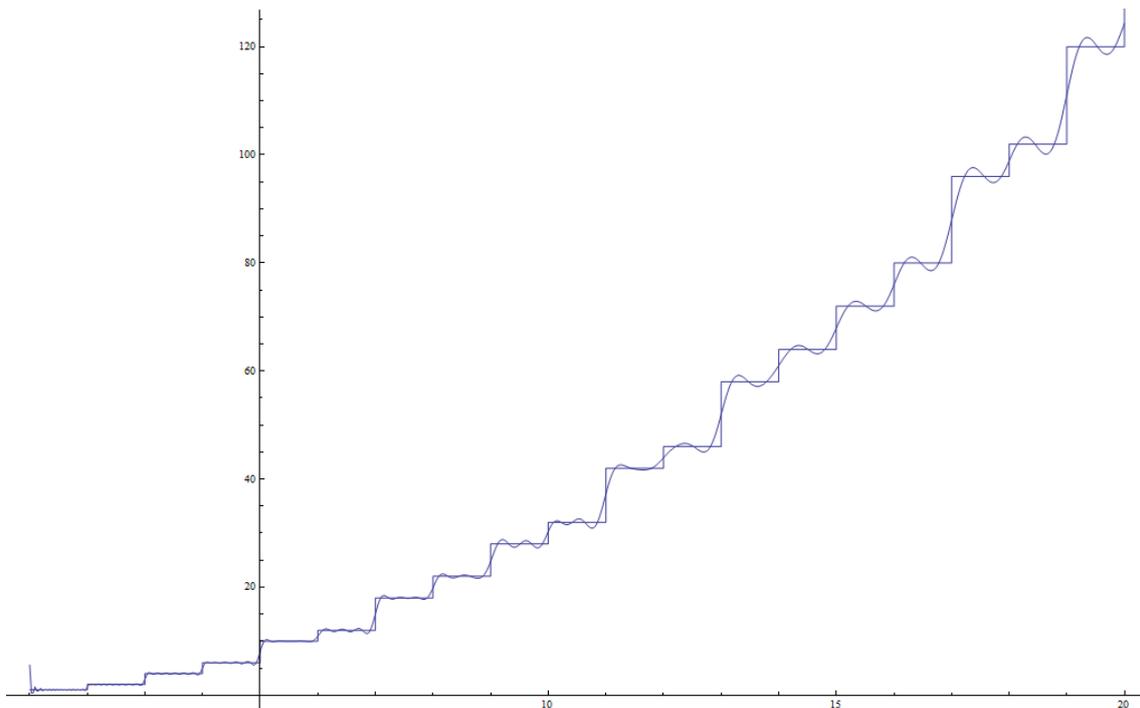


Abbildung 50: Summatorische Funktion $\Phi(n)$ der Phi-Funktion, dargestellt von 1 bis 100 (Vergleich zahlentheoretische und analytische Berechnungsmethode)

Mathematica:

```
Show[Plot[myPhi[x], {x, 1, 20}, MaxRecursion->2, PlotPoints->150],  
ListLinePlot[Table[{n, PhiSum[n]}, {n, 1, 40}], InterpolationOrder->0]]
```

Die analytische Version $\varphi(x)$ von $\varphi(n)$ lautet:

$$\varphi(x) = \Phi(x) - \Phi(x - 1) \quad (83)$$

Vergleich der analytischen $\varphi(x)$ mit der zahlentheoretischen Funktion $\varphi(n)$ (mit Summation über die ersten 50 nichttrivialen und 50 trivialen Nullstellen):

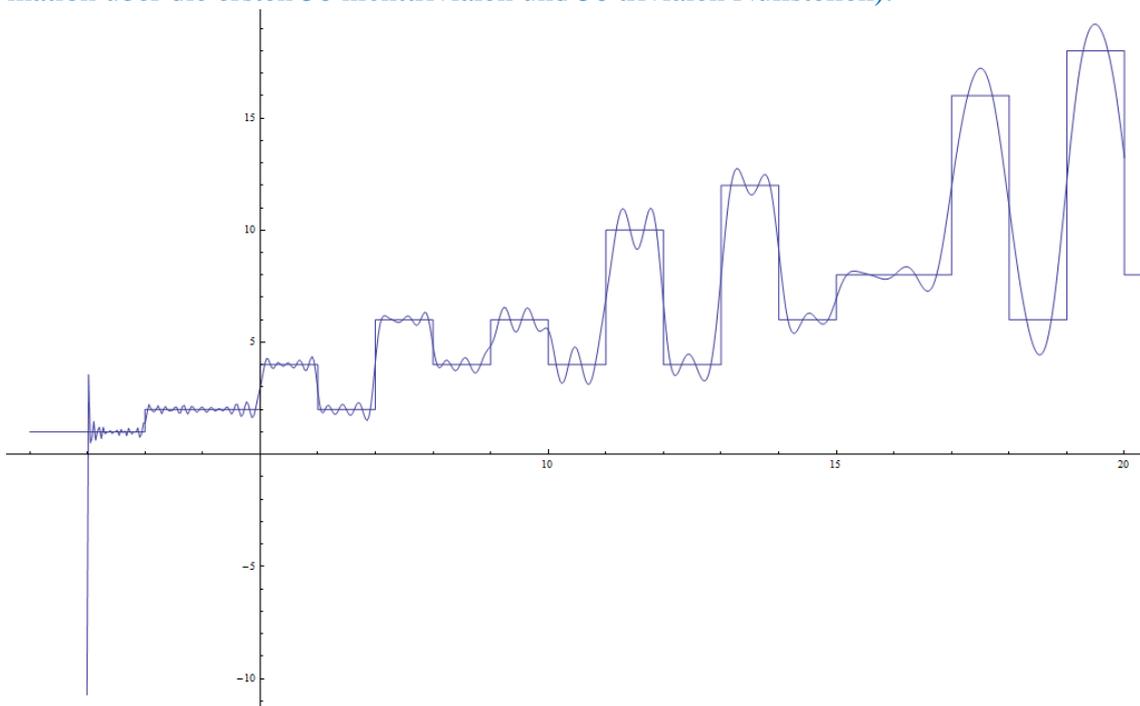


Abbildung 51: Euler Phi-Funktion (Vergleich analytische und zahlentheoretische Berechnung)

Mathematica:

```
myEulerPhi[x_]:=myPhi[x]-myPhi[x-1] (*Definition siehe oben *)
Show[Plot[myEulerPhi[x],{x,1,20},MaxRecursion->2,
PlotPoints->150],ListLinePlot[Table[{n,EulerPhi[n]},{n,1,40}],
InterpolationOrder->0] ]
```

7.4.2 EIGENSCHAFTEN DER PHI-FUNKTION

Eigenschaften von $\varphi(n)$:

$$\varphi(mn) = \varphi(m)\varphi(n) \quad (\text{wenn } \text{ggT}(m, n) = 1)$$

$$\varphi(p) = p - 1 \quad (\text{wenn } p \text{ Primzahl})$$

$$\varphi(p^k) = p^{k-1}(p - 1) = p^k \left(1 - \frac{1}{p}\right) \quad (\text{Primzahlpotenzen})$$

$$\varphi(n) = \frac{2}{n} \sum_{\substack{1 \leq j \leq n-1 \\ \text{ggT}(n,j)=1}} j$$

$$\text{ggT}(a, m) = 1 \Rightarrow a^{\varphi(m)} \equiv 1 \pmod{m} \quad (\text{Satz von Fermat-Euler})$$

$$p \nmid a \Rightarrow a^{p-1} \equiv 1 \pmod{p} \quad (\text{Spezialfall für Primzahlen, kleiner Fermatsche Satz})$$

Die Teilersummenfunktion (Sigma-Funktion)

$$\varphi(mn) = \varphi(m)\varphi(n) \frac{d}{\varphi(d)}, \text{ mit } d = ggT(m, n)$$

$$\varphi(n^m) = n^{m-1}\varphi(n)$$

$$\varphi(n)\sigma_0(n) = \sum_{\substack{1 \leq k \leq n \\ ggT(k, n) = 1}} ggT(k-1, n) \quad (84)$$

$$\sum_{n=1}^{\infty} \frac{\varphi(n)}{n^s} = \frac{\zeta(s-1)}{\zeta(s)} \quad (85)$$

Die Eulersche φ - Funktion wurde von Ramanujan verallgemeinert ($\varphi_1(n) = \varphi(n)$):

$$\varphi_s(n) = n^s \prod_{p|n} \left(1 - \frac{1}{p^s}\right) \quad (86)$$

Ramanujan berechnete $\varphi_s(n)$ zu:

$$\varphi_s(n) = \frac{\mu(n)n^s}{\zeta(s) \sum_{k=1}^{\infty} \frac{\mu(nk)}{k^s}} \quad (87)$$

$$\varphi(n) = \frac{\mu(n)n}{\zeta(s) \sum_{k=1}^{\infty} \frac{\mu(nk)}{k}} \quad (88)$$

$\varphi(n)$ kann auch mit Hilfe einer Ramanujan-Entwicklung berechnet werden:

$$\varphi(n) = \frac{n}{\zeta(s+1)} \sum_{q=1}^{\infty} \frac{\mu(q)c_q(n)}{\varphi_2(q)}$$

Diese Formel ist jedoch unpraktisch, da man zur Berechnung von $\varphi(n)$ auch $\varphi_2(k)$ braucht ($k = 1 \dots \infty$).

7.5 DIE TEILERSUMMENFUNKTION (SIGMA-FUNKTION)

7.5.1 DEFINITION, EIGENSCHAFTEN

Die **Teilersummenfunktion** $\sigma_k(n)$ berechnet die Summe der k -ten Potenzen der positiven Teiler von n (n mitgezählt).

Definition von $\sigma_k(n)$:

$$\sigma_k(n) = \sum_{d|n} d^k \quad (89)$$

Berechnung von $\sigma_k(n)$:

(a_i seien die Exponenten der Primfaktorenzerlegung $n = \prod_{i=1}^r p_i^{a_i}$)

$$\sigma_k(n) = \prod_{i=1}^r \frac{p_i^{(a_i+1)k} - 1}{p_i^k - 1} = \prod_{i=1}^r \sum_{j=0}^{a_i} p_i^{jk} \quad (90)$$

Wenn $\sigma_1(n)$ eine Primzahl ist, dann auch $\sigma_0(n)$. Hier die ersten 23 Paare:

(2, 3) (3, 7) (3, 13) (5, 31) (3, 31) (7, 127) (3, 307) (7, 1093)
 (3, 1723) (5, 2801) (3, 3541) (13, 8191) (3, 5113) (3, 8011)
 (3, 10303) (7, 19531) (3, 17293) (3, 28057) (5, 30941) (3, 30103)
 (17, 131071) (5, 88741) (3, 86143)

Mathematica:

```
For [i=1, i<100000, i++, If[PrimeQ[DivisorSigma[0, i]]==True&&
OddQ[DivisorSigma[0, i]], Print[DivisorSigma[0, i], " ",
FactorInteger[DivisorSigma[1, i]]]]]
```

Ungerade Primzahlwerte von $\sigma_0(n)$ sind selten, unter den ersten 100000 finden sich in aufsteigender Reihenfolge 79 Werte:

{3, 3, 5, 3, 3, 7, 5, 3, 3, 3, 3, 3, 5, 7, 3, 3, 11, 3, 3, 3, 3, 5, 3, 3, 3, 13, 3, 3,
 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 7, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 3, 3, 3, 3, 3, 3, 3,
 , 3, 3, 3, 3, 3, 3, 11, 3, 17, 3, 3, 3, 3, 3, 3, 3, 5, 3, 3, 3, 3}

Mit Mathematica:

```
Select[Select[DivisorSigma[0, Range[100000]], OddQ], PrimeQ]
```

Die Teilersummenfunktion (Sigma-Funktion)

Hier ein paar Plots von $\sigma_k(n)$ für verschiedene k :

Mathematica:

```
k=0;ListLinePlot[Table[{n, DivisorSigma[k, n]}, {n, 1, 50}],
InterpolationOrder->0]
```

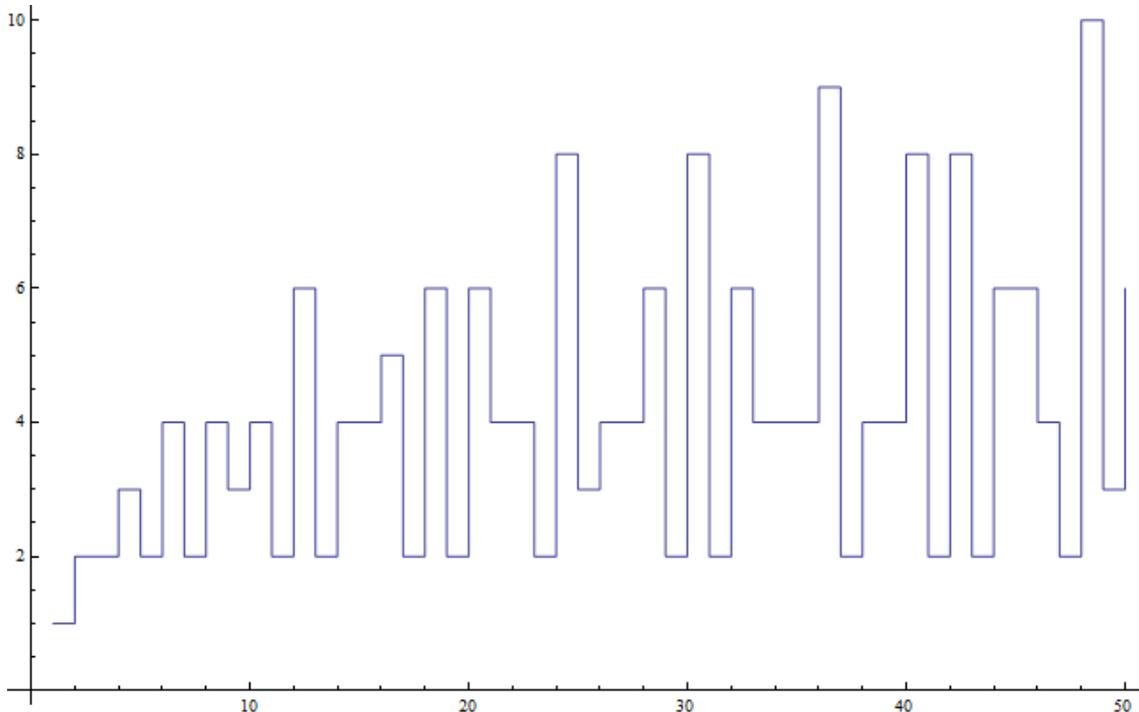


Abbildung 52: Anzahl-Teiler-Funktion $\sigma_0(n)$, dargestellt von 0 bis 50

Mathematica:

```
k=1;ListLinePlot[Table[{n, DivisorSigma[k, n]}, {n, 1, 100}],
InterpolationOrder->0]
```

Die ersten 100 Werte von $\sigma_0(n)$ lauten:

```
{1, 2, 2, 3, 2, 4, 2, 4, 3, 4, 2, 6, 2, 4, 4, 5, 2, 6, 2, 6, 4, 4, 2, 8, 3, 4, 4, 6, 2,
8, 2, 6, 4, 4, 4, 9, 2, 4, 4, 8, 2, 8, 2, 6, 6, 4, 2, 10, 3, 6, 4, 6, 2, 8, 4, 8, 4, 4,
2, 12, 2, 4, 6, 7, 4, 8, 2, 6, 4, 8, 2, 12, 2, 4, 6, 6, 4, 8, 2, 10, 5, 4, 2, 12, 4, 4,
4, 8, 2, 12, 4, 6, 4, 4, 4, 12, 2, 6, 6, 9}
```

Die ersten 100 Werte von $\sigma_1(n)$ lauten:

```
{1, 3, 4, 7, 6, 12, 8, 15, 13, 18, 12, 28, 14, 24, 24, 31, 18, 39, 20, 42, 32, 3
6, 24, 60, 31, 42, 40, 56, 30, 72, 32, 63, 48, 54, 48, 91, 38, 60, 56, 90, 42,
96, 44, 84, 78, 72, 48, 124, 57, 93, 72, 98, 54, 120, 72, 120, 80, 90, 60, 16
8, 62, 96, 104, 127, 84, 144, 68, 126, 96, 144, 72, 195, 74, 114, 124, 140,
96, 168, 80, 186, 121, 126, 84, 224, 108, 132, 120, 180, 90, 234, 112, 168
, 128, 144, 120, 252, 98, 171, 156, 217}
```

Primzahlwerte von $\sigma_1(n)$ sind selten, die ersten 37 Werte (aufsteigend) im Bereich bis 1 Million lauten:

```
{3, 7, 13, 31, 31, 127, 307, 1093, 1723, 2801, 3541, 8191, 5113, 8011, 10
303, 19531, 17293, 28057, 30941, 30103, 131071, 88741, 86143, 147073
, 524287, 292561, 459007, 492103, 797161, 552793, 579883, 598303, 68
4757, 704761, 732541, 735307, 830833}
```

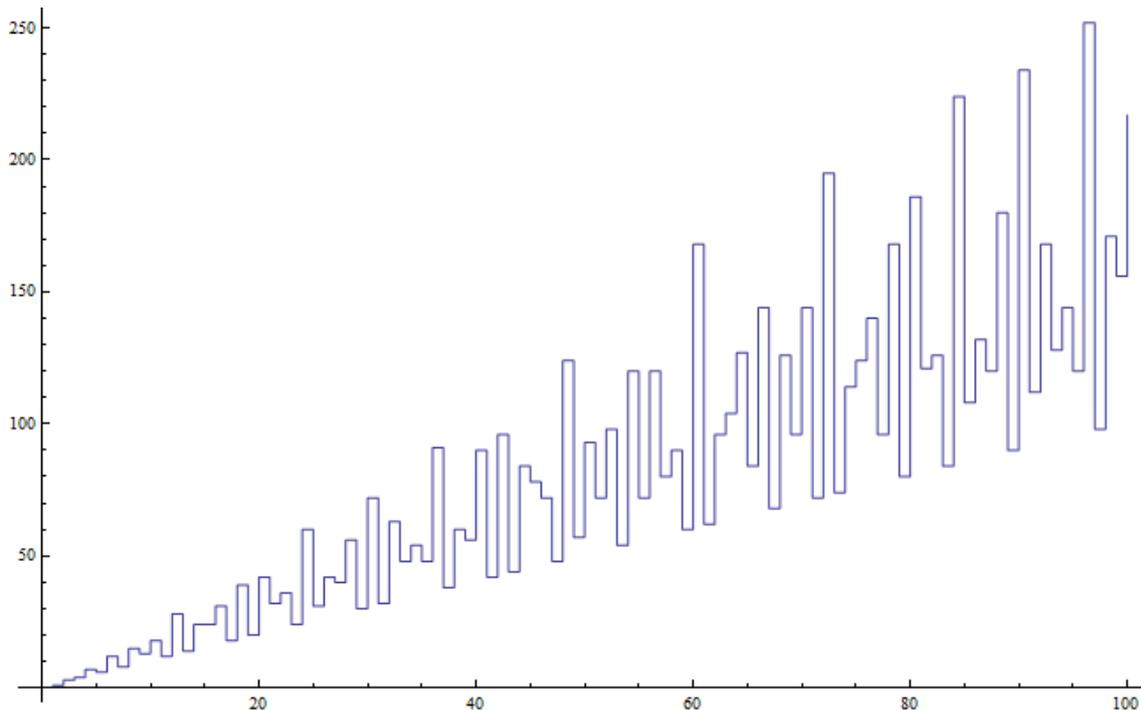


Abbildung 53: Teilersummen-Funktion $\sigma_1(n)$, dargestellt von 0 bis 100

Die Sigma-Funktion kann in eine Ramanujan-Reihe mit den Ramanujan-Summen $c_q(n)$ als Koeffizienten entwickelt werden (man beachte, dass s und n nicht ganzzahlig sein müssen):

$$\sigma_s(n) = n^s \zeta(s+1) \sum_{q=1}^{\infty} \frac{c_q(n)}{q^{s+1}} \quad (91)$$

sowie

$$\sigma_0(n) = - \sum_{q=1}^{\infty} \frac{\ln(q)}{q} c_q(n) \quad (92)$$

Wie in Kapitel 7.9.3 dargestellt, ist es möglich, den Definitionsbereich die Ramanujan-Summen $c_q(n)$ auf reelle bzw. komplexe Werte zu erweitern. Wir bekommen statt (91) dann eine ‘‘Ramanujan Summen-Funktion‘‘,

$$\sigma_s(x) = x^s \zeta(s+1) \sum_{q=1}^{\infty} \frac{c_q(x)}{q^{s+1}} \quad (93)$$

deren beide Real- und Imaginärteile ziemlich schnell oszillieren. Diese komplexe Funktion stellt eine wunderschöne Erweiterung auf \mathbb{R} bzw. \mathbb{C} der nur für ganzzahlige Werte n definierten Ramanujan-Summen dar. Am deutlichsten sehen wir dies, wenn wir den Betrag dieser Funktion anschauen: Man sieht deutlich, wie die Funktion an den ganzzahligen

Die Teilersummenfunktion (Sigma-Funktion)

Stellen exakt mit der zahlentheoretischen Version übereinstimmt. Es wäre interessant, zu untersuchen, welche Information in der ‚Phase‘ dieser Funktion steckt. Es sieht aus, als ob die Phase ‚rotiert‘, manchmal schneller, manchmal langsamer (s. Abbildung 55).

Asymptotisches Verhalten von $\sigma_k(n)$:

$$\sigma_1(n) < e^\gamma n \ln(\ln(n)) + \frac{0.6483}{\ln(\ln(n))}, n > 3$$

Eigenschaften von $\sigma_k(n)$:

$\sigma_0(p) = 2$ (Jede Primzahl besitzt nur 2 Teiler: sich selbst und die 1)

$\sigma_0(p^n) = n + 1$

$\sigma_0(n) = \prod_{i=1}^r (a_i + 1)$ (a_i siehe Formel (90))

$\sigma_1(p) = p + 1$

Es gibt unendliche viele n so dass $\sigma_0(n) = \sigma_0(n + 1)$

Vermutungen:

Die einzige Zahl n , für die $\sigma_2(n)$ prim ist, ist $n = 2$, mit $\sigma_2(2) = 5$.

7.5.2 GRAPHISCHE DARSTELLUNGEN DER SIGMA-FUNKTION

Hier ein paar Grafiken, in denen die analytisch mit Ramanujan-Reihen berechneten Werte von $\sigma(n)$ (in rot) mit den exakten Werten (in blau), die zahlentheoretisch berechnet wurden, verglichen werden. Man sieht deutlich, dass die rote Kurve bei den ganzzahligen Werten exakt mit dem zahlentheoretischen Wert übereinstimmt.

Mathematica Code für die folgende Abbildung:

```
cnqx[q_, n_] := Sum[If[GCD[a, q] == 1, Exp[2.0*Pi*I*a*(n/q)], 0], {a, 1, q}];
s=1.0;
Show[Plot[n^s*Zeta[s+1]*Abs[Sum[cnqx[q, n]/q^(s+1), {q, 1, 1000}]],
{n, 1, 12}, PlotStyle->Red], ListLinePlot[Table[{k, DivisorSigma[s, k]},
{k, 1, 12}], InterpolationOrder->0]]
```

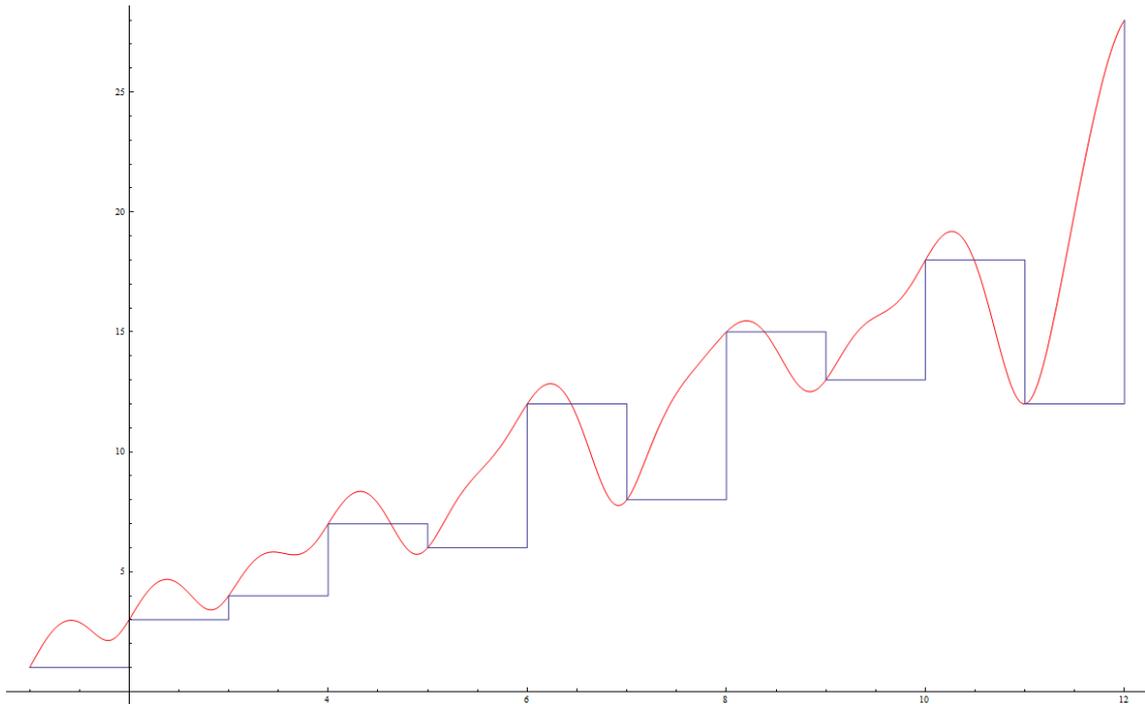


Abbildung 54 $|\sigma_1(x)|$: Vergleich der analytisch berechneten Werte von Sigma mit den exakten Werten. Es wurden die Ramanujan-Summen $c_q(n)$ bis $q=1000$ ausgewertet, n läuft von 0 bis 12

Mathematica Code für die folgende Abbildung:

```
cnqx[q_, n_] := Sum[If[GCD[a, q] == 1, Exp[2.0*Pi*I*a*(n/q)], 0], {a, 1, q}];
s = 1.0;
Show[Plot[Arg[Sum[cnqx[q, n]/q^(s+1), {q, 1, 100}]], {n, 1, 12},
PlotStyle -> Red], ListLinePlot[Table[{k, DivisorSigma[s, k]},
{k, 1, 12}], InterpolationOrder -> 0]]
```

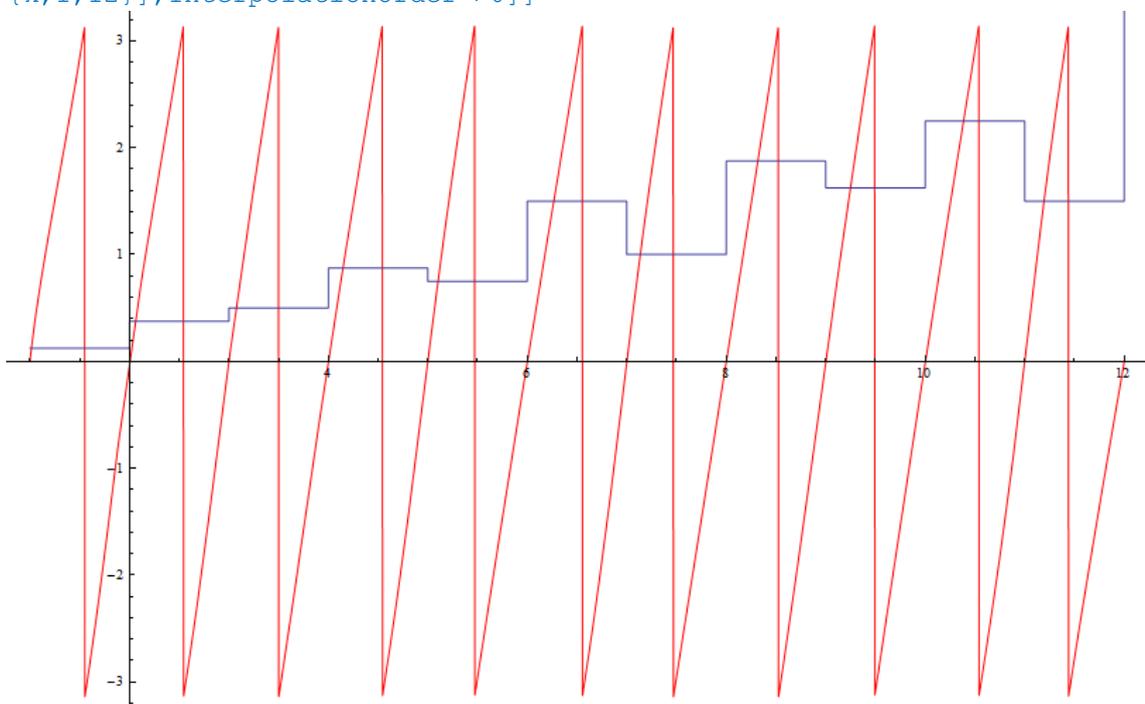


Abbildung 55: $\text{Arg}(\sigma_1(x))$: Phase der erweiterten Sigmafunktion. Es wurden die Ramanujan-Summen $c_q(n)$ bis $q=100$ ausgewertet, n läuft von 0 bis 12. Die blaue Kurve wurde skaliert.

Die Teilersummenfunktion (Sigma-Funktion)

Mathematica Code für die folgende Abbildung:

```
cnqx[q_, n_] := Sum[If[GCD[a, q] == 1, Exp[2.0*Pi*I*a*(n/q)], 0], {a, 1, q}];  
s = 1.0;  
Plot[n^s*Zeta[s+1]*Abs[Sum[cnqx[q, n]/q^(s+1), {q, 1, 50}]], {n, 1, 5000},  
MaxRecursion -> 2, PlotPoints -> 3000]
```

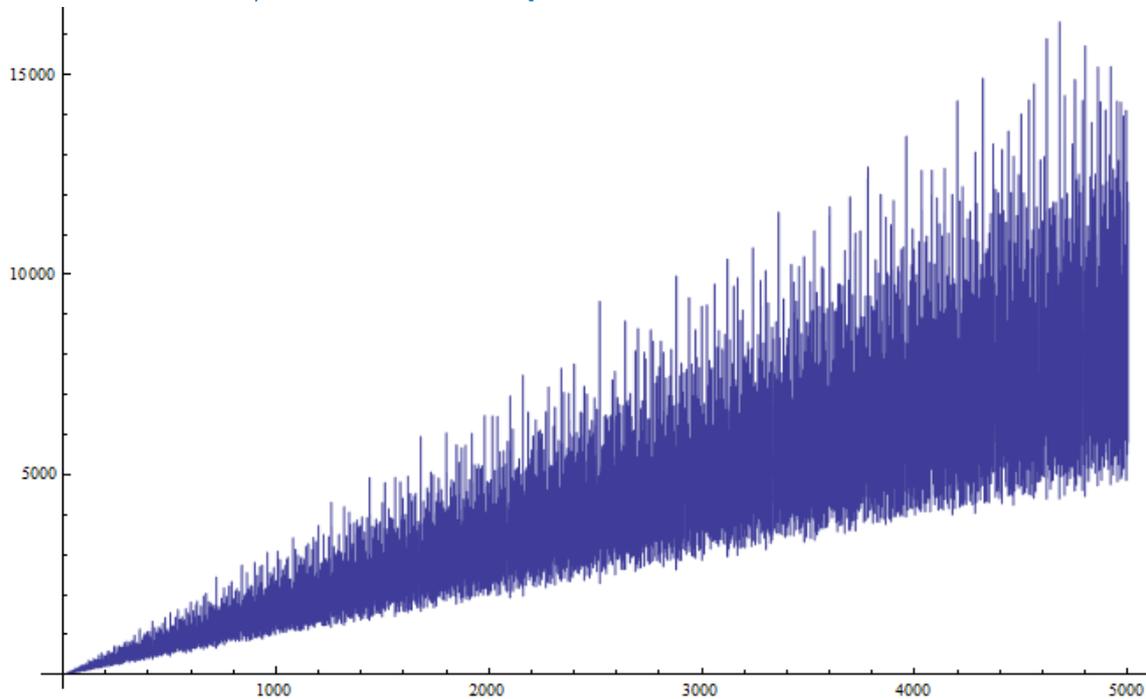


Abbildung 56 $|\sigma_1(x)|$: Analytisch berechnete Werte von Sigma. Es wurden die Ramanujan-Summen $c_q(n)$ bis $q=50$ ausgewertet, n läuft von 0 bis 5000

Mathematica Code für die folgende Abbildung:

```
Show[Monitor[Plot[n^s*Zeta[s+1]*Abs[Sum[cnqx[q, n]/q^(s+1), {q, 1, 100}]]  
, {n, 100, 150}, PlotStyle -  
> Red], n], ListLinePlot[Table[{k, DivisorSigma[s, k]}, {k, 100, 150}], Interpo  
lationOrder -> 0]]
```

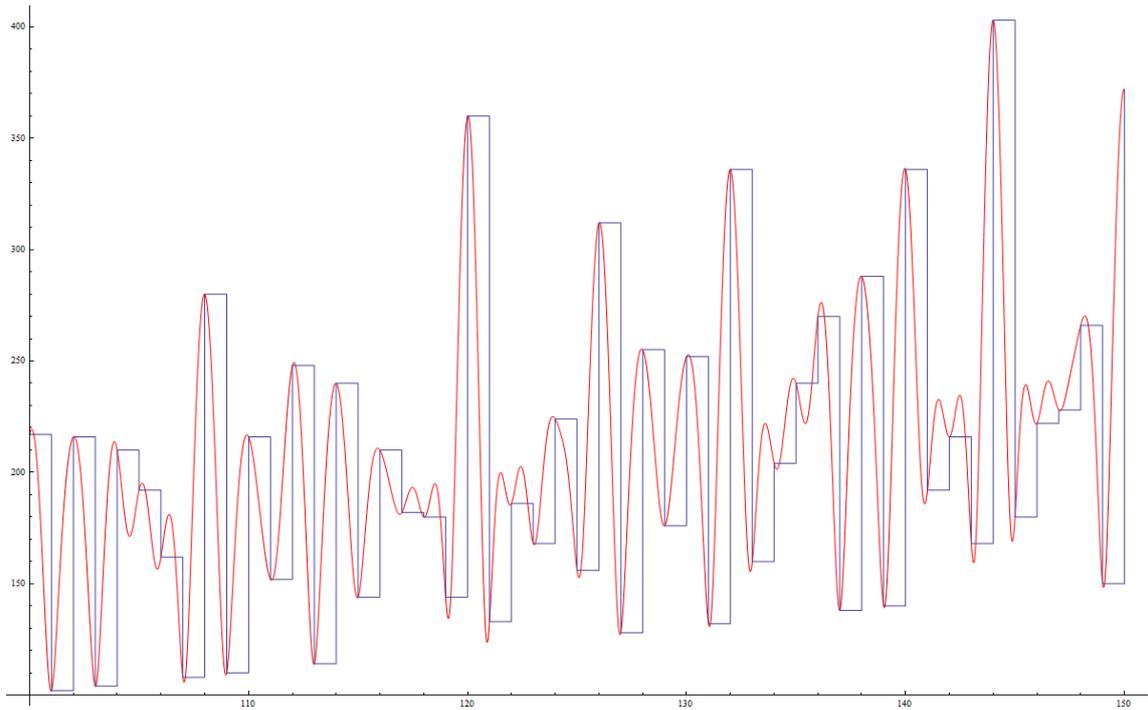


Abbildung 57 $|\sigma_1(x)|$: Vergleich der analytisch berechneten Werte von Sigma mit den exakten Werten. Es wurden die Ramanujan-Summen $c_q(n)$ bis $q=1000$ ausgewertet, n läuft von 100 bis 150

Mathematica Code für die folgende Abbildung:

```
Show[Plot[n^s*Zeta[s+1]*Abs[Sum[cnqx[q,n]/q^(s+1),{q,1,3000}]]-n-1,
{n,1000000000,1000000100},PlotStyle->Red],ListLinePlot[
Table[{k,DivisorSigma[s,k]-k-1},{k,1000000000,1000000100}],InterpolationOrder->0]]
```

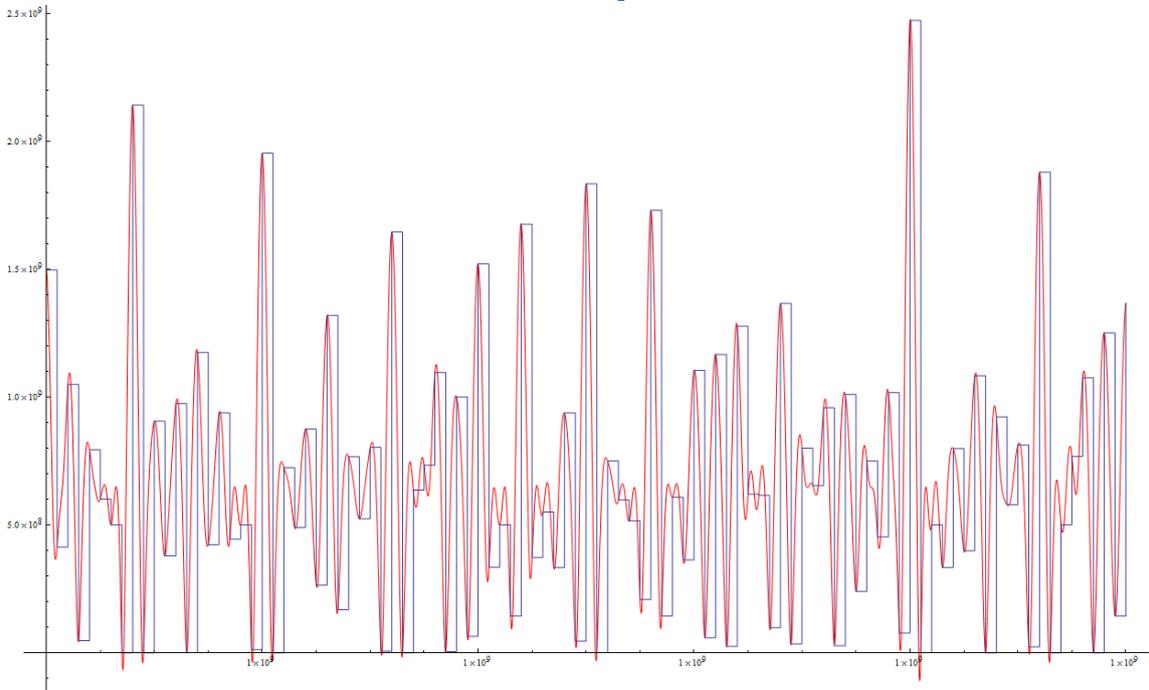


Abbildung 58: $|\sigma_1(x)|-x-1$: Vergleich der analytisch berechneten Werte von Sigma mit den exakten Werten. Es wurden die Ramanujan-Summen $c_q(n)$ bis $q=3000$ ausgewertet, n läuft von $1E9$ bis $1E9+10$. Die Nullstellen fallen mit Primzahlen zusammen

Die Ramanujansche Tau-Funktion

Weitere Formeln zur Sigma-Funktion:

$$\sigma_k(n) = \sum_{m=1}^n m^{k-1} \sum_{j=1}^m \cos\left(\frac{2\pi j n}{m}\right) \quad (94)$$

Mathematica:

```
myDivisorSigma[k_, n_] := Sum[m^(k-1)
Sum[Cos[(2 Pi j n)/m], {j, 1, m}], {m, 1, n}]
```

Graph nach Formel (94):

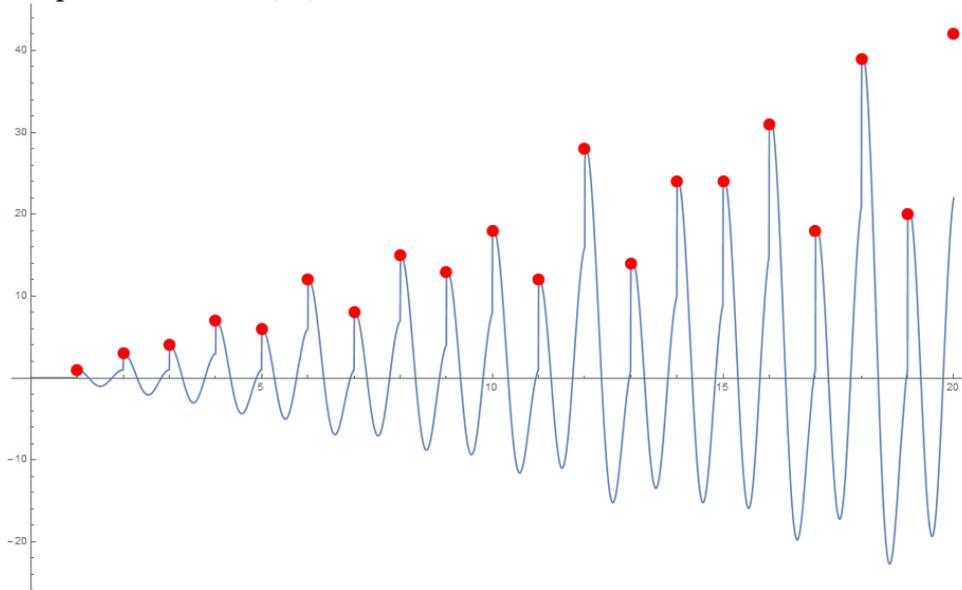


Abbildung 59: Sigma-Funktion analytisch mit Cos()-Termen berechnet

Mathematica:

```
myDivisorSigma[k_, n_] := Sum[m^(k-1) Sum[Cos[(2 Pi j
n)/m], {j, 1, m}], {m, 1, n}]
xmin=0; xmax=20;
Show[Plot[N[myDivisorSigma[1, x]], {x, xmin, xmax},
PlotRange->All, AxesOrigin->{0, 0}], ListPlot[Table[{n, DivisorSigma[1, n]},
{n, xmin, xmax}], PlotStyle->Red]]
```

$$\sigma_0(n) = \sum_{m=1}^{\infty} (-1)^{m+1} (2\pi n)^{2m} \sum_{j=1}^m \frac{(-1)^j 2^{2j-1} \pi^{2j} (B_{2j})^2}{((2j)!)^2 (-2j + 2m + 1)!} \quad (95)$$

Mathematica:

```
myDivisorSigma0[n_] == Sum[(-1)^(m + 1) (2 Pi n)^(2 m)
Sum[((-1)^j (2 Pi)^(2 j) BernoulliB[2 j]^2)/(2 (2 j)!^2 (2 m + 1 - 2
j)!), {j, 1, m}], {m, 1, Infinity}]
```

7.6 DIE RAMANUJANSCHES TAU-FUNKTION

In der mathematischen Literatur findet man gleich vier verschiedene Ramanujan Tau-Funktionen (die Argumente deuten auf den am häufigsten verwendeten Definitionsbereich, n : ganzzahlig, s : komplex, t : reell):

$\tau(n)$: Ramanujan-Tau-Funktion, Mathematica: `RamanujanTau[n]`

$L(s)$: Ramanujan-Tau-Dirichlet-L-Funktion, Mathematica: `RamanujanTauL[s]`

$Z(t)$: Ramanujan-Tau-Z-Funktion, Mathematica: `RamanujanTauZ[n]`

$\Theta(t)$: Ramanujan-Tau-Theta-Funktion, Mathematica: `RamanujanTauTheta[n]`

Der Graph von $\tau(n)$:

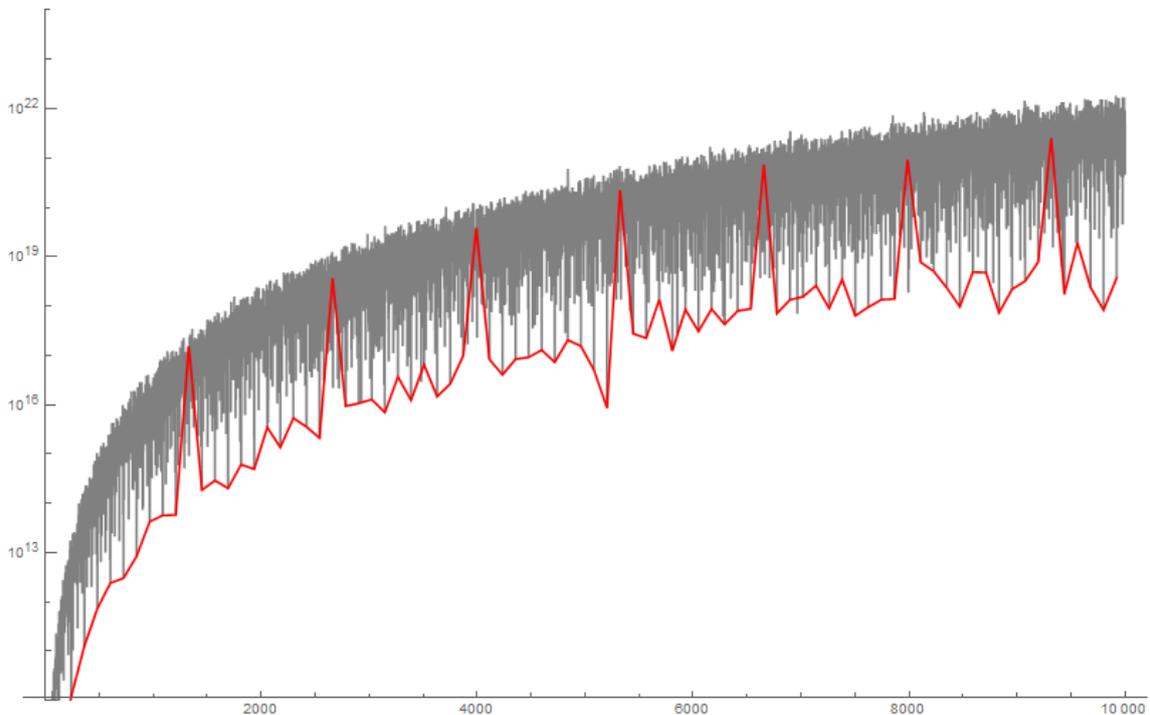


Abbildung 60: Ramanujan $\tau(N)$ (grau), in Rot: Nur n für $\text{Mod}(n, 11^2)=0$

Mathematica:

```
Show[ListLogPlot[Table[{n, Abs[RamanujanTau[n]]}, {n, 1, 10000}], Joined->True, PlotRange->{10^10, 10^24}, PlotStyle->Gray],
ListLogPlot[Table[{n, Abs[RamanujanTau[n]]}, {n, 121, 10000, 121}], Joined->True, PlotRange->{10^10, 10^24}, PlotStyle->Red, InterpolationOrder->1]]
```

**Die Ramanujan-Tau-Funktion besitzt viele Darstellungen (bzw. Definitionen):
Durch ihre erzeugende Funktion³⁹ $G(x)$:**

$$\begin{aligned} G(x) &= x \prod_{n=1}^{\infty} (1 - x^n)^{24} = \sum_{n=1}^{\infty} \tau(n)x^n \\ &= x - 24x^2 + 252x^3 - 1472x^4 + 4830x^5 \\ &\quad - 6048x^6 + \dots = x(1 - 3x + 5x^3 - 7x^6 + \dots)^8 \end{aligned} \quad (96)$$

(*Mathematica (die ersten 50 Werte):*)

```
CoefficientList[Take[Expand[Product[(1-x^k)^24, {k, 1, 50}]], 50], x]:
```

³⁹ Eine Folge $a(n)$ kann durch die Koeffizienten einer Potenzreihenentwicklung $f(x) = \sum_{n=0}^{\infty} a_n x^n$ definiert werden. $f(x)$ heisst ‚erzeugende Funktion‘ von $a(n)$

Die Ramanujansche Tau-Funktion

{1, -24, 252, -1472, 4830, -6048, -16744, 84480, -113643, -115920, ...}

```
(*From Theory of modular Forms:*)
max = 28; g[k_] := -BernoulliB[k]/(2k) + Sum[ DivisorSigma[k - 1, n -
1]*q^(n - 1), {n, 2, max + 1}]; CoefficientList[ Series[ 8000*g[4]^3 -
147*g[6]^2, {q, 0, max}], q] // Rest
```

Eigenschaften von $\tau(n)$:

$$\tau(p^{r+1}) = \tau(p)\tau(p^r) - p^{11}\tau(p^{r-1}), \text{ wenn } p \in \mathbb{P} \text{ und } r > 0 \quad (97)$$

$$|\tau(p)| \leq 2p^{\frac{11}{2}}, \text{ wenn } p \in \mathbb{P} \quad (98)$$

$\tau(n)$ ist multiplikativ: $\tau(mn) = \tau(m)\tau(n)$, wenn $\text{ggT}(m, n) = 1$

Es gibt sehr viele Beziehungen zwischen $\tau(n)$ und den Teilersummen-Funktionen $\sigma_k(n)$.⁴⁰

Ein Beispiel:

$$\tau(n) = \frac{65}{756}\sigma_{11}(n) + \frac{691}{756}\sigma_5(n) - \frac{691}{3}\sum_{k=1}^{n-1}\sigma_5(k)\sigma_5(n-k) \quad (99)$$

Von Ramanujan stammen die folgenden rekursiven Formeln:

$$\begin{aligned} (n-1)\tau(n) &= \sum_{m=1}^{b_n} (-1)^{m+1} (2m \\ &\quad + 1) \\ &\quad \times \left(n - 1 - \frac{9}{2}m(m+1) \right) \tau \left(n - \frac{1}{2}m(m+1) \right), \text{ mit } b_n \\ &= \frac{1}{2}(\sqrt{8n+1} - 1) \end{aligned} \quad (100)$$

$$\tau(p^n) = \sum_{j=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^j \binom{n-j}{n-2j} p^{11j} (\tau(p))^{n-2j} \quad (101)$$

Die Ramanujan-Tau-L-, Tau-Theta- und Tau-Z-Funktionen:

$$Z(t) = e^{i\theta(t)} L(it + 6) \quad (102)$$

⁴⁰ https://en.wikipedia.org/wiki/Ramanujan_tau_function oder: <http://mathworld.wolfram.com/TauFunction.html>

Oder (ähnlich wie die Zerlegung der Zeta-Funktion mit Hilfe der Riemann-Siegel Funktion):

$$L(it + 6) = e^{-i\theta(t)} Z(t)$$

$$L(s) = \sum_{n=1}^{\infty} \frac{\tau(n)}{n^s} \quad (103)$$

Wobei $\theta(t)$ die Ramanujan Tau-Theta-Funktion und $L(s)$ die Ramanujan Tau-L-Funktion sind. Die Funktion $L(s)$ ist auch unter der Bezeichnung ‚Ramanujans Dirichlet-L-Reihe‘ bekannt. Sie besitzt ähnliche Eigenschaften wie Riemanns Zeta-Funktion $\zeta(z)$. Tatsächlich gehört sie zum Typ der verallgemeinerten Zeta-Funktionen. Ramanujan vermutete, dass alle nicht-trivialen Nullstellen von $L(s)$ auf der ‚kritischen‘ Geraden $\operatorname{Re}[s] = 6$ liegen.

Ähnlich wie die Zeta-Funktion besitzt auch $L(s)$ eine Eulersche Produktdarstellung:

$$L(s) = \prod_{p \in \mathbb{P}} \frac{1}{1 - \tau(p)p^{-s} + p^{11-2s}} \quad (104)$$

Weitere Formeln findet der Leser im Internet.⁴¹

Graphische Darstellungen (schwarz: Realteil, rot: Imaginärteil):

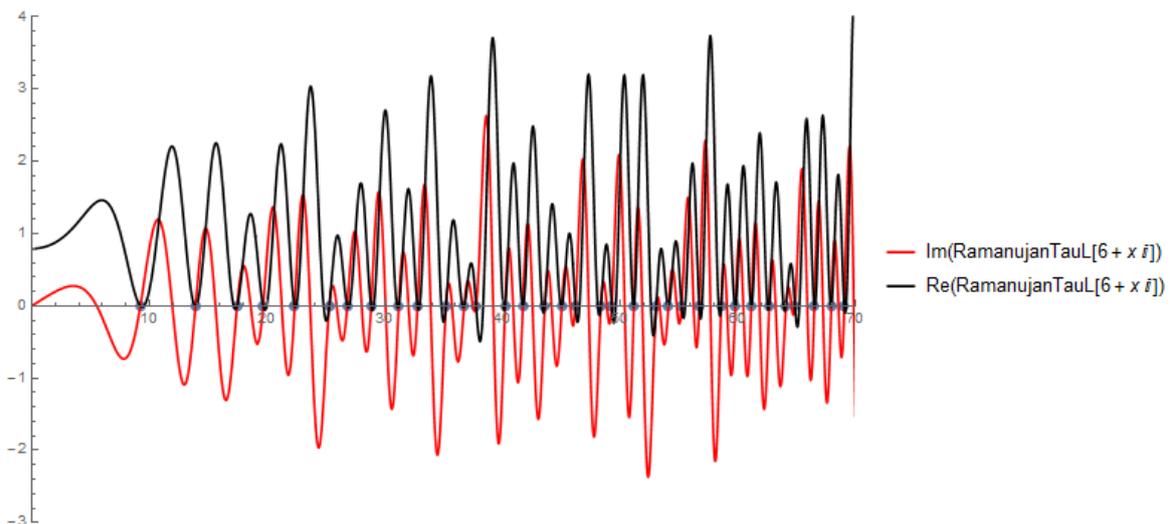


Abbildung 61: Ramanujan-Tau-L-Funktion (Dirichlet-L-Reihe) 0-70, mit 34 Nullstellen, entlang der kritischen Geraden

Mathematica:

```
Show[Plot[{Im[RamanujanTauL[6+x I]], Re[RamanujanTauL[6+x I]]}, {x, 0, xmax}, PlotStyle->{Red, Black}, PlotLegends->"Expressions", PlotRange->{{0, 70}, {-3., 4}}, ImageSize->Large]
```

⁴¹ <http://mathworld.wolfram.com/TauDirichletSeries.html>

Die Mertens-Funktion

Die Nullstellen der Ramanujan Tau-L-Funktion $L(s)$ liegen etwa doppelt so dicht, wie bei der Zeta-Funktion $\zeta(s)$. Im Bereich bis 70 liegen bei der ζ -Funktion 17 Nullstellen, bei der L-Funktion 34 Nullstellen.

Eine Tabelle mit den ersten Nullstellen der Ramanujanschen Tau-L-Funktion findet sich im Anhang „Nullstellen der Ramanujanschen Tau-L-Funktion“

7.7 DIE MERTENS-FUNKTION

Die Mertens-Funktion $M(n)$ ist die summatorische Funktion der Möbius-Funktion $\mu(n)$:

$$M(n) = \sum_{k=1}^n \mu(k) \quad (105)$$

Definition von $\mu(n)$ (für $n > 0$):

$$\mu(n) = \left\{ \begin{array}{l} 1, \text{ wenn } n \text{ quadratfrei und mit einer geraden Anzahl von Primfaktoren} \\ -1, \text{ wenn } n \text{ quadratfrei und mit einer ungeraden Anzahl von Primfaktoren} \\ 0, \text{ wenn } n \text{ einen quadratischen Primfaktor enthält} \end{array} \right\}$$

$\mu(n)$ kann berechnet werden, ohne die Primfaktorenzerlegung von n zu kennen (jedoch mit der gleichen Komplexität):

$$\mu(n) = \sum_{\substack{1 \leq k \leq n \\ \text{ggT}(k,n)=1}} e^{-2\pi i \frac{k}{n}} \quad (106)$$

Eigenschaften der μ -Funktion:

$$\sum_{k=1}^{\infty} \frac{\mu(k)}{k} = 0 \quad (107)$$

Interessant ist die Darstellung als Summe über Farey-Sequenzen:

$$M(n) = \sum_{a \in \mathcal{F}_n} e^{-2\pi i a} \quad (108)$$

Darstellung der Möbius-Funktion:

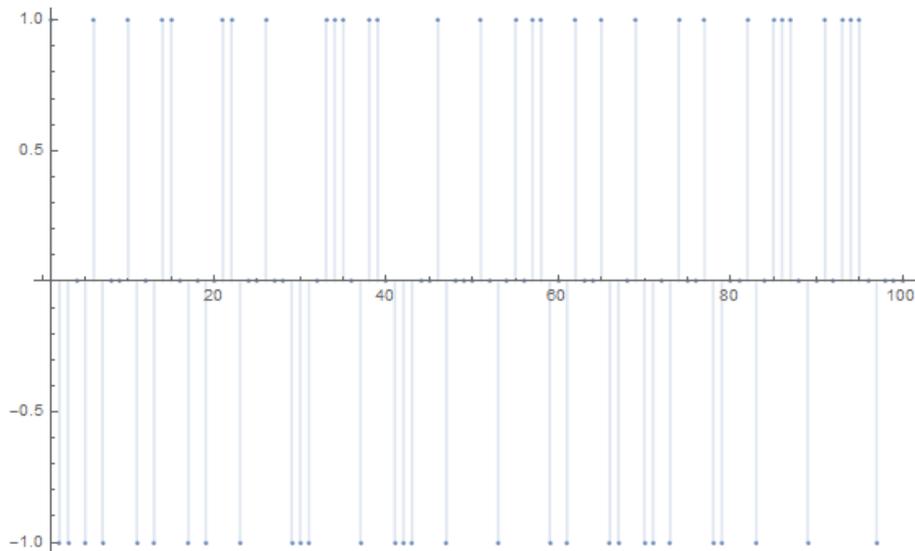


Abbildung 62: Möbius-Funktion $\mu(n)$, von 1 bis 100

Mathematica:

```
DiscretePlot[MoebiusMu[k], {k, 100}]
```

Hinweis: Die Möbius-Funktion $\mu(n)$ kann auch für negative n bzw. ganzzahlige komplexe Zahlen definiert werden.

Formeln:

$\mu(n)$ hängt mit der Riemannschen Zeta-Funktion zusammen:

$$\sum_{n=1}^{\infty} \frac{\mu(n)}{n^s} = \frac{1}{\zeta(s)} \quad (109)$$

Darstellung der Mertens-Funktion:

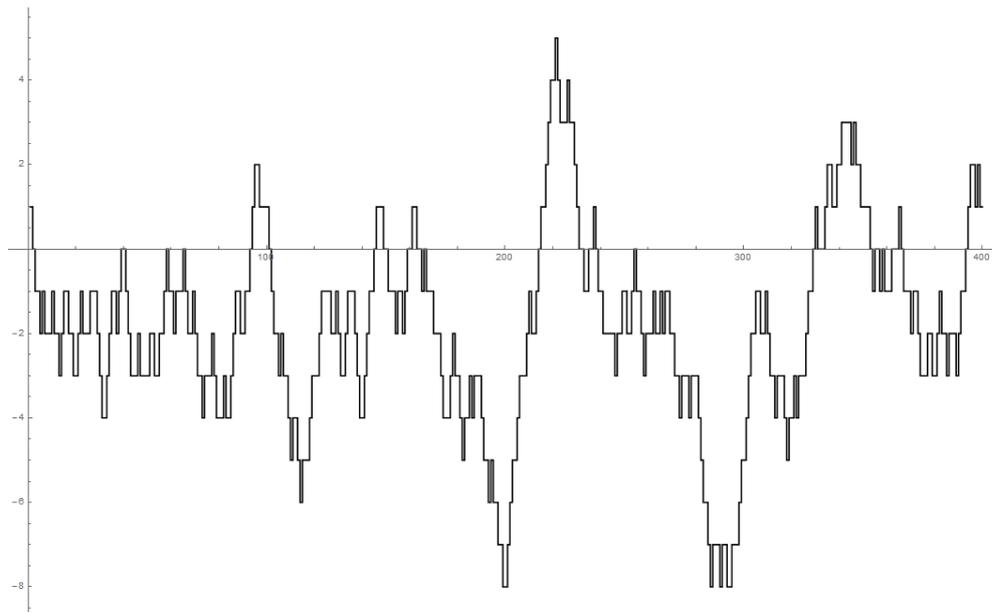


Abbildung 63: Mertens-Funktion $M(n)$ von 1 bis 400

Das Radikal

Mathematica:

```
m[n_]:=Sum[MoebiusMu[k],{k,1,n}]
ListLinePlot[Table[m[n],{n,400}],InterpolationOrder->0,
PlotStyle->Black]
```

Die Mertens-Funktion besitzt Nullstellen bei:

2, 39, 40, 58, 65, 93, 101, 145, 149, 150, 159, 160, 163, 164, 166, 214, 231, 232, 235, 236, 238, 254, 329, 331, 332, 333, 353, 355, 356, 358, 362, 363, 364, 366, 393...

Es existiert auch eine rekursive Darstellung der Mertens-Funktion (hier mit Hilfe des folgenden Mathematica-Programmes berechnet):

Mathematica:

```
(*Conjectured recurrence (two combined recurrences):*)
t[n_,k_]:=t[n,k]=If[And[n==1,k==1],3,If[Or[And[n==1,k==2],And[n==2,k==1]],2,If[n==1,(-t[n,k-1]-Sum[t[i,k],{i,2,k-1}])/(k+1)+t[n,k-1],If[k==1,(-t[n-1,k]-Sum[t[n,i],{i,2,n-1}])/(n+1)+t[n-1,k],If[n>=k,-Sum[t[n-i,k],{i,1,k-1}],-Sum[t[k-i,n],{i,1,n-1}]]]]];
nn=100;
MatrixForm[Table[Table[t[n,k],{k,1,nn}],{n,1,nn}]];
Table[t[1,k],{k,1,nn}]-2 (*Mats Granvik,Jul 10,2011*)
```

Weitere interessante zahlentheoretische Zusammenhänge mit der Zeta-Funktion und anderen Funktionen sind im Internet nachzulesen.⁴²

7.8 DAS RADIKAL

Das Radikal $\text{rad}(n)$ ist definiert als das Produkt unterschiedlicher Primfaktoren von n :

$$\text{rad}(n) = \prod_{\substack{p|n \\ p \in \mathbb{P}}} p \quad (110)$$

Die Berechnung mit Mathematica ist sehr einfach:

```
Table[Last[Select[Divisors[n],SquareFreeQ]],{n,100}]
rad[n_]:=Times@@(First@#&/@FactorInteger@n);Array[rad,100]
```

Die ersten 50 Werte lauten:

{1, 2, 3, 2, 5, 6, 7, 2, 3, 10, 11, 6, 13, 14, 15, 2, 17, 6, 19, 10, 21, 22, 23, 6, 5, 26, 3, 14, 29, 30, 31, 2, 33, 34, 35, 6, 37, 38, 39, 10, 41, 42, 43, 22, 15, 46, 47, 6, 7, 10}

⁴² https://en.wikipedia.org/wiki/Mertens_function

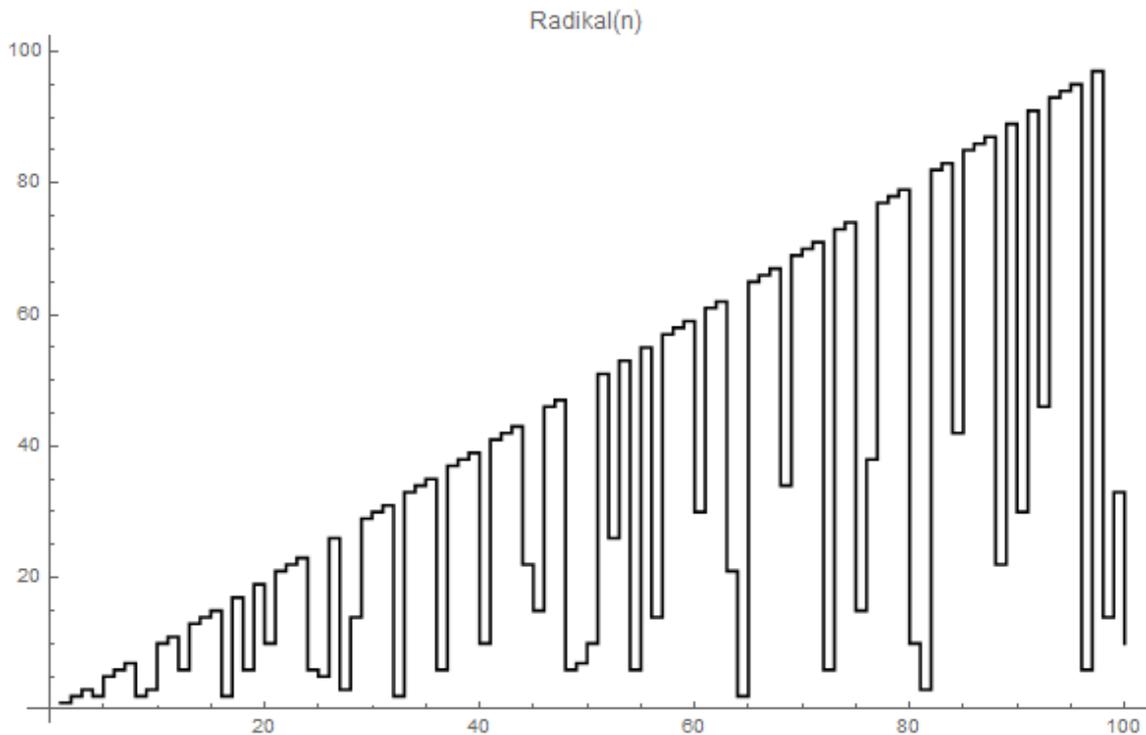


Abbildung 64: Radikal(n) (n=1,100)

Eigenschaften:

Eine wichtige Anwendung findet die Funktion $\text{rad}(n)$ in der Untersuchung der ‚abc-Vermutung‘ (Kapitel 11.1).

Hinweis:

Die Möbius-Transformation von $\text{rad}(n)$ ergibt die Absolutwerte von $\mu(n)\varphi(n)$.

7.9 RAMANUJAN-SUMMEN

Ramanujan-Reihen:

Mit Hilfe der Ramanujan-Summen $c_q(n)$ können viele zahlentheoretische Funktionen durch eine sogenannte Ramanujan-Reihenentwicklung dargestellt werden:

$$\mathbf{0} = \sum_{k=1}^{\infty} \frac{1}{k} \mathbf{c}_q(\mathbf{n}) \quad (\text{Ramanujan-Reihe der Null-Funktion})$$

$$\sigma_s(\mathbf{n}) = \mathbf{n}^s \zeta(s+1) \sum_{q=1}^{\infty} \frac{c_q(\mathbf{n})}{q^{s+1}} \quad (\text{Ramanujan-Reihe der Sigma-Funktion})$$

$$\sigma_0(\mathbf{n}) = - \sum_{q=1}^{\infty} \frac{\ln(q)}{q} \mathbf{c}_q(\mathbf{n}) \quad (\text{Teiler-Anzahl-Funktion})$$

7.9.1 DEFINITION

Ramanujans Summe ist eine Funktion zweier ganzer Zahlen n und q :

$$c_q(n) = \sum_{\substack{a=1 \\ (a,q)=1}}^q e^{2\pi i \frac{a}{q} n}, \quad n = 0, 1, 2, \dots \quad (a, 0) \text{ sei als } a \text{ definiert} \quad (111)$$

$(a, q) = 1$ bedeutet $\text{ggT}(a, q)$ soll 1 sein, d.h. a und q sollen zueinander ‚koprim‘ sein. D.h. die Summe läuft über alle a , die zu q ‚koprim‘ ($\text{ggT}(a, q) = 1$) sind.

Beispiel: Die Ramanujan Summen für $q = 1$ bis 15 (und $n = 0$ bis 20) lauten (die Periodizität beträgt jeweils q , in rot dargestellt):

$$\begin{aligned} c_1(n) &: \{1, \mathbf{1}, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \dots\} \\ c_2(n) &: \{1, \mathbf{-1}, \mathbf{1}, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, \dots\} \\ c_3(n) &: \{2, \mathbf{-1}, \mathbf{-1}, \mathbf{2}, -1, -1, 2, -1, -1, 2, -1, -1, 2, -1, -1, 2, -1, -1, 2, -1, \dots\} \\ c_4(n) &: \{2, \mathbf{0}, \mathbf{-2}, \mathbf{0}, \mathbf{2}, 0, -2, 0, 2, 0, -2, 0, 2, 0, -2, 0, 2, 0, -2, 0, 2, \dots\} \\ c_5(n) &: \{4, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{4}, -1, -1, -1, -1, 4, -1, -1, -1, -1, 4, -1, -1, -1, -1, 4, \dots\} \\ c_6(n) &: \{2, \mathbf{1}, \mathbf{-1}, \mathbf{-2}, \mathbf{-1}, \mathbf{1}, \mathbf{2}, 1, -1, -2, -1, 1, 2, 1, -1, -2, -1, 1, 2, 1, -1, \dots\} \\ c_7(n) &: \{6, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{-1}, \mathbf{6}, -1, -1, -1, -1, -1, -1, 6, -1, -1, -1, -1, -1, -1, \dots\} \\ c_8(n) &: \{4, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{-4}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{4}, 0, 0, 0, -4, 0, 0, 0, 4, 0, 0, 0, -4, \dots\} \\ c_9(n) &: \{6, \mathbf{0}, \mathbf{0}, \mathbf{-3}, \mathbf{0}, \mathbf{0}, \mathbf{-3}, \mathbf{0}, \mathbf{0}, \mathbf{6}, 0, 0, -3, 0, 0, -3, 0, 0, 6, 0, 0, \dots\} \\ c_{10}(n) &: \{4, \mathbf{1}, \mathbf{-1}, \mathbf{1}, \mathbf{-1}, \mathbf{-4}, \mathbf{-1}, \mathbf{1}, \mathbf{-1}, \mathbf{1}, \mathbf{4}, 1, -1, 1, -1, -4, -1, 1, -1, 1, 4, \dots\} \\ c_{11}(n) &: \{10, \mathbf{-1}, \mathbf{10}, -1, -1, -1, -1, -1, -1, -1, -1, \dots\} \quad (11) \\ c_{12}(n) &: \{4, \mathbf{0}, \mathbf{2}, \mathbf{0}, \mathbf{-2}, \mathbf{0}, \mathbf{-4}, \mathbf{0}, \mathbf{-2}, \mathbf{0}, \mathbf{2}, \mathbf{0}, \mathbf{4}, 0, 2, 0, -2, 0, -4, 0, -2, \dots\} \\ c_{13}(n) &: \{12, \mathbf{-1}, \mathbf{12}, -1, -1, -1, -1, -1, -1, -1, \dots\} \\ c_{14}(n) &: \{6, \mathbf{1}, \mathbf{-1}, \mathbf{1}, \mathbf{-1}, \mathbf{1}, \mathbf{-1}, \mathbf{-6}, \mathbf{-1}, \mathbf{1}, \mathbf{-1}, \mathbf{1}, \mathbf{-1}, \mathbf{1}, \mathbf{6}, 1, -1, 1, -1, 1, -1, \dots\} \\ c_{15}(n) &: \{8, \mathbf{1}, \mathbf{1}, \mathbf{-2}, \mathbf{1}, \mathbf{-4}, \mathbf{-2}, \mathbf{1}, \mathbf{1}, \mathbf{-2}, \mathbf{-4}, \mathbf{1}, \mathbf{-2}, \mathbf{1}, \mathbf{1}, \mathbf{8}, 1, 1, -2, 1, -4, \dots\} \\ c_{16}(n) &: \{8, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{-8}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{8}, 0, 0, 0, 0\} \\ c_{17}(n) &: \{16, \mathbf{-1}, \mathbf{16}, -1, -1, -1\} \end{aligned}$$

Mathematica-Programm für obige Tabelle: (Man beachte, dass die Tabelle nicht numerisch, sondern rein durch symbolische Berechnungen erzeugt wurde...):

```
Clear[q]; Clear[n];
Column[Table[FullSimplify[Sum[If[GCD[a, q]==1, Exp[2*Pi*I*a*(n/q)], 0], {a, 1, q}]], {q, 1, 17}, {n, 0, 20}]]
```

Oder (falls n nur bis zur Periodizität läuft):

```
Column[Table[FullSimplify[Sum[If[GCD[a, q]==1, Exp[2*Pi*I*a*(n/q)], 0], {a, 1, q}]], {q, 1, 15}, {n, 0, q}]]
```

Folgende Darstellung mit reellen trigonometrischen Funktionen kann auf ganz \mathbb{R} erweitert werden (hier die $c_q(n)$ von $n = 1$ bis 17), Primzahl-Indices in Rot:

$$\begin{aligned}
& \text{Cos}[2n\pi] \\
& \text{Cos}[n\pi] \\
& \text{Cos}\left[\frac{2n\pi}{3}\right] + \text{Cos}\left[\frac{4n\pi}{3}\right] \\
& \text{Cos}\left[\frac{n\pi}{2}\right] + \text{Cos}\left[\frac{3n\pi}{2}\right] \\
& \text{Cos}\left[\frac{2n\pi}{5}\right] + \text{Cos}\left[\frac{4n\pi}{5}\right] + \text{Cos}\left[\frac{6n\pi}{5}\right] + \text{Cos}\left[\frac{8n\pi}{5}\right] \\
& \text{Cos}\left[\frac{n\pi}{3}\right] + \text{Cos}\left[\frac{5n\pi}{3}\right] \\
& \text{Cos}\left[\frac{2n\pi}{7}\right] + \text{Cos}\left[\frac{4n\pi}{7}\right] + \text{Cos}\left[\frac{6n\pi}{7}\right] + \text{Cos}\left[\frac{8n\pi}{7}\right] + \text{Cos}\left[\frac{10n\pi}{7}\right] + \text{Cos}\left[\frac{12n\pi}{7}\right] \\
& \text{Cos}\left[\frac{n\pi}{4}\right] + \text{Cos}\left[\frac{3n\pi}{4}\right] + \text{Cos}\left[\frac{5n\pi}{4}\right] + \text{Cos}\left[\frac{7n\pi}{4}\right] \\
& \text{Cos}\left[\frac{2n\pi}{9}\right] + \text{Cos}\left[\frac{4n\pi}{9}\right] + \text{Cos}\left[\frac{8n\pi}{9}\right] + \text{Cos}\left[\frac{10n\pi}{9}\right] + \text{Cos}\left[\frac{14n\pi}{9}\right] + \text{Cos}\left[\frac{16n\pi}{9}\right] \\
& \text{Cos}\left[\frac{n\pi}{5}\right] + \text{Cos}\left[\frac{3n\pi}{5}\right] + \text{Cos}\left[\frac{7n\pi}{5}\right] + \text{Cos}\left[\frac{9n\pi}{5}\right] \\
& \text{Cos}\left[\frac{2n\pi}{11}\right] + \text{Cos}\left[\frac{4n\pi}{11}\right] + \text{Cos}\left[\frac{6n\pi}{11}\right] + \text{Cos}\left[\frac{8n\pi}{11}\right] + \text{Cos}\left[\frac{10n\pi}{11}\right] + \\
& \text{Cos}\left[\frac{12n\pi}{11}\right] + \text{Cos}\left[\frac{14n\pi}{11}\right] + \text{Cos}\left[\frac{16n\pi}{11}\right] + \text{Cos}\left[\frac{18n\pi}{11}\right] + \text{Cos}\left[\frac{20n\pi}{11}\right] \\
& \text{Cos}\left[\frac{n\pi}{6}\right] + \text{Cos}\left[\frac{5n\pi}{6}\right] + \text{Cos}\left[\frac{7n\pi}{6}\right] + \text{Cos}\left[\frac{11n\pi}{6}\right] \\
& \text{Cos}\left[\frac{2n\pi}{13}\right] + \text{Cos}\left[\frac{4n\pi}{13}\right] + \text{Cos}\left[\frac{6n\pi}{13}\right] + \text{Cos}\left[\frac{8n\pi}{13}\right] + \text{Cos}\left[\frac{10n\pi}{13}\right] + \text{Cos}\left[\frac{12n\pi}{13}\right] + \\
& \text{Cos}\left[\frac{14n\pi}{13}\right] + \text{Cos}\left[\frac{16n\pi}{13}\right] + \text{Cos}\left[\frac{18n\pi}{13}\right] + \text{Cos}\left[\frac{20n\pi}{13}\right] + \text{Cos}\left[\frac{22n\pi}{13}\right] + \text{Cos}\left[\frac{24n\pi}{13}\right] \\
& \text{Cos}\left[\frac{n\pi}{7}\right] + \text{Cos}\left[\frac{3n\pi}{7}\right] + \text{Cos}\left[\frac{5n\pi}{7}\right] + \text{Cos}\left[\frac{9n\pi}{7}\right] + \text{Cos}\left[\frac{11n\pi}{7}\right] + \text{Cos}\left[\frac{13n\pi}{7}\right] \\
& \text{Cos}\left[\frac{2n\pi}{15}\right] + \text{Cos}\left[\frac{4n\pi}{15}\right] + \text{Cos}\left[\frac{8n\pi}{15}\right] + \text{Cos}\left[\frac{14n\pi}{15}\right] + \text{Cos}\left[\frac{16n\pi}{15}\right] + \text{Cos}\left[\frac{22n\pi}{15}\right] + \text{Cos}\left[\frac{26n\pi}{15}\right] + \text{Cos}\left[\frac{28n\pi}{15}\right] \\
& \text{Cos}\left[\frac{n\pi}{8}\right] + \text{Cos}\left[\frac{3n\pi}{8}\right] + \text{Cos}\left[\frac{5n\pi}{8}\right] + \text{Cos}\left[\frac{7n\pi}{8}\right] + \text{Cos}\left[\frac{9n\pi}{8}\right] + \text{Cos}\left[\frac{11n\pi}{8}\right] + \text{Cos}\left[\frac{13n\pi}{8}\right] + \text{Cos}\left[\frac{15n\pi}{8}\right] \\
& \text{Cos}\left[\frac{2n\pi}{17}\right] + \text{Cos}\left[\frac{4n\pi}{17}\right] + \text{Cos}\left[\frac{6n\pi}{17}\right] + \text{Cos}\left[\frac{8n\pi}{17}\right] + \text{Cos}\left[\frac{10n\pi}{17}\right] + \text{Cos}\left[\frac{12n\pi}{17}\right] + \text{Cos}\left[\frac{14n\pi}{17}\right] + \text{Cos}\left[\frac{16n\pi}{17}\right] + \\
& \text{Cos}\left[\frac{18n\pi}{17}\right] + \text{Cos}\left[\frac{20n\pi}{17}\right] + \text{Cos}\left[\frac{22n\pi}{17}\right] + \text{Cos}\left[\frac{24n\pi}{17}\right] + \text{Cos}\left[\frac{26n\pi}{17}\right] + \text{Cos}\left[\frac{28n\pi}{17}\right] + \text{Cos}\left[\frac{30n\pi}{17}\right] + \text{Cos}\left[\frac{32n\pi}{17}\right]
\end{aligned}$$

Tabelle: Ramanujan-Summen $c_q(n)$ mit Cosinus-Termen dargestellt. Diese Darstellung lässt sich für n von \mathbb{N} auf \mathbb{R} oder \mathbb{C} erweitern (siehe Abbildung unten).

Mathematica-Programm für obige Tabelle:

```

Clear[n];
Column[Table[FullSimplify[Sum[If[GCD[a, q]==1, Cos[2*Pi*a*(n/q)], 0], {a, 1, q}]], {q, 1, 17}]]

```

Für ganzzahlige n vereinfacht sich die Darstellung (jeweils die zweite Hälfte eines Terms ist genauso groß wie die erste Hälfte, Primzahl-Indices in rot):

$$\begin{aligned}
 & 1, \\
 & \cos[n\pi], \\
 & 2\cos\left[\frac{2n\pi}{3}\right], \\
 & 2\cos\left[\frac{n\pi}{2}\right], \\
 & 2\left(\cos\left[\frac{2n\pi}{5}\right] + \cos\left[\frac{4n\pi}{5}\right]\right), \\
 & \cos\left[\frac{n\pi}{3}\right], \\
 & 2\left(\cos\left[\frac{2n\pi}{7}\right] + \cos\left[\frac{4n\pi}{7}\right] + \cos\left[\frac{6n\pi}{7}\right]\right), \\
 & 2\left(\cos\left[\frac{n\pi}{4}\right] + \cos\left[\frac{3n\pi}{4}\right]\right), \\
 & 2\left(\cos\left[\frac{2n\pi}{9}\right] + \cos\left[\frac{4n\pi}{9}\right] + \cos\left[\frac{8n\pi}{9}\right]\right), \\
 & 2\left(\cos\left[\frac{n\pi}{5}\right] + \cos\left[\frac{3n\pi}{5}\right]\right), \\
 & 2\left(\cos\left[\frac{2n\pi}{11}\right] + \cos\left[\frac{4n\pi}{11}\right] + \cos\left[\frac{6n\pi}{11}\right] + \cos\left[\frac{8n\pi}{11}\right] + \cos\left[\frac{10n\pi}{11}\right]\right), \\
 & 2\left(\cos\left[\frac{n\pi}{6}\right] + \cos\left[\frac{5n\pi}{6}\right]\right), \\
 & 2\left(\cos\left[\frac{2n\pi}{13}\right] + \cos\left[\frac{4n\pi}{13}\right] + \cos\left[\frac{6n\pi}{13}\right] + \cos\left[\frac{8n\pi}{13}\right] + \cos\left[\frac{10n\pi}{13}\right] + \cos\left[\frac{12n\pi}{13}\right]\right) \\
 & 2\left(\cos\left[\frac{n\pi}{7}\right] + \cos\left[\frac{3n\pi}{7}\right] + \cos\left[\frac{5n\pi}{7}\right]\right), \\
 & 2\left(\cos\left[\frac{2n\pi}{15}\right] + \cos\left[\frac{4n\pi}{15}\right] + \cos\left[\frac{8n\pi}{15}\right] + \cos\left[\frac{14n\pi}{15}\right]\right), \\
 & 2\left(\cos\left[\frac{n\pi}{8}\right] + \cos\left[\frac{3n\pi}{8}\right] + \cos\left[\frac{5n\pi}{8}\right] + \cos\left[\frac{7n\pi}{8}\right]\right), \\
 & 2\left(\cos\left[\frac{2n\pi}{17}\right] + \cos\left[\frac{4n\pi}{17}\right] + \cos\left[\frac{6n\pi}{17}\right] + \cos\left[\frac{8n\pi}{17}\right] + \right. \\
 & \left. \cos\left[\frac{10n\pi}{17}\right] + \cos\left[\frac{12n\pi}{17}\right] + \cos\left[\frac{14n\pi}{17}\right] + \cos\left[\frac{16n\pi}{17}\right]\right)
 \end{aligned}$$

Tabelle: Ramanujan-Summen $c_q(n)$ mit Cosinus-Termen dargestellt, für ganzzahlige n .

Es gibt eine alternative Berechnungsmöglichkeit der Ramanujan-Summe mit der Möbius-Funktion $\mu(n)$ und Euler Totient-Funktion $\varphi(q)$:

$$c_q(n) = \mu\left(\frac{q}{(q, n)}\right) \frac{\varphi(q)}{\varphi\left(\frac{q}{(q, n)}\right)} \tag{112}$$

Mathematica:

```

nmax=14; cnq[q_, n_] := EulerPhi[q] * (MoebiusMu[q/GCD[q, n]] /
EulerPhi[q/GCD[q, n]]); Column[Table[cnq[q, n], {q, 1, nmax}, {n, 1, nmax}]]

```

Hier ein paar graphische Darstellungen von Ramanujan-Summen:

Mathematica-Programm: Bitte den Autor kontaktieren.

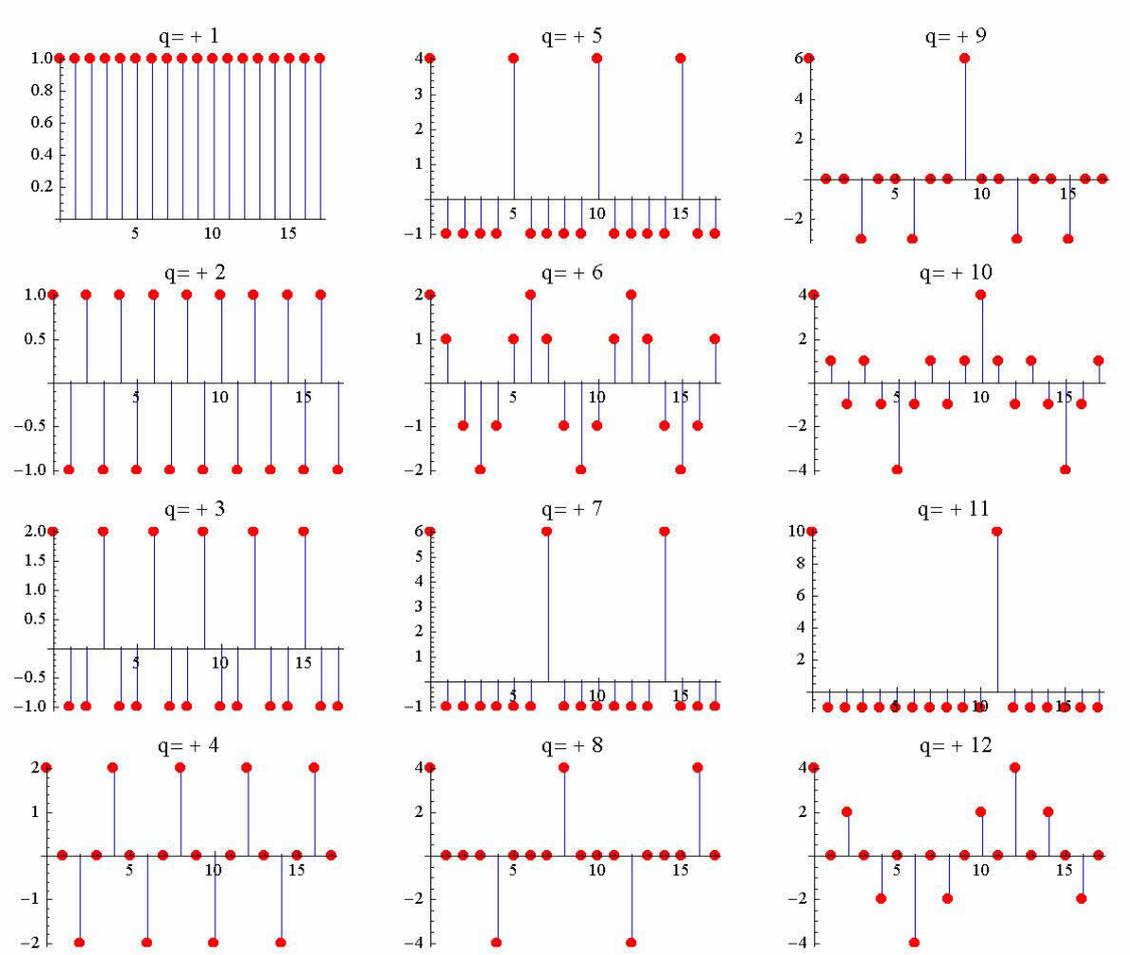


Abbildung 65: Ramanujan-Summen $c_q(n)$ von $q = 1$ bis 12 und n von 0 bis 17

Mathematica-Programm: Bitte den Autor kontaktieren.

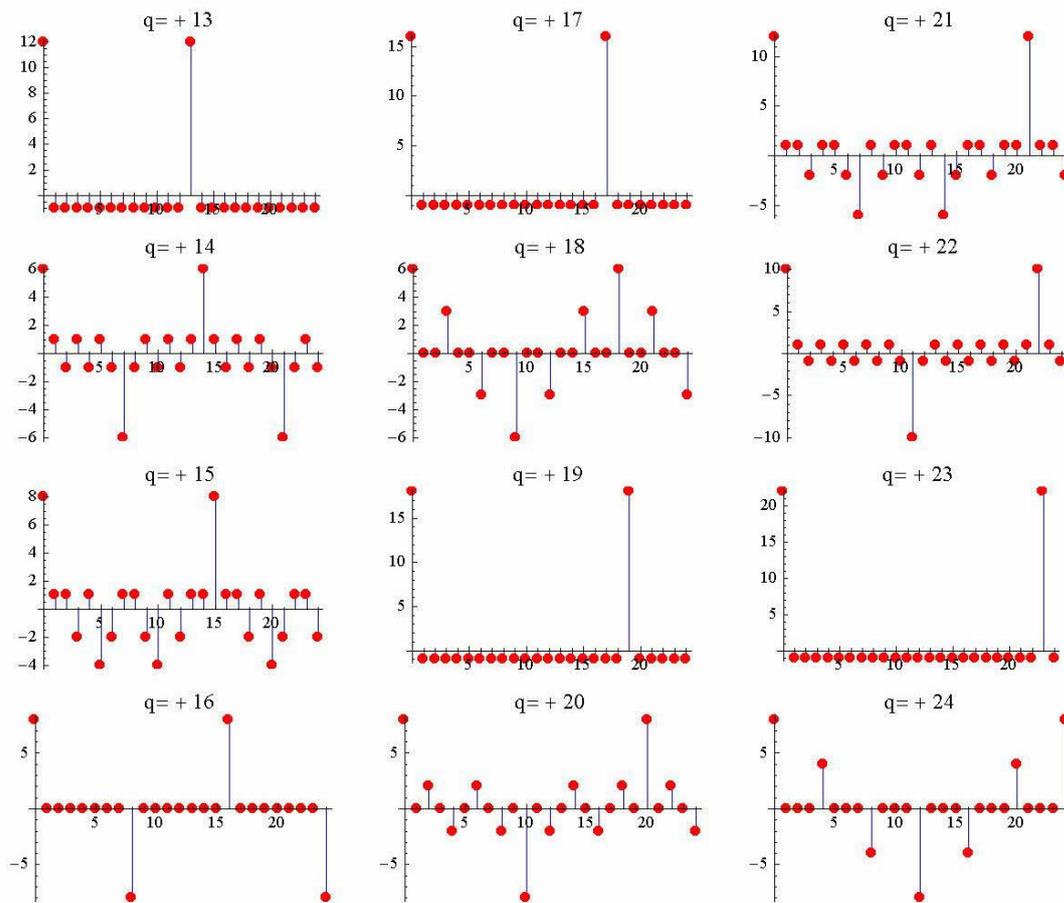


Abbildung 66: Ramanujan-Summen $c_q(n)$ von $q = 1$ bis 24 und n von 0 bis 24

7.9.2 EIGENSCHAFTEN

$c_q(n)$ besitzt eine ganze Reihe von bemerkenswerten Eigenschaften. Folgende lassen sich leicht an obiger Tabelle überprüfen:

$c_q(n)$ ist immer reell und ganzzahlig trotz der komplexen Definition.

$$c_q(n) = c_q(-n)$$

$$c_q(0) = \varphi(q)$$

$$c_q(1) = \mu(q)$$

$$c_{qr}(n) = c_q(n)c_r(n), \text{ wenn } (q, r) = 1 \text{ (Multiplikativität)}$$

$$c_q(n) = c_q((q, n))$$

$c_q(n)$ wird betragsmäßig nie größer als $\varphi(q)$ (bei festem q)

$c_q(n)$ wird betragsmäßig nie größer als n (bei festem n)

$c_q(n) = 0$, wenn die natürliche Zahl $\frac{q}{(q,n)}$ für eine Primzahl p von p^2 geteilt wird

$$c_q(q) = c_q(kq) = \varphi(q), k = 0, 1, 2, \dots$$

$$c_p(n) = \begin{cases} -1, & \text{wenn } p \nmid n \\ \varphi(p), & \text{wenn } p \mid n \end{cases}$$

$$c_{p^k}(n) = \begin{cases} 0, & \text{wenn } p^{k-1} \nmid n \\ -1p^{k-1}, & \text{wenn } p^{k-1} \mid n \text{ und } p^k \nmid n \\ \varphi(p^k), & \text{wenn } p^k \mid n \end{cases}$$

$$\sum_{n=a}^{a+q-1} c_q(n) = 0 \quad (\text{Die Summe über alle Terme einer Periode ergibt immer } 0!)$$

$$\frac{1}{m} \sum_{k=1}^m c_{m_1}(k) c_{m_2}(k) = \begin{cases} \varphi(m), & \text{wenn } m_1 = m_2 = m \\ 0, & \text{sonst} \end{cases} \quad (m = \text{kgV}(m_1, m_2))$$

(Orthogonalität)

7.9.3 ERWEITERUNG AUF \mathbb{R}

Lassen wir für n beliebige reelle Werte x zu, dann bekommen wir eine reelle Funktion, die von einem ganzzahligen Parameter q abhängt:

$$c_q(x) = \sum_{\substack{a=1 \\ (a,q)=1}}^q e^{2\pi i \frac{a}{q} x}, \quad (113)$$

Hier ein paar graphische Darstellungen von Ramanujan-Summenfunktionen:

Mathematica-Programm: Bitte den Autor kontaktieren.

Ramanujan-Summen

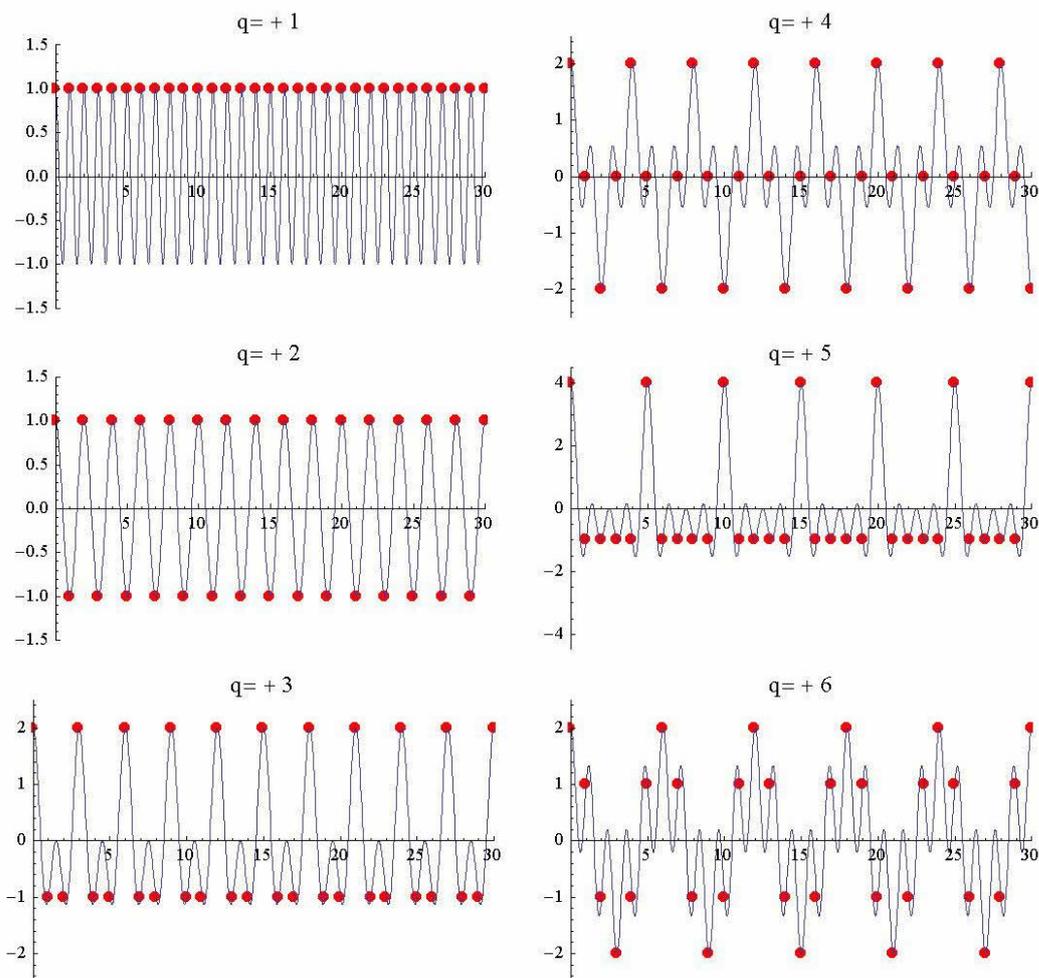


Abbildung 67: Ramanujan-Summen $c_q(x)$, analytisch fortgesetzt ($q = 1$ bis 6 und $x = 0$ bis 30)

Mathematica-Programm: Bitte den Autor kontaktieren.

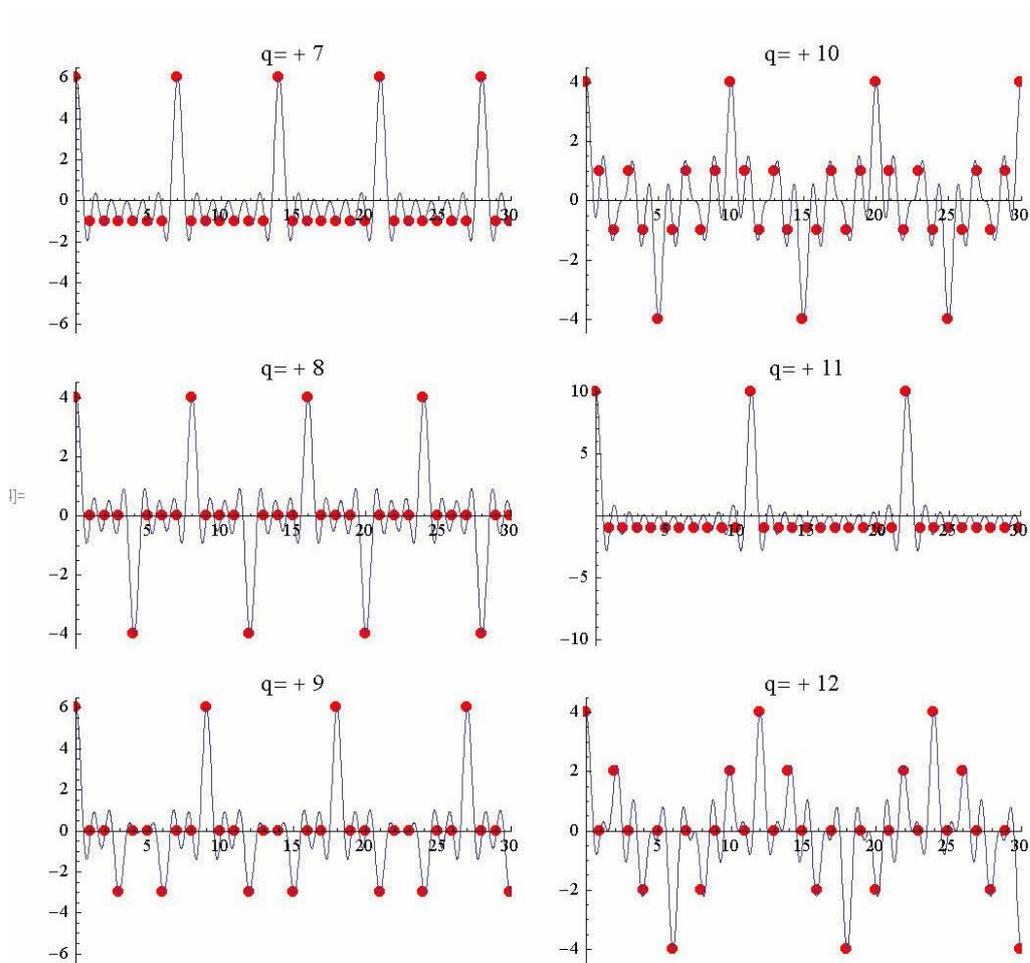


Abbildung 68: Ramanujan-Summen $c_q(x)$, analytisch fortgesetzt ($q = 7$ bis 12 und $x = 0$ bis 30)

8 FUNKTIONEN ZUR BERECHNUNG VON PRIMZAHLEN

Wir unterscheiden folgende Arten von Funktionen, mit denen sich Primzahlen berechnen lassen:

- Funktionen $f(n)$, die exakt p_n liefern
- Funktionen $f(n)$, die immer eine Primzahl ergeben
- Funktionen $f(n)$, deren positive Wertemenge (an ganzen Zahlen) Primzahlen ergeben
- Funktionen, die die Anzahl von Primzahlen bis zu einer vorgegebenen Grenze berechnen

8.1 FUNKTIONEN, DIE EXAKT ALLE PRIMZAHLEN LIEFERN

$$p_n = \left\lceil 1 - \log_2 \left(-\frac{1}{2} + \sum_{d|P_{n-1}} \frac{\mu(d)}{2^d - 1} \right) \right\rceil \quad (114)$$

Mit P_n als Primfakultät, dem Produkt aller Primzahlen $p_1 p_2 p_3 \dots p_n$ (auch $P\#$ geschrieben). Die Formel stammt von J.M. Gandhi (1971).

Die nächste Formel stammt von Williams (1964). Er braucht dafür entweder die Primzahl-Zählfunktion $\pi(n)$ oder die Funktion $F(j)$, die wie folgt definiert ist:

$$F(j) = \left\lceil \cos^2 \left(\pi \frac{(j-1)! + 1}{j} \right) \right\rceil$$

$$p_n = 1 + \sum_{m=1}^{2^n} \left\lceil \left[\frac{n}{\sum_{j=1}^m F(j)} \right]^{\frac{1}{n}} \right\rceil \quad (115)$$

bzw.

$$p_n = 1 + \sum_{m=1}^{2^n} \left\lceil \left[\frac{n}{1 + \pi(m)} \right]^{\frac{1}{n}} \right\rceil \quad (116)$$

$$p_n = \lceil 10^{2^n} \alpha \rceil - 10^{2^{n-1}} \lceil 10^{2^{n-1}} \alpha \rceil, \quad \text{mit } \alpha = \sum_{m=1}^{\infty} \frac{p_m}{10^{2^m}} \quad (117)$$

Alle diese Formeln sind zwar theoretisch sehr interessant, jedoch für die praktische Berechnung von Primzahlen ungeeignet.

8.2 FUNKTIONEN, DIE IMMER PRIMZAHLEN LIEFERN

$$p_n = \lfloor A^{3^n} \rfloor \quad (118)$$

A heisst ‚Mills Konstante‘ und hat näherungsweise den Wert 1.3063778838. Die von dieser Formel erzeugten ersten sechs Primzahlen (‚Mills‘-Primzahlen) lauten:

2, 11, 1361, 2521008887, 16022236204009818131831320183,
41131011492151048000305295379159531704861396235397599331359
49994882770404074832568499

Bisher wurden die ersten 11 ‚Mills‘-Primzahlen der Form $\lfloor A^{3^n} \rfloor$ berechnet, die größte mit über 20000 Ziffern. Die Mills-Konstante A wurde auf 6850 Stellen genau berechnet (Stand Nov. 2015).

Von Wright (1951) stammt die Formel:

$$p_n = \left\lfloor 2^{2^{2^{\dots^{2^\omega}}}} \right\rfloor \quad (\text{Potenzturm aus } n \text{ Exponenten, } \omega = 1.9287800) \quad (119)$$

Die ersten Primzahlen dieser Folge lauten:

3,13,16381,... (die vierte hat bereits über 5000 Stellen)

8.3 FUNKTIONEN, DEREN POSITIVE WERTEMENGEN DIE MENGE DER PRIMZAHLEN SIND

Seit dem Jahr 1976 ist ein Polynom 25-ten Grades mit 26 Variablen bekannt (Jones, Sato, Wada & Wies)⁴³, dessen positive Wertemenge mit der Menge der Primzahlen zusammenfällt, vorausgesetzt die 26 Variablen sind aus der Menge \mathbb{N}_0 .

Setzen wir:

$$\begin{aligned} C0 &= wz + h + j - q \\ C1 &= (gk + 2g + k + 1)(h + j) + h - z \\ C2 &= 2n + p + q + z - e \\ C3 &= 16(k + 1)^3(k + 2)(n + 1)^2 + 1 - f^2 \\ C4 &= e^3(e + 2)(a + 1)^2 + 1 - o^2 \\ C5 &= (a^2 - 1)y^2 + 1 - x^2 \\ C6 &= 16r^2y^4(a^2 - 1) + 1 - u^2 \\ C7 &= \left((a + u^2(u^2 - a))^2 - 1 \right) (n + 4dy)^2 + 1 - (x + cu)^2 \\ C8 &= n + l + v - y \\ C9 &= (a^2 - 1)l^2 + 1 - m^2 \end{aligned}$$

43

https://www.maa.org/sites/default/files/pdf/upload_library/22/Ford/JonesSatoWadaWiens.pdf

$$\begin{aligned}
 C10 &= ai + k + 1 - l - i \\
 C11 &= p + l(a - n - 1) + b(2an + 2a - n^2 - 2n - 2) - m \\
 C12 &= q + y(a - p - 1) + s(2ap + 2a - p^2 - 2p - 2) - x \\
 C13 &= z + pl(a - p) + t(2ap - p^2 - 1) - pm
 \end{aligned}$$

Dann existiert folgende Ungleichung, deren positive Lösungen exakt der Menge aller Primzahlen entspricht.

$$(k + 2)(1 - C0^2 - C1^2 - \dots - C13^2) > 0 \quad (120)$$

Man kann ein Mathematica-Programm schreiben, das diese Formel benutzt, um nach Primzahlen zu suchen (siehe Anhang). Man kann das Programm auch laufen lassen, damit es nach positiven Lösungen dieses Polynoms sucht. Man braucht jedoch sehr viel Geduld.... Selbst nach einer Woche Rechenzeit auf einem Quad-Core Pentium mit 3 Ghz wird leider noch keine Lösung gefunden!

8.4 REKURSIVE FORMELN

$$p_n = p_{n-1} + ggT(n, p_{n-1}), \quad \text{mit } p_1 = 7 \quad (121)$$

Diese Sequenz enthält nur Primzahlen oder Einsen.⁴⁴ Streichen wir die Einsen so erhalten wir:

```
{5, 3, 11, 3, 23, 3, 47, 3, 5, 3, 101, 3, 7, 11, 3, 13, 233, 3, 467, 3, 5, 3, 941, 3, 7, 1889, 3, 3779, 3, 7559, 3, 13, 15131, 3, 53, 3, 7, 30323, 3, 60647, 3, 5, 3, 101, 3, 121403, 3, 242807, 3, 5, 3, 19, 7, 5, 3, 47, 3, 37, 5, 3, 17, 3, 199, 53, 3, 29, 3, 486041, 3, 7, 421, 23, 3, 972533, 3, 577, 7}
```

```
f[1] = 7; f[n_] := f[n] = f[n - 1] + GCD[n, f[n - 1]];
DeleteCases[Differences[Table[f[n], {n, 10^6}]], 1]
```

⁴⁴ Eric S. Rowland, A simple prime-generating recurrence, Abstracts Amer. Math. Soc., 29 (No. 1, 2008), p. 50

8.5 FUNKTIONEN, DEREN ‚NULLSTELLEN‘ ODER MINIMA BEI PRIMZAHLEN LIEGEN

8.5.1 EINE VARIANTE DER DER \mathfrak{Z} -FUNKTION

In Kapitel 5.4 haben wir eine Funktion vorgestellt, deren Minima bei den Primzahlen liegen:

$$\mathfrak{Z}(s) = \prod_{n=1}^{\infty} \frac{1}{\left(1 - s^{-\frac{1}{2} - \rho_n i}\right)} \quad \text{mit } \rho_n: \text{Nullstellen von } \zeta(s)$$

Dabei haben wir ein unendliches Produkt, das über alle nicht-trivialen Nullstellen der Zeta-Funktion läuft. Zur praktischen Berechnung von Primzahlen ist diese Funktion jedoch ungeeignet, da sie sehr rechenaufwendig ist und es Probleme mit den Konvergenzeigenschaften des unendlichen Produktes gibt, da das unendliche Produkt nur lokal an den Primzahl-Stellen konvergiert. Bricht man das unendliche Produkt bei einem endlichen Wert N (z. B. 100 oder 1000) ab, so bekommt man tatsächlich einen Funktionsgraph, der bei den Primzahlen Minima nahe 0 besitzt. Die Genauigkeit der Lage der Minima ist umso höher, je mehr Produktterme berücksichtigt werden. Der Nachteil ist, dass die ‚primzahl-fernen‘ Bereiche divergieren und sehr groß werden können. Eine kleine Modifikation bringt noch einen ‚glatteren‘ Funktionsverlauf:

$$\mathfrak{Z}^*(x) = \ln \left(1 + \text{Abs} \prod_{n=1}^{[3x]} \frac{1}{\left(1 - x^{-\frac{1}{2} - \rho_n i}\right)} \right) \quad (122)$$

Hier ist eine Darstellung von $\mathfrak{Z}^*(x)$:

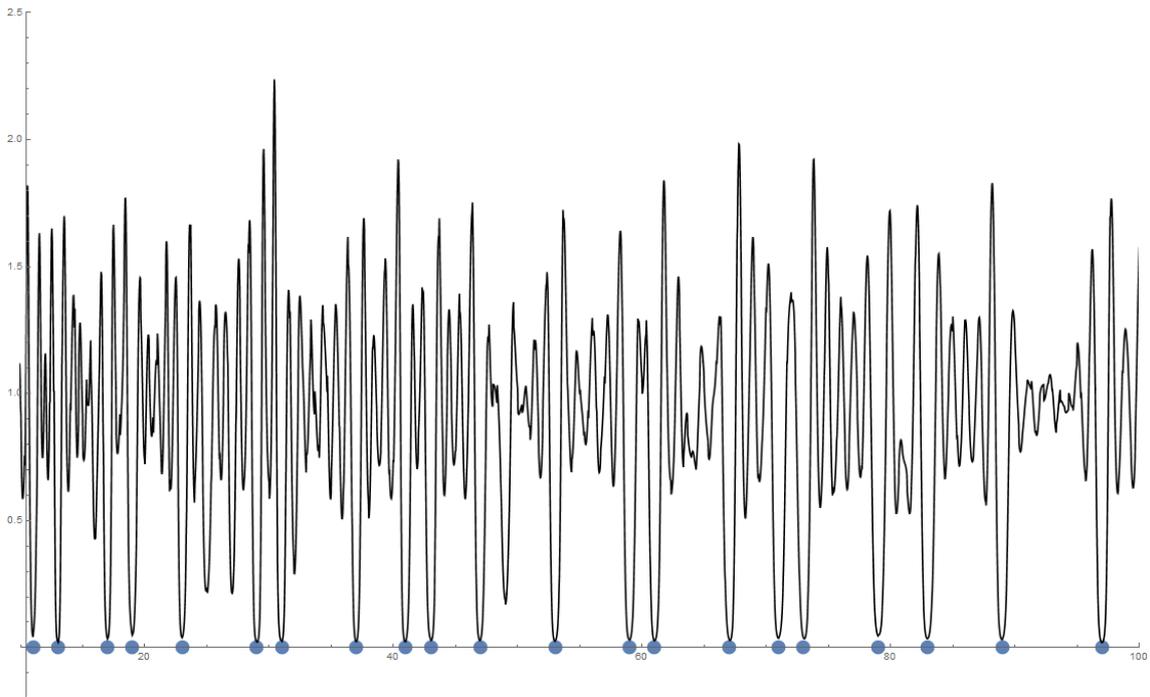


Abbildung 69: Variante der $\mathfrak{J}(x)$ nach (122) von 10 bis 100, mit Nullstellen (bzw. Minima) bei Primzahlen

Mathematica:

```

cterm[n_,p_]:=N[1/(1-p^(-ZetaZero[n]))];
myFunc[p_]:=Product[cterm[n,p],{n,1,3*p}]
xmin=10;xmax=100;Show[ListPlot[Table[{Prime[i],0},{i,5,PrimePi[xmax]}],
,PlotRange->{{xmin,xmax},{-0.2,2.5}},Plot[Log[1+Abs[myFunc[x]]],
{x,2,xmax},PlotStyle->Black,PlotRange->{{xmin,xmax},{-0.2,2.5}}]]
    
```

Hinweis: Auch bei Primzahl-Potenzen gibt es (wenn auch weniger ausgeprägte) Minima.

8.5.2 REED-JAMESON-FUNKTION

Im Kapitel 4.10.1.4 wurde die rekursive Reed-Jameson-Folge behandelt. Die Summe jeweiliger zu einem Index gehörender Modulus-Werte von negativen und positiven Indizes ergibt 0, wenn der Index n eine Primzahl ist.

Die Reed-Jameson Folge ist rekursiv definiert durch:

$$a_n = a_{n-5} + a_{n-2}, \quad \text{mit } a_0 = 5, a_1 = 0, a_2 = 2, a_3 = 0, a_4 = 2$$

Die inverse Reed-Jameson Folge ist rekursiv definiert durch:

$$b_n = b_{n-5} - b_{n-3}, \quad \text{mit } b_0 = 5, b_1 = 0, b_2 = 0, b_3 = -3, b_4 = 0$$

$$R_n = (a_n \bmod n) + (b_n \bmod n) \tag{123}$$

Die Vermutung, dass $R_n = 0$ ist (und nur dann) wenn n eine Primzahl ist, wurde 2018 von Peter Danzeglocke numerisch widerlegt.

Er fand Reed Jameson Pseudoprimzahlen im Bereich $> 10^{15}$ (s. Anhang). Bemerkenswert bleibt jedoch die Tatsache, dass die Methode im Bereich bis 10^{10} funktioniert (Stand: Dezember 2018).⁴⁵

Graph der Reed-Jameson-Funktion:

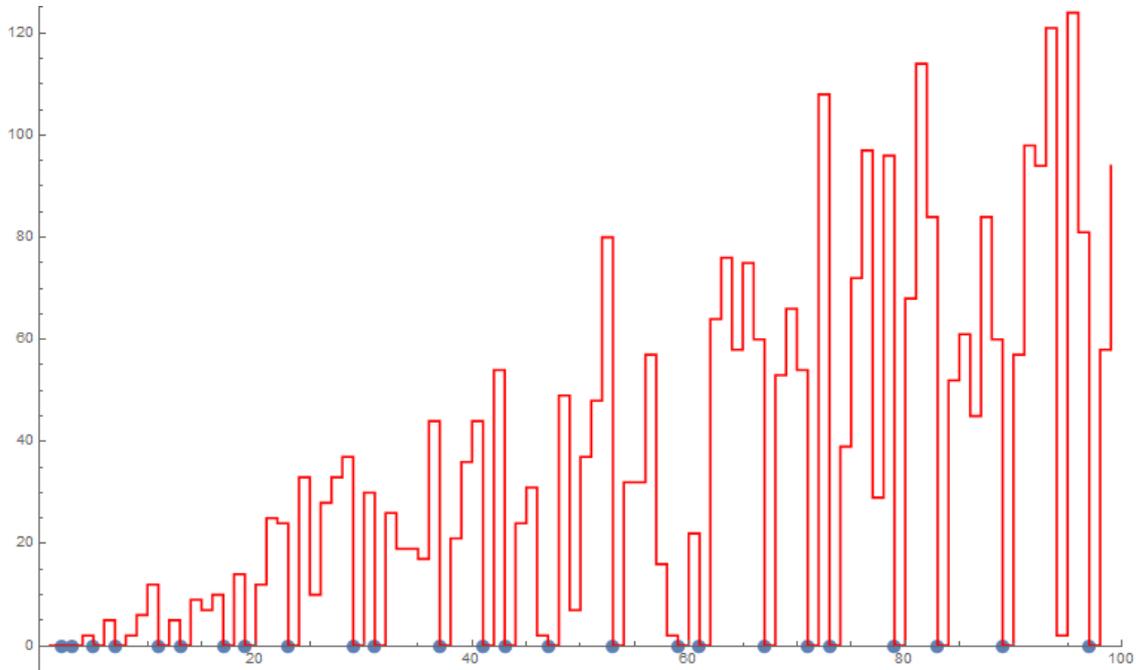


Abbildung 70: Reed-Jameson-Funktion von 0 bis 100

Mathematica-Programm: [Bitte den Autor kontaktieren.](#)

8.5.3 SONSTIGE ZAHLENTHEORETISCHE FUNKTIONEN MIT NULLSTELLEN BEI PRIMZAHLEN

Eulersche Totient-Funktion $\varphi(n)$:

$$f(n) = \varphi(n) - n + 1 \text{ (ist 0, wenn } n \text{ eine Primzahl ist)}$$

Teilersummenfunktion $\sigma_k(n)$:

$$f(n) = \sigma_1(n) - n - 1 \text{ (ist 0, wenn } n \text{ eine Primzahl ist)}$$

$$f(n) = \sigma_0(n) - 2 \text{ (ist 0, wenn } n \text{ eine Primzahl ist)}$$

⁴⁵ Viele linear rekursiv konstruierbare Folgen enthalten unendlich viele Pseudoprimzahlen.

8.6 FORMELN ZUR BERECHNUNG DER ANZAHL VON PRIMZAHLEN

Wir betrachten die sogenannte Primzahlzählfunktion $\pi(n)$:

$$\pi: \mathbb{N} \rightarrow \mathbb{N}, n \mapsto \pi(n): \quad \pi(n) = |\{p \in \mathbb{P} \mid p \leq n\}|$$

Dabei ist \mathbb{P} die Menge der Primzahlen und $|\dots|$ bedeutet die Anzahl der Elemente der Menge. Die Pi-Funktion wird meist auf den Bereich der reellen Zahlen erweitert: $\pi(x)$ mit $x \in \mathbb{R}$

Ein paar exakte Formeln:

Hardy und Wright (1979)

$$\pi(n) = -1 + \sum_{j=3}^n \left[(j-2)! - j \left\lfloor \frac{(j-2)!}{j} \right\rfloor \right], \quad (124)$$

mit der Floor Funktion $\lfloor \cdot \rfloor$

Williams (1964):

$$\pi(n) = -1 + \sum_{j=1}^n F(j), \quad \text{mit } F(j) = \left[\cos^2 \left(\pi \frac{(j-1)! + 1}{j} \right) \right] \quad (125)$$

Oder:

$$\pi(n) = -1 + \sum_{j=2}^n H(j), \quad \text{mit } H(j) = \frac{\sin^2 \left(\pi \frac{(j-1)!^2}{j} \right)}{\sin^2 \frac{\pi}{j}} \quad (126)$$

Eine ähnliche Formel stammt von Mini:

$$\pi(n) = \sum_{j=2}^n \left[\frac{(j-1)! + 1}{j} - \left\lfloor \frac{(j-1)!}{j} \right\rfloor \right] \quad (127)$$

Die einfachste Näherung ist bekannt unter dem Namen ‚Primzahlsatz‘ (Legendre und Gauß, 1798):

$$\pi(x) \approx \frac{x}{\ln(x) - 1.08366} \quad (128)$$

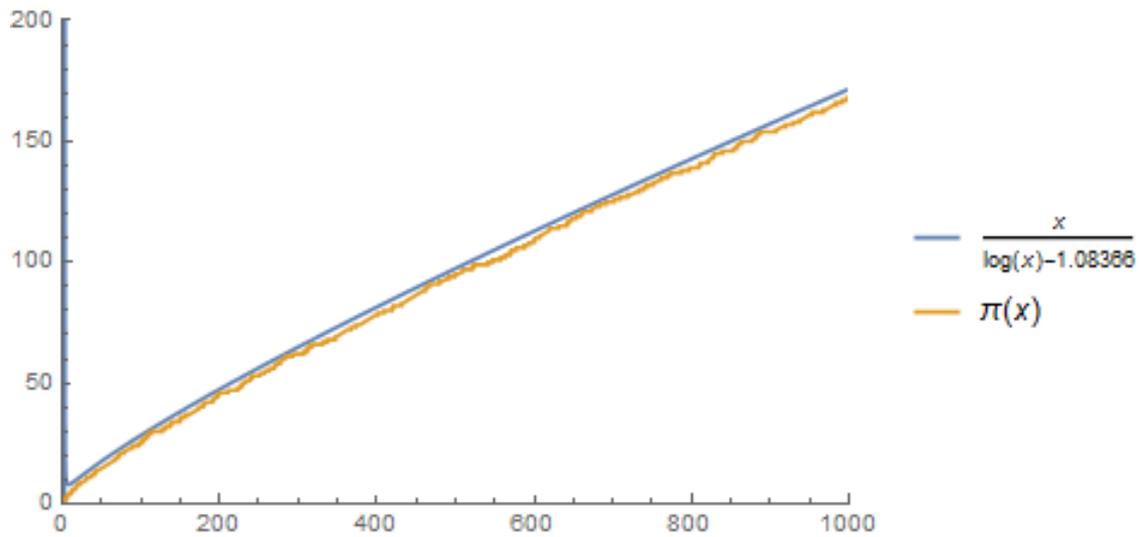


Abbildung 71: Vergleich $\pi(n)$ mit Gausscher Näherung

Mathematica:

```
Plot[{x/(Log[x]-1.08366), PrimePi[x]}, {x, 1, 1000}, PlotRange->{{0, 1000}, {0, 200}}, PlotPoints->200, PlotLegends->"Expressions"]
```

Eine bessere Näherung von C. F. Gauß:

$$\pi(x) = Li(x) + O(\sqrt{x} \cdot \ln(x))$$

$$\text{wobei: } Li(x) = \int_2^x \frac{dt}{\ln(t)} \text{ (Integrallogarithmus)} \quad (129)$$

Formeln zur Berechnung der Anzahl von Primzahlen

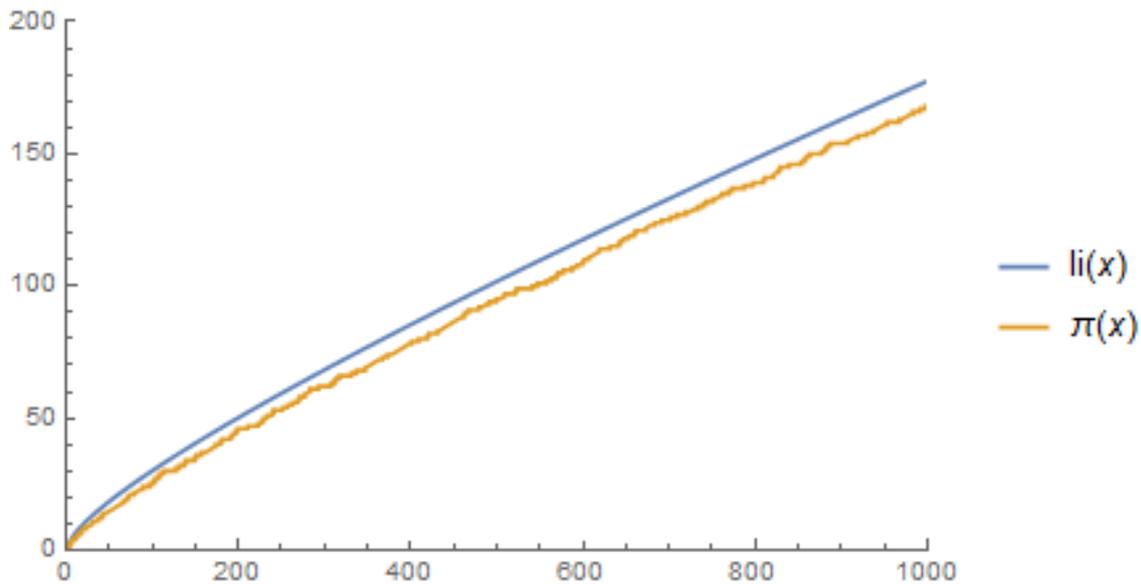


Abbildung 72: Vergleich $\pi(n)$ mit Integrallogarithmus, dargestellt von 1 bis 1000

Mathematica:

```
Plot[{LogIntegral[x], PrimePi[x]}, {x, 1, 1000}, PlotRange->{{0, 1000}, {0, 200}}, PlotPoints->200, PlotLegends->"Expressions"]
```

Es sieht aus, als wäre $Li(x)$ stets größer als $\pi(x)$. Für kleine x mag dies stimmen. Es ist jedoch bewiesen worden, dass für sehr große x $Li(x)$ auch ab und zu kleiner wird als $\pi(x)$. Im Jahr 1914 bewies L. E. Littlewood, dass die Differenz $\pi(x) - Li(x)$ unendlich oft das Vorzeichen wechselt. Inzwischen ist bewiesen worden, dass der erste Wechsel des Vorzeichens mindestens bei $1,39822 \cdot 10^{316}$ liegen muss (Richard Hudson, 2000). Der Punkt des ersten Vorzeichenwechsels kann jedoch nicht vor 10^{14} liegen (Kotnik, 2008).

Dies ist eine sehr erstaunliche Eigenschaft der Primzahl-Zähl-Funktion $\pi(x)$ bzw. der Funktion $Li(x)$. Zeigt sie doch, dass auch sehr, sehr große Zahlen neue, unerwartete Eigenschaften besitzen können. Oder anders ausgedrückt: Auch in astronomisch hohen Zahlenbereichen können völlig unerwartete Phänomene auftreten. Außerdem zeigt dies, dass wir der ‚numerischen Evidenz‘ nicht immer vertrauen können!

Eine noch bessere Näherung ist die Riemannsche Funktion $R(x)$:

$$\pi(x) \approx R(x) \quad (130)$$

Die beste asymptotische Formel lautet:

$$\pi(x) \approx R(x) - \frac{1}{\ln(x)} + \arctan\left(\frac{\pi}{\ln(x)}\right) \quad (131)$$

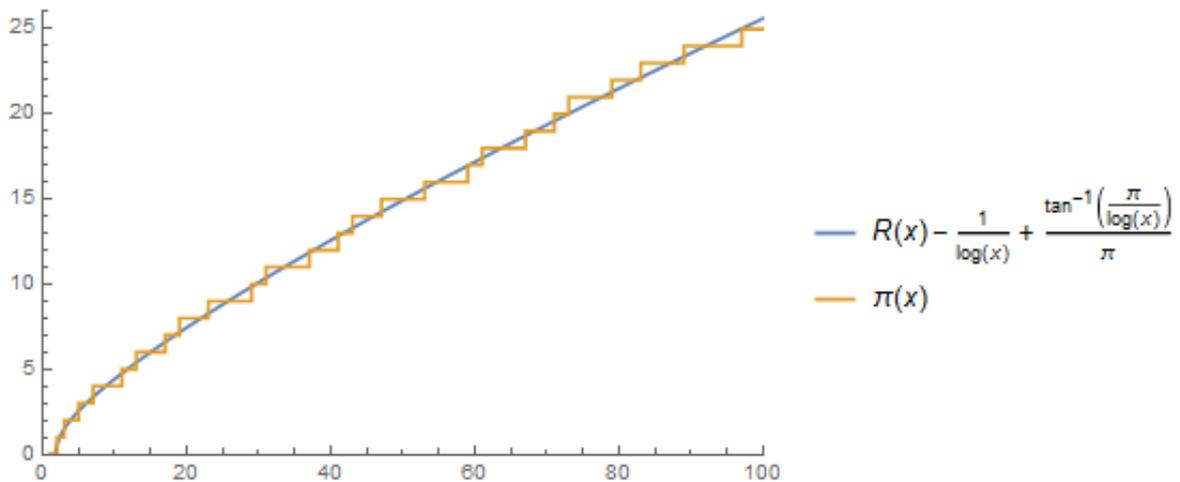


Abbildung 73: Vergleich $\pi(n)$ mit Riemann-Funktion $R(x)$, im Bereich von 0 bis 100

Mathematica: `Plot[{RiemannR[x]-1/Log[x]+ArcTan[Pi/Log[x]]/Pi,PrimePi[x]},{x,1,100},PlotRange->{{0,100},{0,26}},PlotPoints->200,PlotLegends->"Expressions"]`

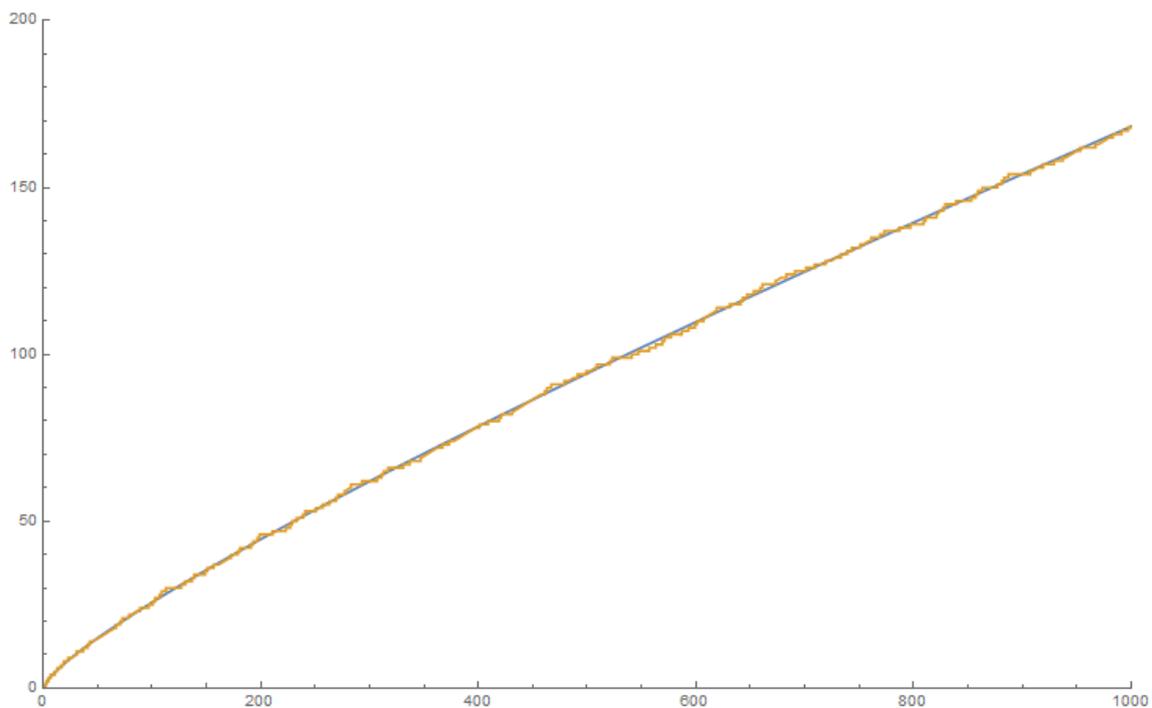


Abbildung 74: Vergleich $\pi(n)$ mit Riemann-Funktion $R(x)$, Bereich von 0 bis 1000

Formeln zur Berechnung der Anzahl von Primzahlen

Mathematica:

```
Plot[{RiemannR[x]-  
1/Log[x]+ArcTan[Pi/Log[x]]/Pi,PrimePi[x]},{x,1,1000},PlotRange-  
>{{0,1000},{0,200}},PlotPoints->200,PlotLegends->"Expressions"]
```

Und hier die geheimnisvolle exakte Formel von Riemann:

$$\pi(x) = R(x) - \sum_{\rho} R(x^{\rho}) \quad (132)$$

Mit der Riemann-Funktion $R(x)$.

$$\pi_0(x) = R(x) - \sum_{\rho} R(x^{\rho}) - \frac{1}{\ln(x)} + \frac{1}{\pi} \arctan\left(\frac{\pi}{\ln(x)}\right) \quad (133)$$

Die Riemann-Funktion ist eine sehr gute Approximation für $\pi(x)$ aber Riemanns Formel (132) ist noch viel genauer, sie berücksichtigt auch die kleinen lokalen Schwankungen von $\pi(x)$ und schmiegt sich schließlich (bei Berücksichtigung genügend vieler Nullstellen-Terme der Zetafunktion) an die exakte ganzzahlige Treppenfunktion der zahlentheoretischen Berechnung an. Hinweis: Die Aufsummierung über die Nullstellen-Terme der Zetafunktion muss in aufsteigender Reihenfolge zunehmender Werte von $\text{Im}(\rho)$ erfolgen, da die Summe nur bedingt konvergent ist. Die Berechnung von $R(x^{\rho})$ verlangt die Berechnung von $\text{Li}(x^{\rho})$ und ist nicht ganz so einfach, da hier der komplexe Logarithmus von x^{ρ} berechnet werden muss. Der wiederum ist mehrdeutig und nicht eindeutig definiert. Die Berechnung auf dem sogenannten ‚Hauptzweig‘ des komplexen Logarithmus⁴⁶ würde zu falschen Ergebnissen führen. Kurz gesagt liegt das Problem darin, dass für den komplexen Logarithmus die Gleichung $\ln(x^{\rho}) = \rho \ln(x)$ nicht immer gilt. Man kann das Problem jedoch umgehen, indem man statt $\text{Li}(x^{\rho})$ einfach $\text{Ei}(\rho \ln(x))$ verwendet, wobei $\text{Ei}(x)$ die komplexe Integraleponentialfunktion bezeichnet ($\text{Li}(x)$ ist der komplexe Integrallogarithmus). $\text{Ei}(x)$ ist mit $\text{Li}(x)$ eng verwandt, da $\mathbf{Li}(x) = \mathbf{Ei}(\ln x)$.

Hinweis: Die größten bekannten Werte von $\pi(x)$ wurden nicht zahlentheoretisch, sondern mit Methoden der analytischen Zahlentheorie gewonnen.

Die folgenden Graphen zeigen, wie die analytische Näherung sich an die exakte Treppenfunktion anschmiegt:

Mathematica-Programm: Bitte den Autor kontaktieren.

⁴⁶ Der Logarithmus des k -ten Zweiges ist definiert als $w = \ln|z| + i(\arg z + 2k\pi)$, $k \in \mathbb{Z}$. Für $k = 0$ haben wir den Hauptzweig der komplexen Logarithmusfunktion.

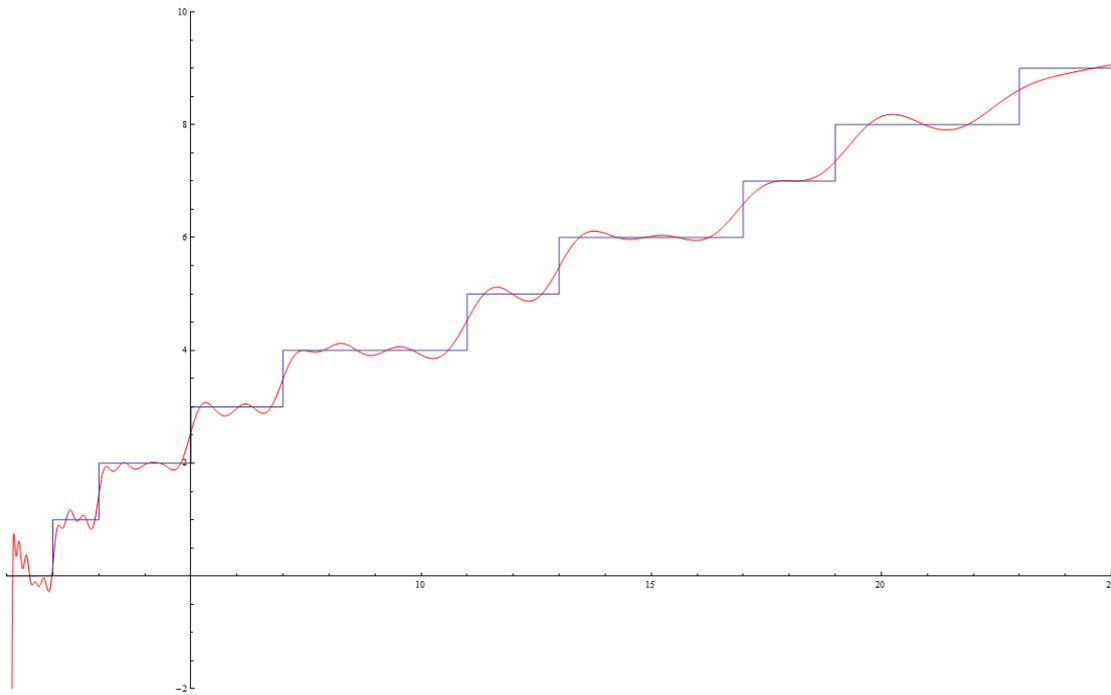


Abbildung 75: Riemanns exakte Formel für $\pi(x)$ (von $x = 1$ bis 25, mit Summe über die ersten 10 Nullstellenpaare der Zetafunktion)

Mathematica-Programm: Bitte den Autor kontaktieren.

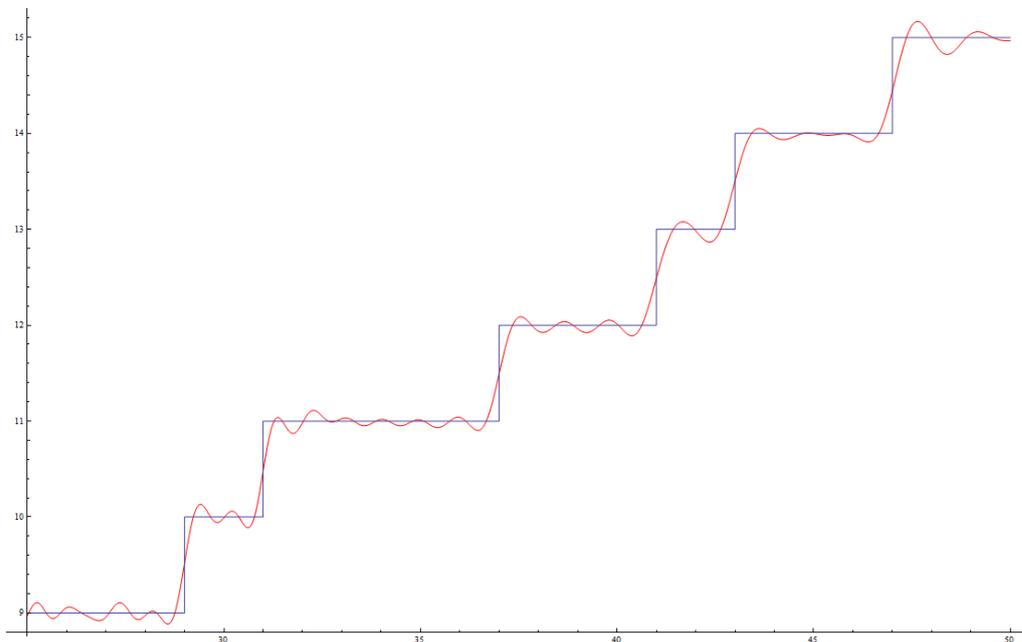


Abbildung 76: Riemanns exakte Formel für $\pi(x)$ (von $x = 25$ bis 50, mit Summe über die ersten 100 Nullstellenpaare der Zetafunktion)

Die Anzahl der zusammengesetzten Zahlen („Nicht-Primzahlen“) $\tilde{\pi}(n)$ bis zu einer gegebenen Grenze n ist einfach

Formeln zur Berechnung der n-ten Primzahl

$$\tilde{\pi}(n) = n - \pi(n)$$

Mathematica:
n-PrimePi[n]

Weitere Formeln für $\pi(x)$:

$$\pi(x) \approx \text{li}(x) - \frac{\sqrt{x}}{\ln x} \left(1 + 2 \sum_{\gamma} \frac{\sin(\gamma \ln x)}{\gamma} \right) \quad \text{mit } \gamma = \text{Im}(\rho) \quad (134)$$

Dabei sind ρ die komplexen Nullstellen der Zeta-Funktion.

8.7 FORMELN ZUR BERECHNUNG DER N-TEN PRIMZAHL

Wir wollen hier im Gegensatz zu Kapitel 8.1 analytische, näherungsweise gültige Lösungen betrachten.

Die analytische Berechnung der n-ten Primzahl gestaltet sich schwierig. Hierfür ist keine explizite, geschlossene Formel bekannt.

Die beste asymptotische Näherung ist

$$p_n = n \left(\ln n + \ln \ln n - 1 + \frac{(\ln \ln n - 2)}{\ln n} - \frac{(\ln \ln n)^2 - 6 \ln \ln n + 11}{2(\ln n)^2} \right) \quad (135)$$

```
Mathematica:
Table[Prime[n], {n, 1, 100}]
prime[n_] := Block[{logn=N[Log[n], 15], loglogn=N[Log[Log[n]], 15]}
,
n (logn+loglogn-1+(loglogn-2)/logn- (loglogn^2-
6loglogn+11)/(2logn^2))]
```

8.8 FORMELN ZUR BERECHNUNG DER N-TEN NICHT-PRIMZAHL

Die n-te Nicht-Primzahl kann mit folgendem Mathematica-Programm berechnet werden (Beispiel: Plot bis n = 1000)

```
Mathematica:
composite[n_Integer] := FixedPoint[n+PrimePi[#] &, n+PrimePi[n]]
ListLinePlot[Table[{k, composite[k]}, {k, 0, 1000, 10}], Filling->Axis]
```

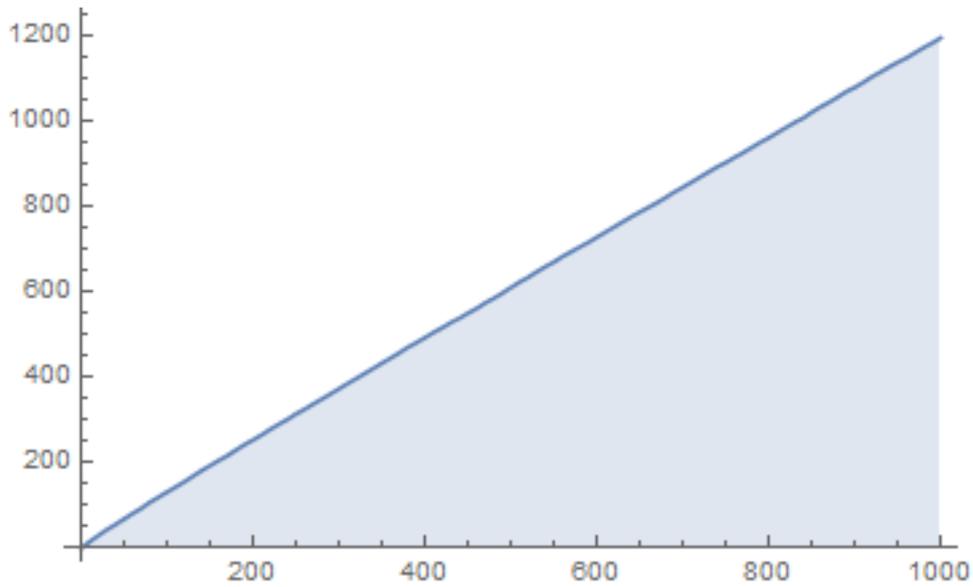


Abbildung 77: n-te zusammengesetzte Zahl („Nicht-Primzahl“)

Eine asymptotische Näherung für die n-te Nicht-Primzahl c_n lautet:

$$c_n = n \left(1 + \frac{1}{\ln n} + \frac{2}{\ln^2 n} + \frac{4}{\ln^3 n} + \frac{19}{2\ln^4 n} + \frac{181}{6\ln^5 n} + o\left(\frac{1}{\ln^5 n}\right) \right) \quad (136)$$

9 JETZT WIRD'S INTERESSANT: VIERDIMENSIONALE KUGELN UND PRIMZAHLEN

Was haben Kugeln, oder gar vierdimensionale Kugeln mit Primzahlen zu tun? Dieser Frage wollen wir in diesem Kapitel nachgehen. Im Prinzip geht es zunächst um die Frage, wie viele ganzzahlige Gitterpunkte im n -dimensionalen Raum den gleichen Abstand zum Ursprung haben (also auf der ‚Oberfläche‘ einer n -dimensionalen Kugel liegen). In der Mathematik hat sich dafür der Ausdruck ‚ n -Sphere‘ eingebürgert. So ist z.B. eine 1-Sphere die umgrenzende Kreislinie eines Kreises, eine 2-Sphere die gekrümmte 2-dimensionale Oberfläche einer Kugel. Eine 3-Sphere ist die Begrenzung einer 4-dimensionalen Kugel, also ein in die vierte Dimension hinein gekrümmter drei-dimensionaler Raum, den wir der Einfachheit halber gelegentlich auch als ‚Oberfläche‘ der vier-dimensionalen Kugel bezeichnen, manchmal wird auch das Wort ‚Glome‘ dafür verwendet. In diesem Kapitel suchen wir nach ganzzahligen Gitterpunkten (eines kartesischen Koordinatensystems), die auf n -Spheres im zwei, drei, oder vier-dimensionalen Raum sitzen.

Die Software Mathematica bietet hierfür drei mächtige Werkzeuge, mit denen dieses Problem angegangen werden kann:

`FindInstance`:

Findet Punkte, auf einer n -Sphere, hier z.B. auf einer 2-Sphere mit Radius \sqrt{n} :

```
FindInstance[x^2+y^2+z^2==n, {x, y, z}, Integers, numberOfInstances]
```

Da aufgrund der Symmetrieeigenschaften (die Symmetrie wird mit steigender Dimensionszahl immer höher) die Lösungen von `FindInstance` viele Permutationen oder achsen- bzw. punktsymmetrisch gespiegelte Lösungen enthalten, ist auch die folgende Funktion interessant, die wirklich nur den ‚Kern‘ der Lösungen liefert, also ohne die in negative Quadranten, Oktanten usw. gespiegelten oder durch Permutationen erzeugbaren Lösungen:

`PowersRepresentations`:

Findet alle (wirklich unterschiedlichen, ganzzahligen positiven) Lösungen der Gleichung $x^2 + y^2 + z^2 = n$

```
PowersRepresentations[n, 3, 2]
```

Und die Funktion `SquaresR`:

Liefert (nur) die Anzahl aller Lösungen von `FindInstance`.

Z.B.: `SquaresR[3, n]` gibt die Anzahl der Lösungen von `FindInstance[x^2+y^2+z^2==n, {x, y, z}, Integers, Infinity]`

Es wird sich zeigen, dass die Kugelpunkte auf den n -Spheres nicht zufällig verteilt sind, sondern dass sie im Gegenteil sehr schöne Strukturen bilden, die umso interessanter werden je höher die Dimensionalität der n -Spheres ist. Im Fall der 3-Spheres (‚Oberflächen‘ vier-dimensionaler Kugeln) kommt ein bemerkenswerter Zusammenhang zwischen der Anzahl der Kugelpunkte und den Primzahlen hinzu. Dieser Zusammenhang ist sehr einfach und sei hier vorweggenommen:

Wenn das Quadrat $n = rad^2$ des Radius einer 4-dimensionalen Kugel den Wert einer Primzahl p annimmt, dann (und nur dann) gilt folgender Zusammenhang:

$$rad^2 = n = \frac{r_4(n)}{8} - 1, \text{ wenn } n \in \mathbb{P} \quad (137)$$

Dieser Zusammenhang ist in der Literatur schon längst bekannt, da sich die Funktion $r_4(n)$ leicht aus der Sigma-Funktion $\sigma_1(n)$ berechnen läßt. Es wird allerdings nirgends in der einschlägigen Literatur über diesen schönen Zusammenhang mit den 3-Spheres (,Oberflächen') vier-dimensionaler Kugeln hingewiesen...

Eine Kuriosität am Rande: Die Zahl 12 spielt in der Folge $r_4(n)$ eine besondere Rolle, da sie die einzige Zahl ist, für die gilt:

$$n = \frac{r_4(n)}{8}, \text{ nur wenn } n = 12 \quad (138)$$

Da es schwierig ist, sich vierdimensionale Objekte vorzustellen, ist es immer eine gute Idee, mit dem Pendant in ein oder zwei niedrigeren Dimensionen zu beginnen. Also fangen wir mit den zweidimensionalen Kugeln (1-Spheres) an, die wir, wie die meisten Menschen als Kreise bezeichnen.

9.1 ZWEITE DIMENSION: KREISE UND GANZZAHLIGE GITTERPUNKTE

Wir suchen nach Gitterpunkten unseres ganzzahligen, rechtwinkligen Koordinatensystems, die entlang der Kreislinie liegen können (uns interessieren nicht die Gitterpunkte, die innerhalb des Kreises liegen können, sondern nur die Gitterpunkte, die genau auf der Kreislinie liegen. Die Anzahl dieser möglichen Gitterpunkte, die von der Kreislinie berührt werden, hängt natürlich sehr stark von Radius des Kreises ab. Stellen wir uns vor, wir vergrößern den Kreisradius kontinuierlich, dann wird die Kreislinie der Reihe nach durch die Gitterpunkte unseres Koordinatensystems laufen. Uns interessieren die Gitterpunkte, die exakt auf der Kreislinie liegen. Hier 6 Beispiele für $r^2 = 8$ bis 13:

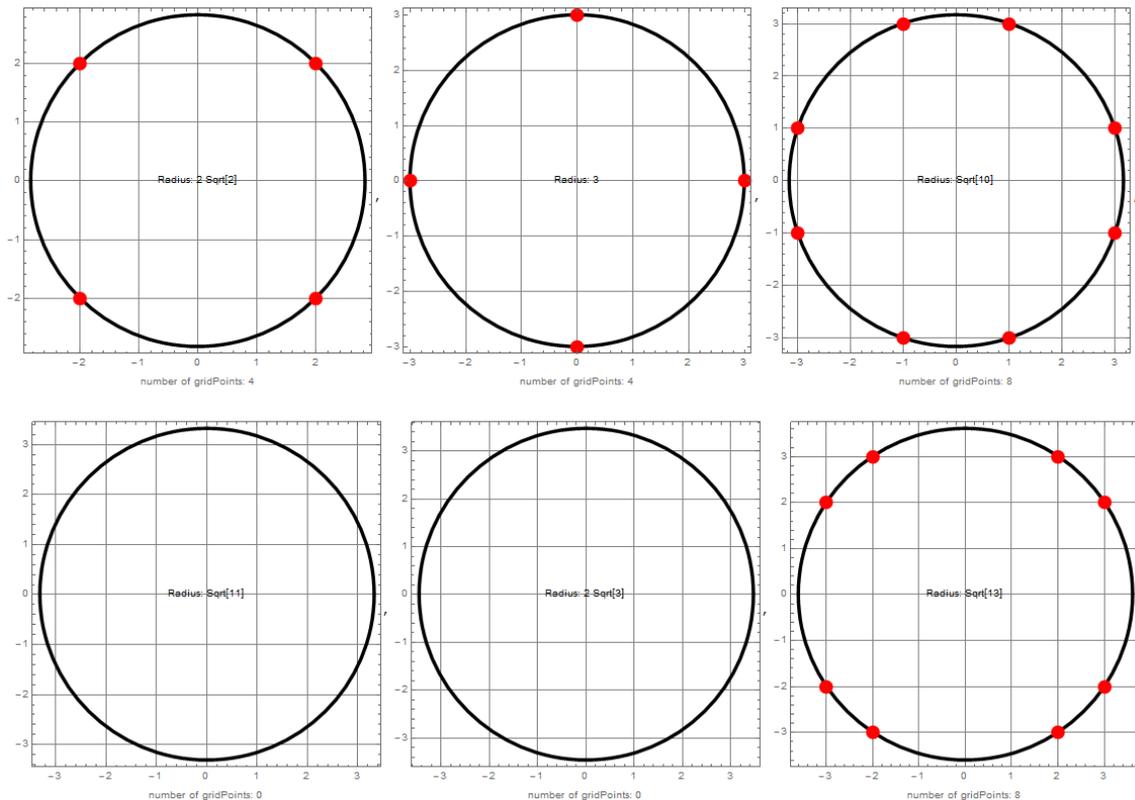


Abbildung 78: Gitterpunkte auf 1-Sphere, quadratischer Radius von 8 bis 13

Mathematica-Programm: Bitte den Autor kontaktieren.

Für einige Werte mit Radius r gibt es keine ganzzahligen Lösungen der Gleichung $x^2 + y^2 = r^2$ und deshalb auch keine entsprechende Gitterpunkte, die von der Kreislinie berührt werden. Das ‚Durchlaufen‘ der Kreislinie durch die zwei-dimensionalen Gitterpunkte kann in einer Animation betrachtet werden (als Video auf der beiliegenden CD) oder als Mathematica-Animation (siehe Anhang Gitterpunkte auf n-Spheres (n-dimensionalen Kugeln)).

Die Funktion, die die Anzahl von ganzzahligen Gitterpunkten auf einer Kreislinie im 2-dimensionalen Raum berechnet, wird in der Literatur mit $r_2(n)$ bezeichnet. Die Funktionswerte sind alle durch 4 teilbar. Die ersten 100 Werte lauten:

```
{4, 4, 0, 4, 8, 0, 0, 4, 4, 8, 0, 0, 8, 0, 0, 4, 8, 4, 0, 8, 0, 0, 0, 0, 12, 8, 0, 0, 8, 0, 0, 4, 0, 8,
0, 4, 8, 0, 0, 8, 8, 0, 0, 0, 8, 0, 0, 0, 4, 12, 0, 8, 8, 0, 0, 0, 0, 8, 0, 0, 8, 0, 0, 4, 16, 0, 0, 8,
0, 0, 0, 4, 8, 8, 0, 0, 0, 0, 0, 8, 4, 8, 0, 0, 16, 0, 0, 0, 8, 8, 0, 0, 0, 0, 0, 0, 8, 4, 0, 12}
Mathematica:
SquaresR[2, Range[100]]
```

Die 8 Lösungen für $r^2 = 5$ lauten z. B.:

```
{{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}, {2, -1}, {2, 1}}
Mathematica:
FindInstance[x^2+y^2==5, {x, y}, Integers, 8]
```

Diese Lösungen gehen gegenseitig durch Permutationen oder symmetrische Spiegelungen auseinander hervor. Die Anzahl der wirklich unterschiedlichen Lösungen ist in diesem Fall 1:

```
{{1, 2}}
Mathematica:
PowersRepresentations[5, 2, 2]
```

$r_2(n)$ ist die Anzahl von Gitterpunkten im 2-dimensionalen Raum, die auf einer Kreislinie mit Radius \sqrt{n} liegen. Mit $r_2^*(n)$ bezeichnen wir die Anzahl der unterschiedlichen, positiven Gitterpunkte, so dass gilt: $0 \leq n_1 \leq n_2$ und $n_1^2 + n_2^2 = n$.

$r_2(n)$ hat für viele Werte von n den Wert 0. Das bedeutet, dass nicht jede natürliche Zahl als Summe von 2 Quadratzahlen geschrieben werden kann. Hier eine Liste der ersten Werte dieser ‚nicht darstellbaren‘ Zahlen:

```
{3, 6, 7, 11, 12, 14, 15, 19, 21, 22, 23, 24, 27, 28, 30, 31, 33, 35, 38, 39, 42, 43, 44, 46,
47, 48, 51, 54, 55, 56, 57, 59, 60, 62, 63, 66, 67, 69, 70, 71, 75, 76, 77, 78, 79, 83, 84,
86, 87, 88, 91, 92, 93, 94, 95, 96, 99, ...}
```

```
Mathematica:
Select[Range[199], Length[PowersRepresentations[#, 2, 2]] == 0 &]
```

Hier zwei Graphen von $r_2(n)$:

Zweite Dimension: Kreise und ganzzahlige Gitterpunkte

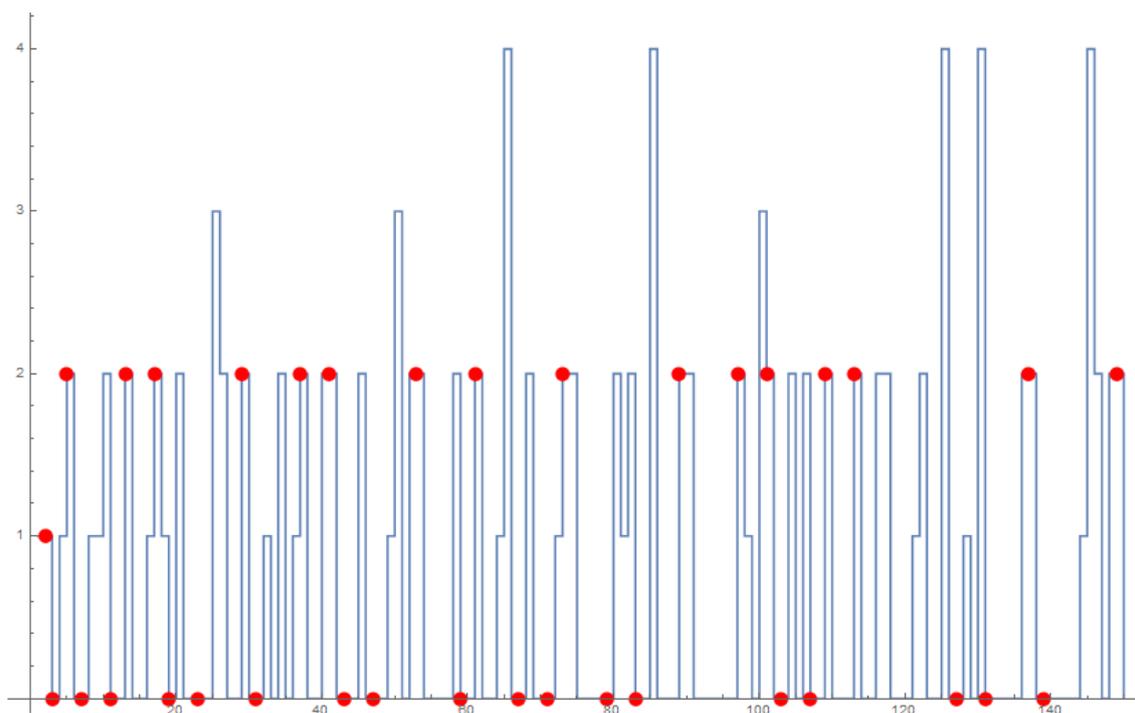


Abbildung 79: $r_2(n)$: Anzahl/4 der möglichen Darstellungen von n als Summe von 2 Quadraten. Mit Primzahlen (rote Punkte) ist kein Zusammenhang erkennbar.

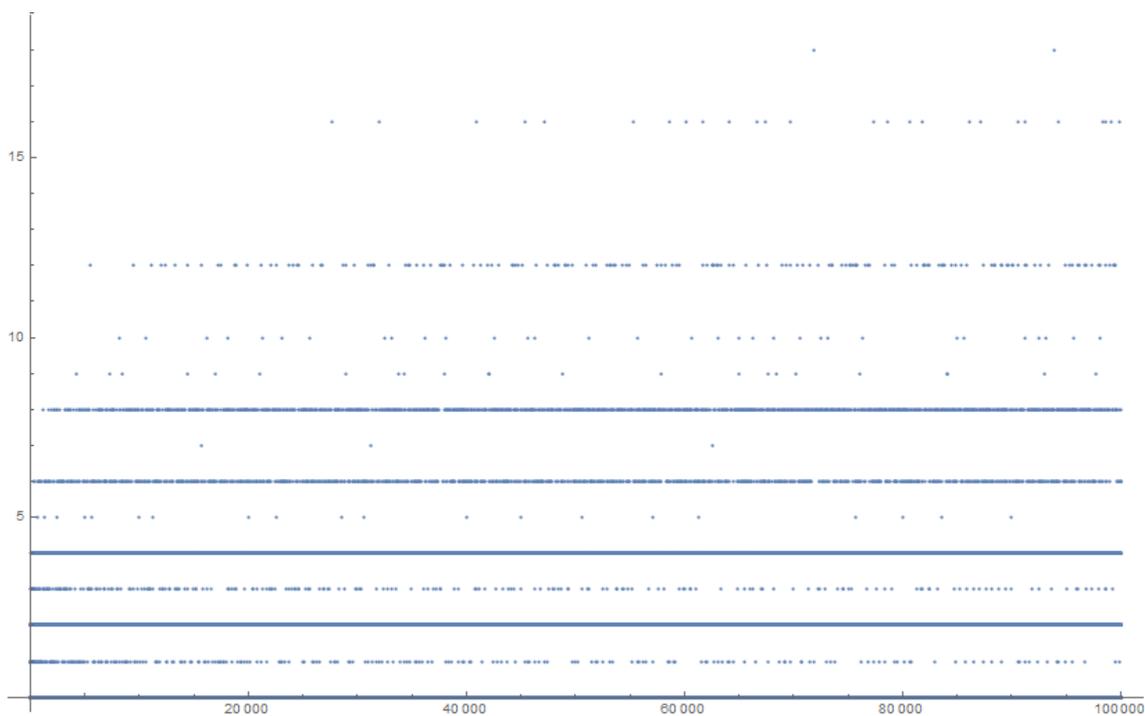


Abbildung 80: $r_2(n)$: Anzahl/4 der Darstellungen von n als Summe von 2 Quadraten (bis 100000)

Und hier ein paar Graphen von $r_2^*(n)$ für verschiedene n :

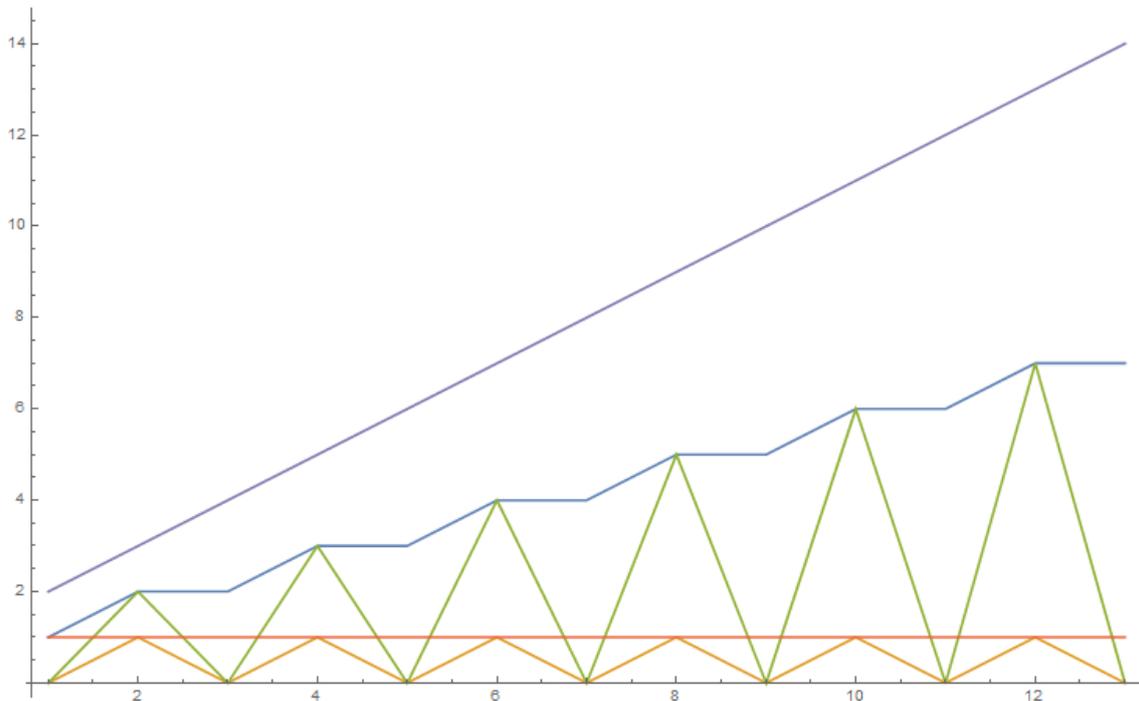


Abbildung 81: $r_2^*(b^n)$: Anzahl unterschiedlicher Darstellungen von b^n als Summe von zwei Quadraten

Mathematica-Programm (Abbildung 79) :

```
Show[ListLinePlot[Table[{n, SquaresR[2, n]/4}, {n, 1, 150}],
InterpolationOrder->0], ListPlot[Table[{Prime[n],
SquaresR[2, Prime[n]]/4}, {n, 1, PrimePi[150]}], PlotStyle->Red]
]
```

Mathematica (Abbildung 80) :

```
Show[ListPlot[Table[{n, SquaresR[2, n]/4}, {n, 1, 100000}], PlotRange->Full]
]
```

Mathematica (Abbildung 81) :

```
ListLinePlot[{Table[Length[PowersRepresentations[10^i, 2, 2]], {i, 1, 13}],
Table[Length[PowersRepresentations[14^i, 2, 2]], {i, 1, 13}],
Table[Length[PowersRepresentations[15^i, 2, 2]], {i, 1, 13}],
Table[Length[PowersRepresentations[16^i, 2, 2]], {i, 1, 13}],
Table[Length[PowersRepresentations[25^i, 2, 2]], {i, 1, 13}]
}, PlotLegends->Automatic, PlotRange->All]
```

Hinweis: Ganzzahlige Lösungen von $x^2 + y^2 = r^2$ (auch r ganzzahlig) werden in der Literatur als ‚pythagoreische Tripel‘ bezeichnet. Diese entsprechen den Gitterpunkten auf einer Kreislinie mit einem ganzzahligen Radius r .

9.1.1 FORMELN UND EIGENSCHAFTEN

Wir beschränken uns hier auf die Funktion $r_2(n)$, die die Anzahl von Gitterpunkten im zwei-dimensionalen Raum, die auf einer Kreislinie mit Radius \sqrt{n} liegen, berechnet. Für die Funktion $r_2^*(n)$, die die Anzahl der unterschiedlichen, positiven Gitterpunkte

Zweite Dimension: Kreise und ganzzahlige Gitterpunkte

berechnet, so dass gilt: $0 \leq n_1 \leq n_2$ und $n_1^2 + n_2^2 = n$ sei hier verwiesen auf z.B. (s. Fußnote).⁴⁷

Die erzeugende Funktion von $r_2(n)$ ist die elliptische Jacobi $\vartheta_3(n)$ -Funktion im Quadrat:

$$\sum_{n=0}^{\infty} r_2(n)x^n = \vartheta_3^2(x) = 1 + 4x + 4x^2 + 4x^4 + 8x^5 + \dots \quad (139)$$

Explizite Formeln:

$$r_2(n) = 4[d_1 - d_3], \quad d_k: \text{Anzahl Teiler von } n \text{ der Form } 4m + k \quad (140)$$

⁴⁷ <http://mathworld.wolfram.com/SumofSquaresFunction.html>

9.2 DRITTE DIMENSION: KUGELN UND GITTERPUNKTE

Auch hier suchen wir nach Gitterpunkten im ganzzahligen, rechtwinkligen Koordinatensystem, die auf der Oberfläche einer Kugel liegen können. Die Anzahl dieser möglichen Gitterpunkte, die von der Kugeloberfläche berührt werden, hängt auch hier sehr stark von Radius der Kugel ab. Stellen wir uns vor, wir vergrößern den Kugelradius kontinuierlich, dann wird die Kugeloberfläche der Reihe nach durch die Gitterpunkte unseres Koordinatensystems laufen. Uns interessieren die Gitterpunkte, die exakt auf der Kugeloberfläche liegen. Hier ein paar Beispiele:

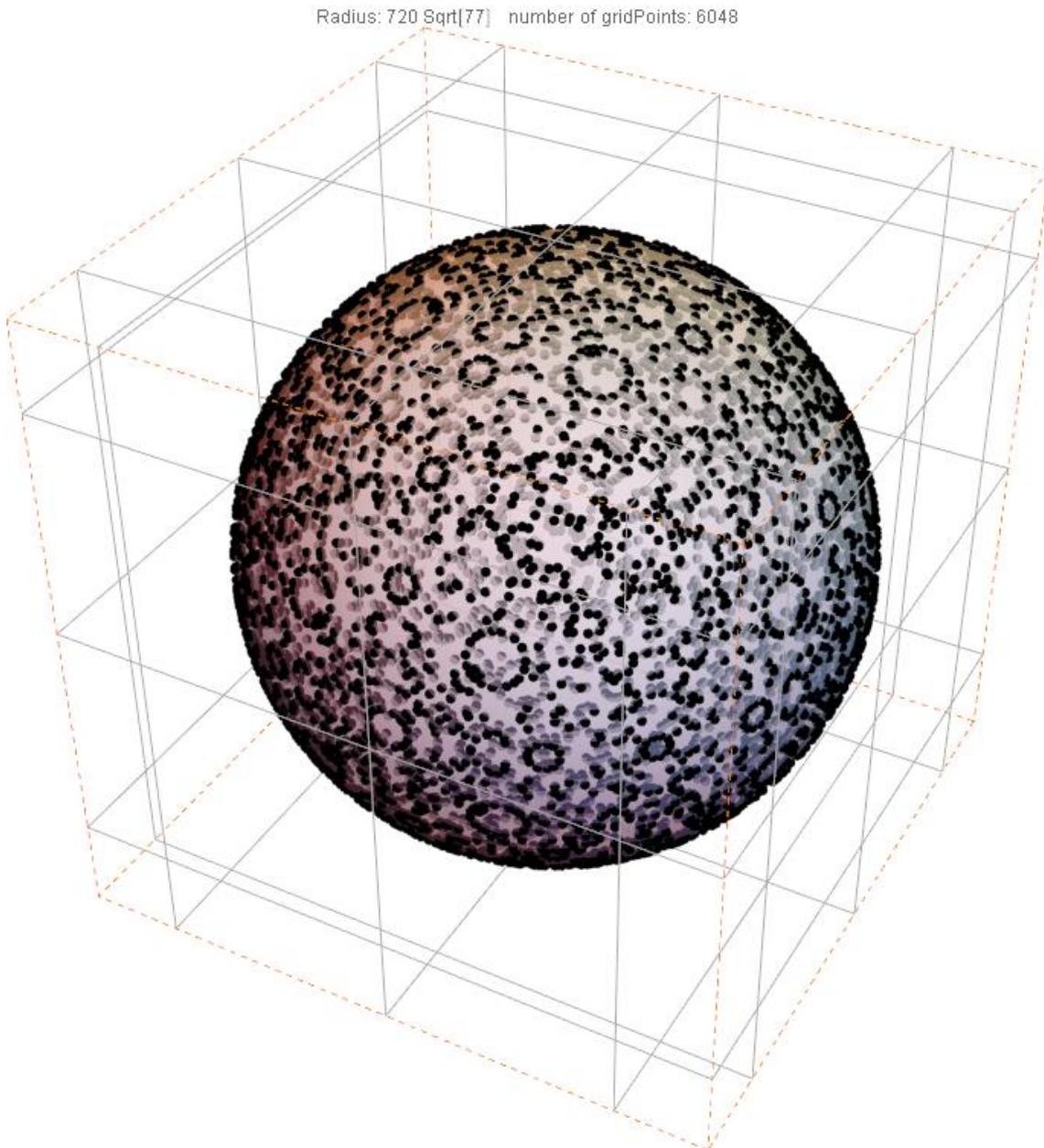


Abbildung 82: Ganzzahlige Gitterpunkte einer Kugel mit (quadrierten) Radius 11!

Mathematica-Programm: Bitte den Autor kontaktieren.

Dritte Dimension: Kugeln und Gitterpunkte

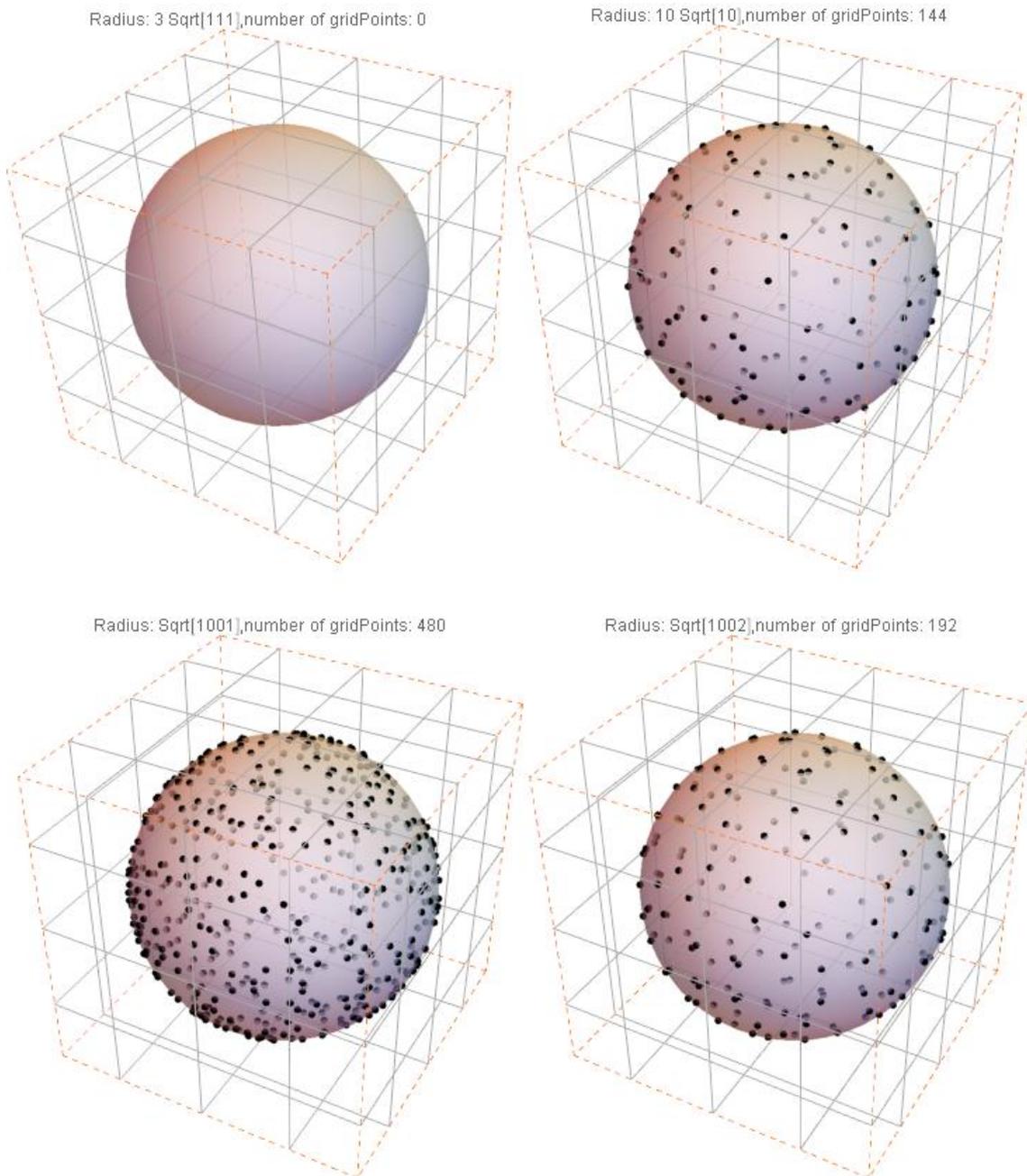


Abbildung 83: Ganzzahlige Gitterpunkte von Kugeln (quad. Radius 999-102).

Für einige Werte mit Radius r gibt es keine ganzzahlige Lösungen der Gleichung $x^2 + y^2 + z^2 = r^2$ und deshalb auch keine entsprechende Gitterpunkte, die von der Kugeloberfläche berührt werden. Manche Strukturen werden erst sichtbar, wenn man die Kugel von verschiedenen Blickwinkeln betrachtet. Dies kann in einer Animation betrachtet werden (als Video auf der beiliegenden CD) oder als Mathematica-Animation (siehe Anhang). Die Funktion, die die Anzahl von ganzzahligen Gitterpunkten auf der Oberfläche einer Kugel im drei-dimensionalen Raum berechnet, wird in der Literatur mit $r_3(n)$ bezeichnet. Mit $r_3^*(n)$ bezeichnen wir die Anzahl der unterschiedlichen, positiven Gitterpunkte, so dass gilt: $0 \leq n_1 \leq n_2 \leq n_3$ und $n_1^2 + n_2^2 + n_3^2 = n$.

Viele (jedoch nicht alle) Funktionswerte von $r_3(n)$ sind durch 6 teilbar. Die ersten 100 Werte lauten:

```
{6, 12, 8, 6, 24, 24, 0, 12, 30, 24, 24, 8, 24, 48, 0, 6, 48, 36, 24, 24, 48, 24, 0, 24, 30, 72,
, 32, 0, 72, 48, 0, 12, 48, 48, 48, 30, 24, 72, 0, 24, 96, 48, 24, 24, 72, 48, 0, 8, 54, 84, 48,
, 24, 72, 96, 0, 48, 48, 24, 72, 0, 72, 96, 0, 6, 96, 96, 24, 48, 96, 48, 0, 36, 48, 120, 56, 2
4, 96, 48, 0, 24, 102, 48, 72, 48, 48, 120, 0, 24, 144, 120, 48, 0, 48, 96, 0, 24, 48, 108, 7
2, 30}
```

Mathematica:

```
SquaresR[3, Range[100]]
```

Die 8 Lösungen für $r^2 = 3$ lauten z. B.:

```
{{-1, -1, -1}, {-1, -1, 1}, {-1, 1, -1}, {-1, 1, 1}, {1, -1, -1}, {1, -1, 1}, {1, 1, -1}, {1, 1, 1}}
```

Mathematica:

```
FindInstance[x^2+y^2+z^2==3, {x, y, z}, Integers, 8]
```

Diese Lösungen gehen gegenseitig durch Permutationen oder symmetrische Spiegelungen auseinander hervor. Die Anzahl der wirklich unterschiedlichen Lösungen $r_3^*(n)$ ist in diesem Fall 1: $\{\{1, 1, 1\}\}$

Mathematica:

```
PowersRepresentations[3, 3, 2]
```

Die ersten 100 Werte von $r_3^*(n)$ lauten:

```
{1, 1, 1, 1, 1, 1, 0, 1, 2, 1, 1, 1, 1, 1, 0, 1, 2, 2, 1, 1, 1, 1, 0, 1, 2, 2, 2, 0, 2, 1, 0, 1, 2, 2, 1
, 2, 1, 2, 0, 1, 3, 1, 1, 1, 2, 1, 0, 1, 2, 3, 2, 1, 2, 3, 0, 1, 2, 1, 2, 0, 2, 2, 0, 1, 3, 3, 1, 2, 2, 1
, 0, 2, 2, 3, 2, 1, 2, 1, 0, 1, 4, 2, 2, 1, 2, 3, 0, 1, 4, 3, 1, 0, 1, 2, 0, 1, 2, 3, 3, 2}
```

Mathematica:

```
Table[Length[PowersRepresentations[i, 3, 2]], {i, 1, 100}] oder
a[ n_] := If[ n < 0, 0, Sum[ Boole[ n == i^2 + j^2 + k^2], {i, 0,
Sqrt[n]}, {j, 0, i}, {k, 0, j}]];
```

Wie oben bereits erwähnt hat $r_3(n)$ für einige n den Wert 0. Das bedeutet, dass nicht jede natürliche Zahl als Summe von 3 Quadratzahlen geschrieben werden kann. Hier eine Liste der ersten 'nicht darstellbaren' Zahlen:

```
{7, 15, 23, 28, 31, 39, 47, 55, 60, 63, 71, 79, 87, 92, 95, 103, 111, 112, 119, 124, 127, 1
35, 143, 151, 156, 159, 167, 175, 183, 188, 191, 199}
```

Mathematica:

```
Select[Range[199], Length[PowersRepresentations[#, 3, 2]] == 0 &]
```

Hier zwei Graphen von $r_3(n)$:

Dritte Dimension: Kugeln und Gitterpunkte

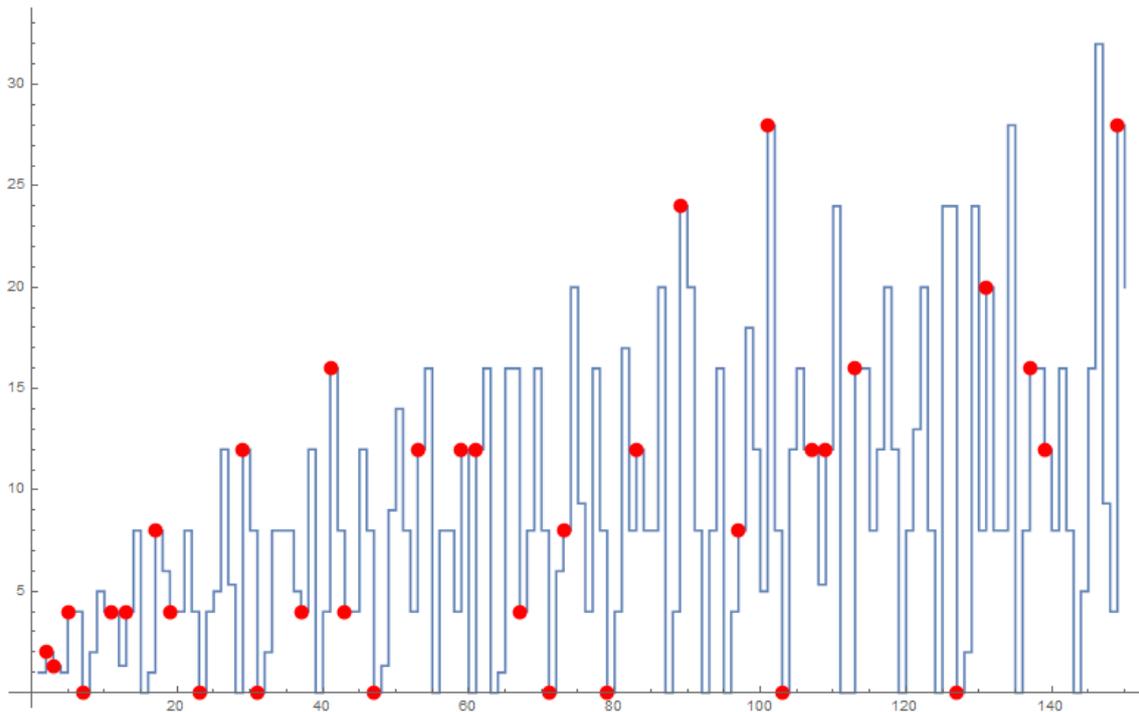


Abbildung 84: $r_3(n)$: Anzahl/6 der möglichen Darstellungen von n als Summe von 3 Quadraten. Mit Primzahlen (rote Punkte) ist kein Zusammenhang erkennbar.

Mathematica:

```
Show[ListLinePlot[Table[{n, SquaresR[3, n]/6}, {n, 1, 150}], InterpolationOrder->0], ListPlot[Table[{Prime[n], SquaresR[3, Prime[n]]/6}, {n, 1, PrimePi[150]}], PlotStyle->Red]]
```

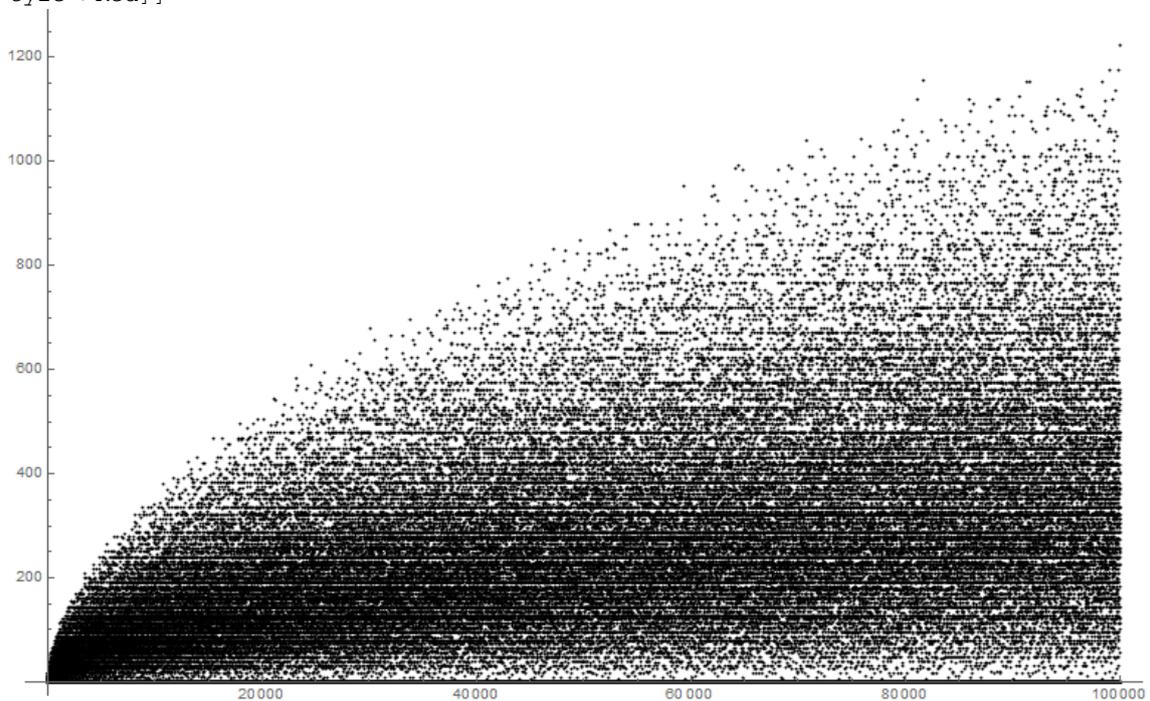


Abbildung 85: $r_3(n)$: Anzahl/6 der Darstellungen von n als Summe von 3 Quadraten (bis 100000)

Mathematica:

```
ListPlot[Table[{n, SquaresR[3, n]/6}, {n, 1, 100000}], PlotRange->Full, PlotStyle->Black]
```

Und hier ein paar Graphen von $r_3^*(n)$:

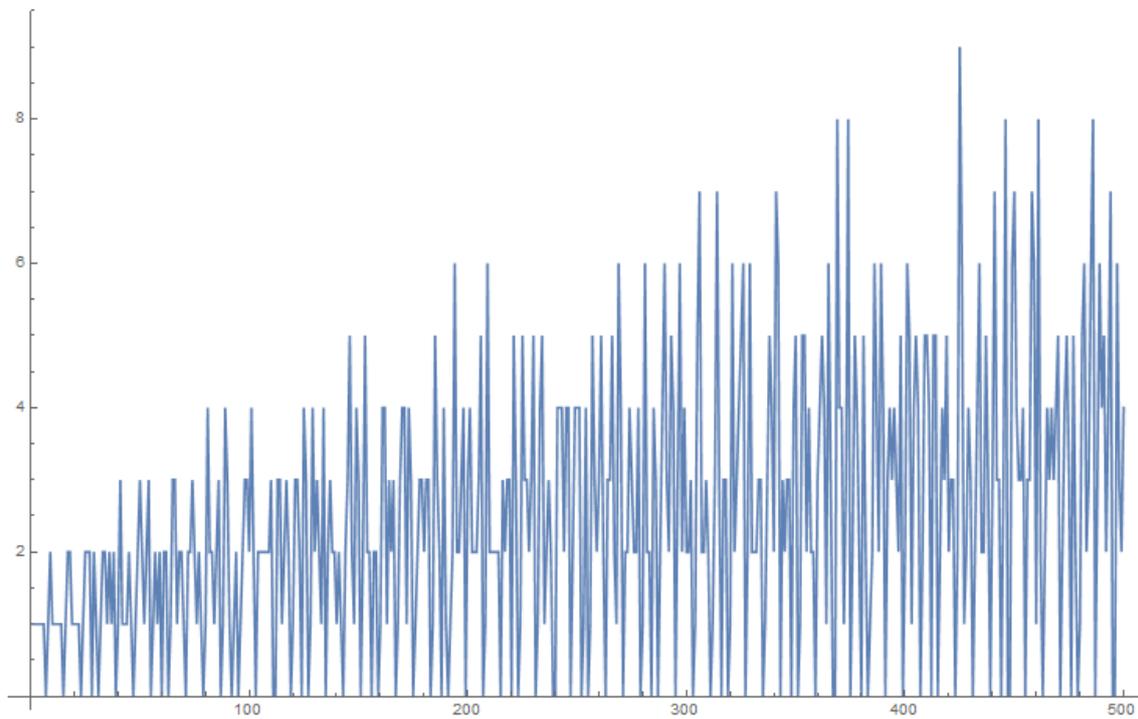


Abbildung 86: $r_3^*(n)$: Anzahl unterschiedlicher Darstellungen von n als Summe von drei Quadraten, ($n=1$ bis 500)

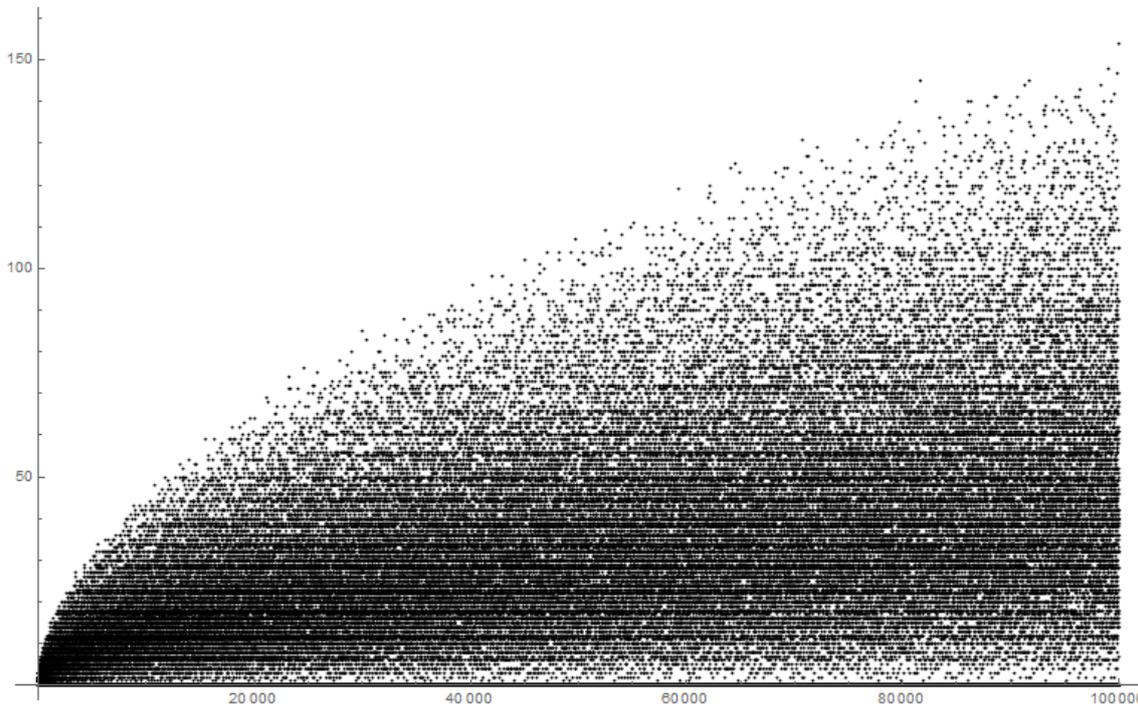


Abbildung 87: $r_3^*(n)$: Anzahl unterschiedlicher Darstellungen von n als Summe von drei Quadraten, ($n=1$ bis 100000)

```
ListPlot[Table[{n, Length[PowersRepresentations[n, 3, 2]]}, {n, 1, 100000}],
PlotRange->Full, PlotStyle->Black]
```

Dritte Dimension: Kugeln und Gitterpunkte

Da sich alles auf einer n -Sphere (hier einer 2-Sphere, einer Kugeloberfläche) abspielt, ist es naheliegend, statt kartesischen Koordinaten (x, y, z) Kugelkoordinaten (r, φ, θ) zu verwenden, da bei unseren Untersuchungen der Kugelradius r konstant bleibt und nur die beiden Freiheitsgrade der Winkel φ und θ übrigbleiben. Dies führt zu der verlockenden Idee, φ und θ als 2-dimensionale kartesische Koordinaten zu interpretieren. Damit kann man alle interessanten Muster auf den Kugeloberflächen auch als zweidimensionale Darstellungen sehen:

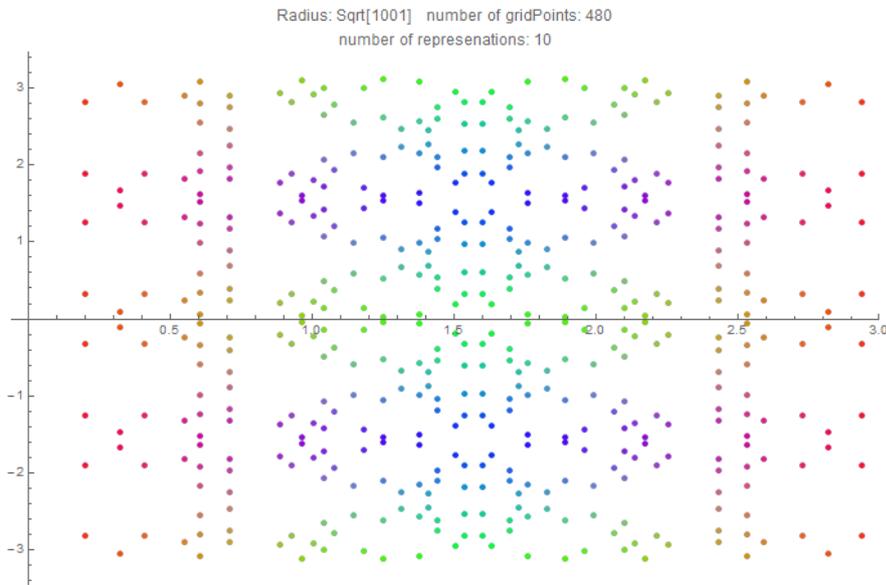


Abbildung 88, Gitterpunkte auf Oberfläche einer Kugel mit $r^2 = 1001$, Winkel der Kugelkoordinaten als 2-dimensionale kartesische Koordinaten interpretiert (gleiche Farbe bedeutet gleiche Punkte hinsichtl. Spiegel-Operationen)

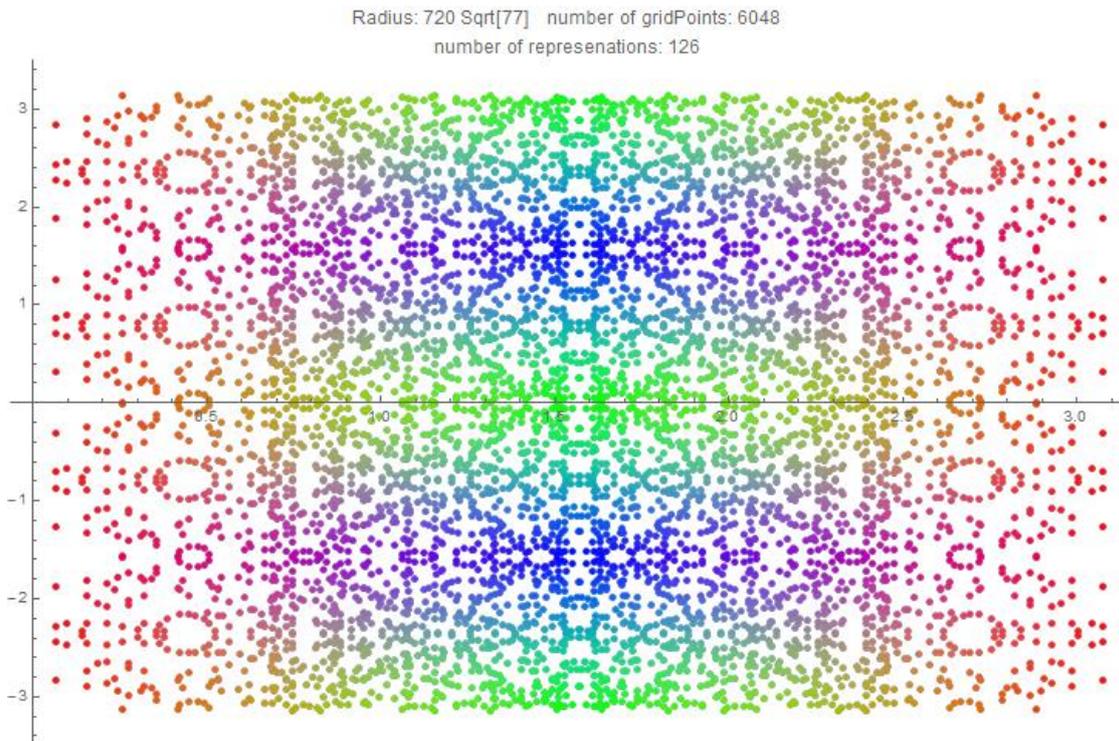


Abbildung 89: wie oben, jedoch $r^2 = 11$! (Mathematica-Programme im Anhang)

9.2.1 FORMELN UND EIGENSCHAFTEN

Wir beschränken uns hier auf die Funktion $r_3(n)$, die die Anzahl von Gitterpunkten im 3-dimensionalen Raum, die auf einer Kugeloberfläche mit Radius \sqrt{n} liegen, berechnet. Für die Funktion $r_3^*(n)$, die die Anzahl der unterschiedlichen, positiven Gitterpunkte berechnet, so dass gilt: $0 \leq n_1 \leq n_2 \leq n_3$ und $n_1^2 + n_2^2 + n_3^2 = n$ sei hier verwiesen auf z.B. ⁴⁸

Die erzeugende Funktion von $r_3(n)$ ist die elliptische Jacobi $\vartheta_3(n)$ -Funktion hoch 3:

$$\sum_{n=0}^{\infty} r_3(n)x^n = \vartheta_3^3(x) = 1 + 6x + 12x^2 + 8x^3 + 6x^4 + 24x^5 + \dots \quad (141)$$

Explizite Formeln:

$$r_3(n) = \begin{cases} 24h(-n), & \text{wenn } n \equiv 3 \pmod{8} \\ 12h(-4n), & \text{wenn } n \equiv 1,2,5,6 \pmod{8} \\ 0, & \text{wenn } n \equiv 7 \pmod{8} \end{cases} \quad (142)$$

(wobei $h(n)$ die ‚Klassenzahl‘⁴⁹ von n bedeutet).

Erwähnenswert ist noch der ‚Drei-Quadrate-Satz‘: von C. F. Gauß:

Für jede natürliche Zahl n , die sich als Summe von drei Quadraten darstellen lässt ($x^2 + y^2 + z^2 = n$, mit $n, x, y, z \in \mathbb{N}$) gilt:

$$n = 4^k m \text{ mit } 4 \nmid m \text{ und } m \not\equiv 7 \pmod{8}$$

⁴⁸ <http://mathworld.wolfram.com/SumofSquaresFunction.html>

⁴⁹ <https://de.wikipedia.org/wiki/Klassenzahl>

9.3 VIERTE DIMENSION: HYPERKUGELN UND GITTERPUNKTE AUF ‚GLOMES‘

Ganz analog wie im dreidimensionalen Raum suchen wir hier nach Gitterpunkten in einem ganzzahligen, rechtwinkligen Koordinatensystem, die auf der Oberfläche einer ‚Hyperkugel‘ liegen können. Die Anzahl dieser möglichen Gitterpunkte, die von der Hyperkugeloberfläche berührt werden, hängt auch hier sehr stark von Radius der Hyperkugel ab. Wir wollen diese Hyperkugeloberfläche im Folgenden als ‚Glome‘ bezeichnen und halten uns damit an den allgemeinen Sprachgebrauch. Stellen wir uns vor, wir vergrößern den Hyperkugelradius kontinuierlich, dann wird unser Glome der Reihe nach durch die Gitterpunkte unseres vierdimensionalen Koordinatensystems laufen. Uns interessieren die Gitterpunkte, die exakt auf der Glome liegen. Hier ein paar Beispiele:

(Hier hätte der Autor gerne ein paar Beispiele eingefügt, jedoch ist es nicht so einfach, vierdimensionale Objekte visuell darzustellen. Es gibt jedoch einen Trick, wie man dies trotzdem bewerkstelligen kann, s. unten).

Bleiben wir vorerst noch im abstrakten, mathematischen Raum.

Gab es im dreidimensionalen Raum noch für einige Werte mit ganzzahligem Radius-Quadrat r^2 keine ganzzahlige Lösungen der Gleichung $x^2 + y^2 + z^2 = r^2$ (und deshalb auch keine entsprechende Gitterpunkte), so ist dies im vierdimensionalen Bereich nicht mehr der Fall: Die Gleichung $x^2 + y^2 + z^2 + t^2 = r^2$ besitzt für jedes ganzzahlige r^2 (d.h. für jede natürliche Zahl) ganzzahlige Lösungen! Jede natürliche Zahl lässt sich als Summe von vier Quadraten darstellen. Dies ist der berühmte Satz von Lagrange aus dem Jahr 1770.

Die Funktion, die die Anzahl von ganzzahligen Gitterpunkten auf einer Glome (‚Oberfläche‘ einer vierdimensionalen Hyperkugel) berechnet, wird in der Literatur mit $r_4^*(n)$ bezeichnet. Mit $r_4^*(n)$ bezeichnen wir die Anzahl der unterschiedlichen, positiven Gitterpunkte, so dass gilt: $0 \leq n_1 \leq n_2 \leq n_3 \leq n_4$ und $n_1^2 + n_2^2 + n_3^2 + n_4^2 = n$.

Alle Funktionswerte von $r_4(n)$ sind durch 8 teilbar. Die ersten 50 Werte lauten:

```
{8, 24, 32, 24, 48, 96, 64, 24, 104, 144, 96, 96, 112, 192, 192, 24, 144, 312, 160, 144, 256, 288, 192, 96, 248, 336, 320, 192, 240, 576, 256, 24, 384, 432, 384, 312, 304, 480, 448, 144, 336, 768, 352, 288, 624, 576, 384, 96, 456, 744}
```

Mathematica:

```
SquaresR[4, Range[50]]
```

Die 8 Lösungen für $r^2 = 3$ lauten z. B.:

```
{{1, -1, -1, 0}, {1, 1, -1, 0}, {-1, 0, -1, -1}, {-1, -1, 0, 1}, {1, 1, 0, 1}, {1, -1, 0, 1}, {-1, 0, 1, 1}, {0, 1, -1, -1}}
```

Mathematica:

```
FindInstance[x^2+y^2+z^2+t^2==3, {x, y, z, t}, Integers, 8]
```

Diese Lösungen gehen gegenseitig durch Permutationen oder symmetrische Spiegelungen auseinander hervor. Die Anzahl der wirklich unterschiedlichen Lösungen $r_4^*(n)$ ist in diesem Fall 1:

```
{0, 1, 1, 1}
Mathematica:
PowersRepresentations[3, 4, 2]
```

Die ersten 50 Werte von $r_4^*(n)$ lauten:

```
{1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 3, 2, 2, 2, 2, 1, 1, 3, 3, 3, 3, 2, 2, 2, 1, 3, 4, 2, 4, 3, 3, 2, 2, 3, 4, 3, 2, 4, 2, 2, 2, 4, 5}
```

```
Mathematica:
Table[Length[PowersRepresentations[i, 4, 2]], {i, 1, 50}] (*oder:*)
a[n_] := If[n < 0, 0, Sum[Boole[n == i^2 + j^2 + k^2 + l^2], {i, 0, Sqrt[n]}, {j, 0, i}, {k, 0, j}, {l, 0, k}]];
Table[a[n], {n, 1, 50}]
```

Hier zwei Graphen von $r_4(n)$:

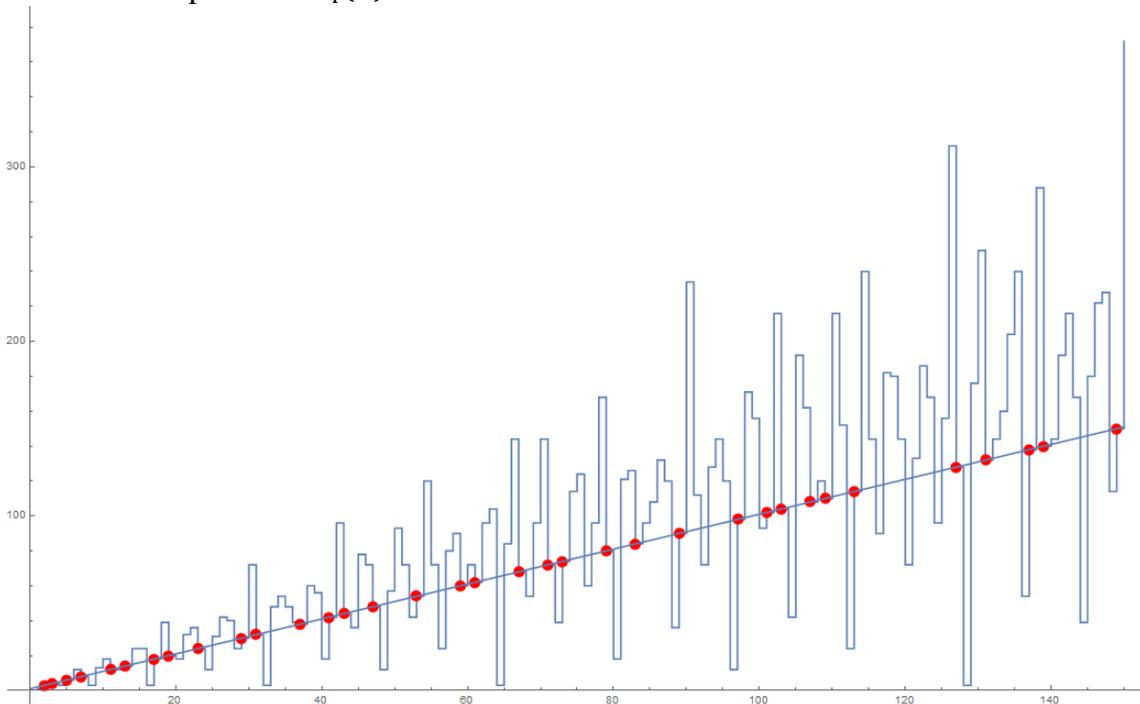


Abbildung 90: $r_4(n)$: Anzahl/8 der möglichen Darstellungen von n als Summe von 4 Quadraten. Werte, die bei Primzahlen liegen, sind rot markiert.

```
Mathematica:
Show[ListLinePlot[Table[{n, SquaresR[4, n]/8}, {n, 1, 150}],
InterpolationOrder->0], ListPlot[
Table[{Prime[n], SquaresR[4, Prime[n]]/8}, {n, 1, PrimePi[150]}],
PlotStyle->{Red, PointSize[0.01]}, Plot[x+1, {x, 0, 150}]]
```

Vierte Dimension: Hyperkugeln und Gitterpunkte auf ‚Glomes‘

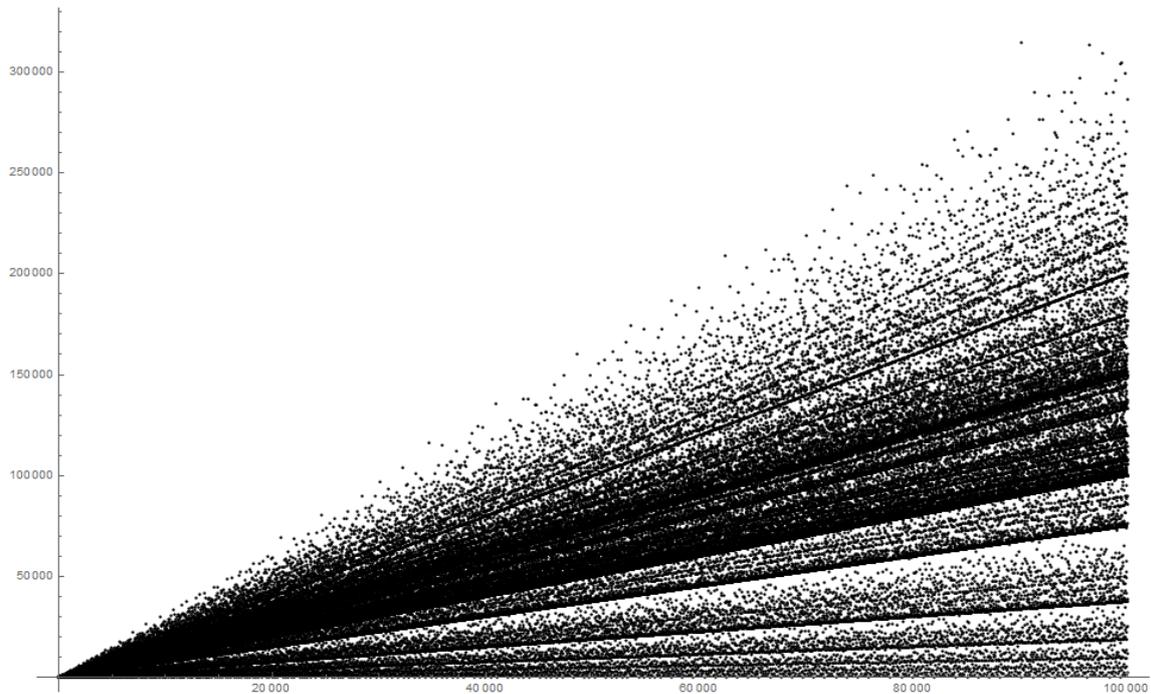


Abbildung 91: $r_4(n)$: Anzahl/8 der Darstellungen von n als Summe von 4 Quadraten (bis 100000)

Mathematica:

```
ListPlot[ParallelTable[{n, SquaresR[4, n]/8}, {n, 1, 100000}],  
PlotRange->Full, PlotStyle->Black]
```

In Abbildung 90 ist deutlich zu sehen, dass alle Werte von $r_4(n)$ auf einer Geraden liegen wenn n eine Primzahl ist, siehe Formel (137). Diese Erscheinung tritt nur in der vierten Dimension auf. Weder in den niedrigeren Dimensionen noch in höheren Dimensionen ist so ein einfacher Zusammenhang zwischen Primzahlen und der Anzahl von Gitterpunkten auf n -Spheres erkennbar.

Und hier ein paar Graphen von $r_4^*(n)$:

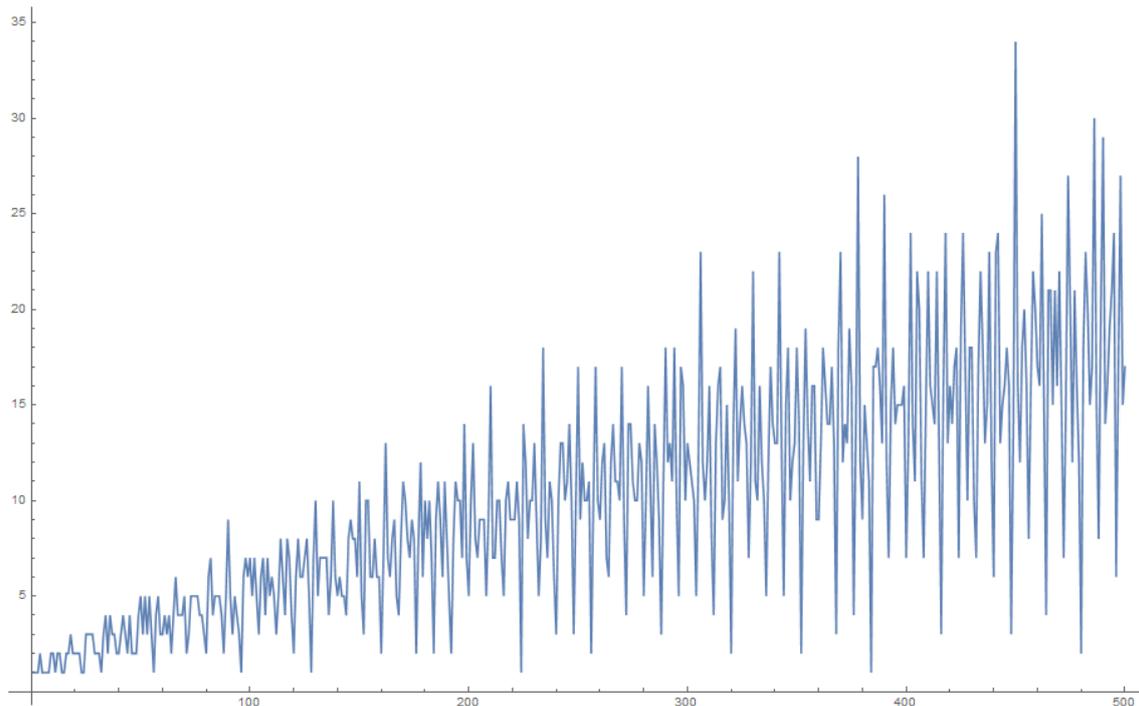
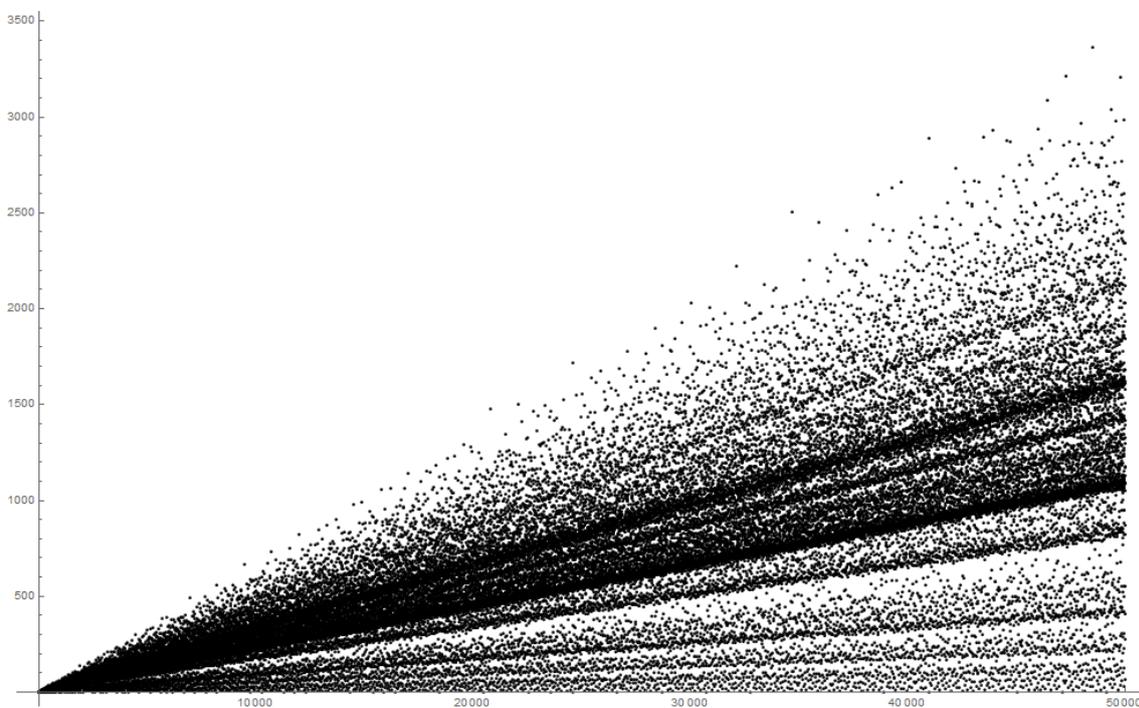


Abbildung 92: $r_4^*(n)$: Anzahl unterschiedlicher Darstellungen von n als Summe von vier Quadraten, ($n=1$ bis 500)

Mathematica:

```
ListLinePlot[Table[Length[PowersRepresentations[i,4,2]],{i,1,500}],
PlotLegends->Automatic,PlotRange->All]
```



Mathematica:

```
ListPlot[ParallelTable[{n,Length[PowersRepresentations[n,4,2]]},{n,1,50000}],
PlotRange->Full,PlotStyle->Black]
```

Aus den Graphen für $r_4(n)$ und $r_4^*(n)$ sehen wir, dass das asymptotische Verhalten beider Funktionen linear ist.

Wie im letzten Abschnitt für dreidimensionale Kugeln verwenden wir einen Trick, um das Ganze um eine Dimension zu reduzieren, indem wir statt kartesischen Koordinaten (x, y, z, t) Hyper-Kugelkoordinaten $(r, \varphi, \theta, \psi)$ verwenden, da bei unseren Untersuchungen der Kugelradius r konstant bleibt und nur die drei Freiheitsgrade der Winkel φ , θ und ψ übrig bleiben. φ , θ und ψ werden als dreidimensionale kartesische Koordinaten interpretiert. Damit kann man alle interessanten Muster auf den Hyper-Kugeloberflächen auch als dreidimensionale Darstellungen sehen:

Die farbige Darstellung wurde so gewählt, dass gleiche (absolute) x, y, oder z-Koordinaten gleiche R-, G, oder B-Werte im RGB-Farbraum bedeuten.

Beispiel 1: $r^2 = 1001$.

Beispiel 2: $r^2 = 10007$

Animationen (Blicke auf die Oberfläche der 4-dimensionalen Kugeln) befinden sich auf der beiliegenden Computer-CD, bzw. als Mathematica-Programm der Animationen im Anhang.

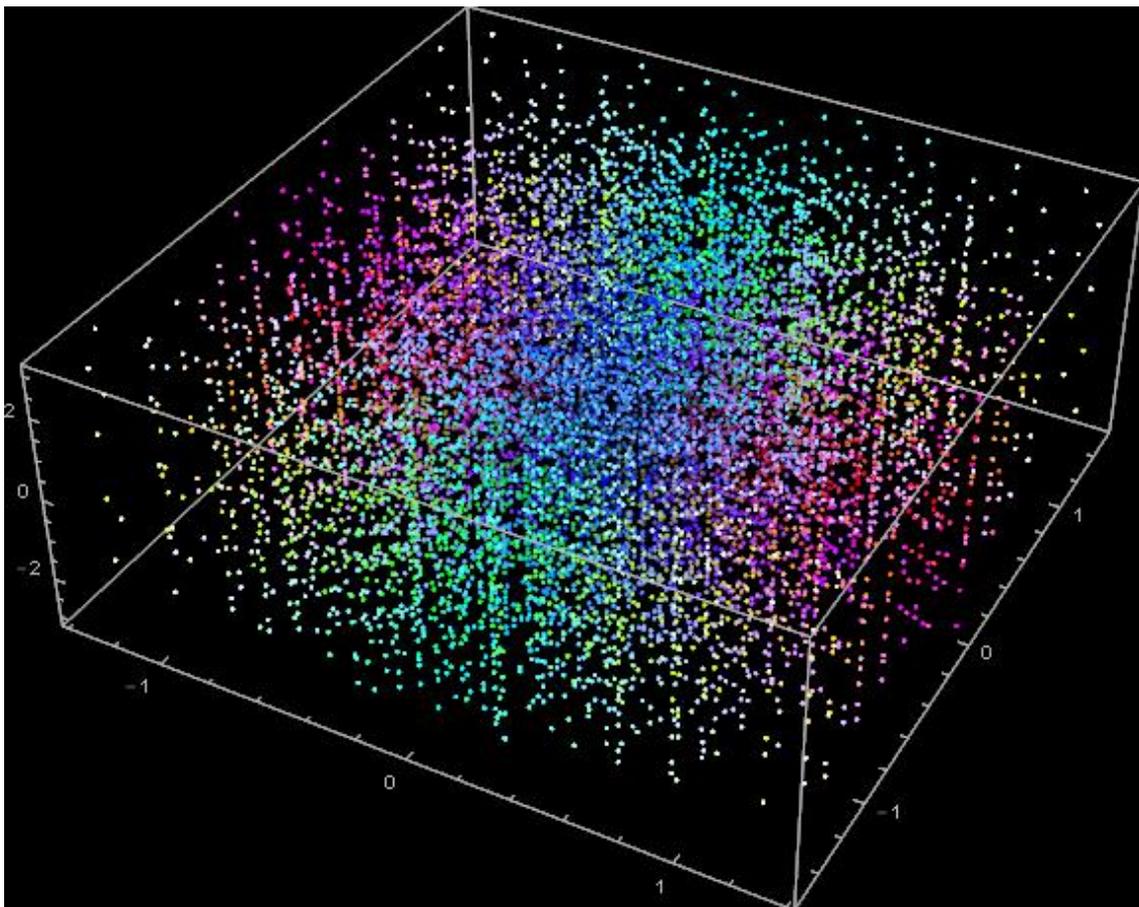


Abbildung 93: Beispiel1: Gitterpunkte auf der 3-Sphere einer 4-dimension. Kugel mit $r^2=1001$

(Mathematica-Programm im Anhang).

Die verborgenen Strukturen erscheinen erst bei Ansichten parallel zu den Koordinatenachsen:

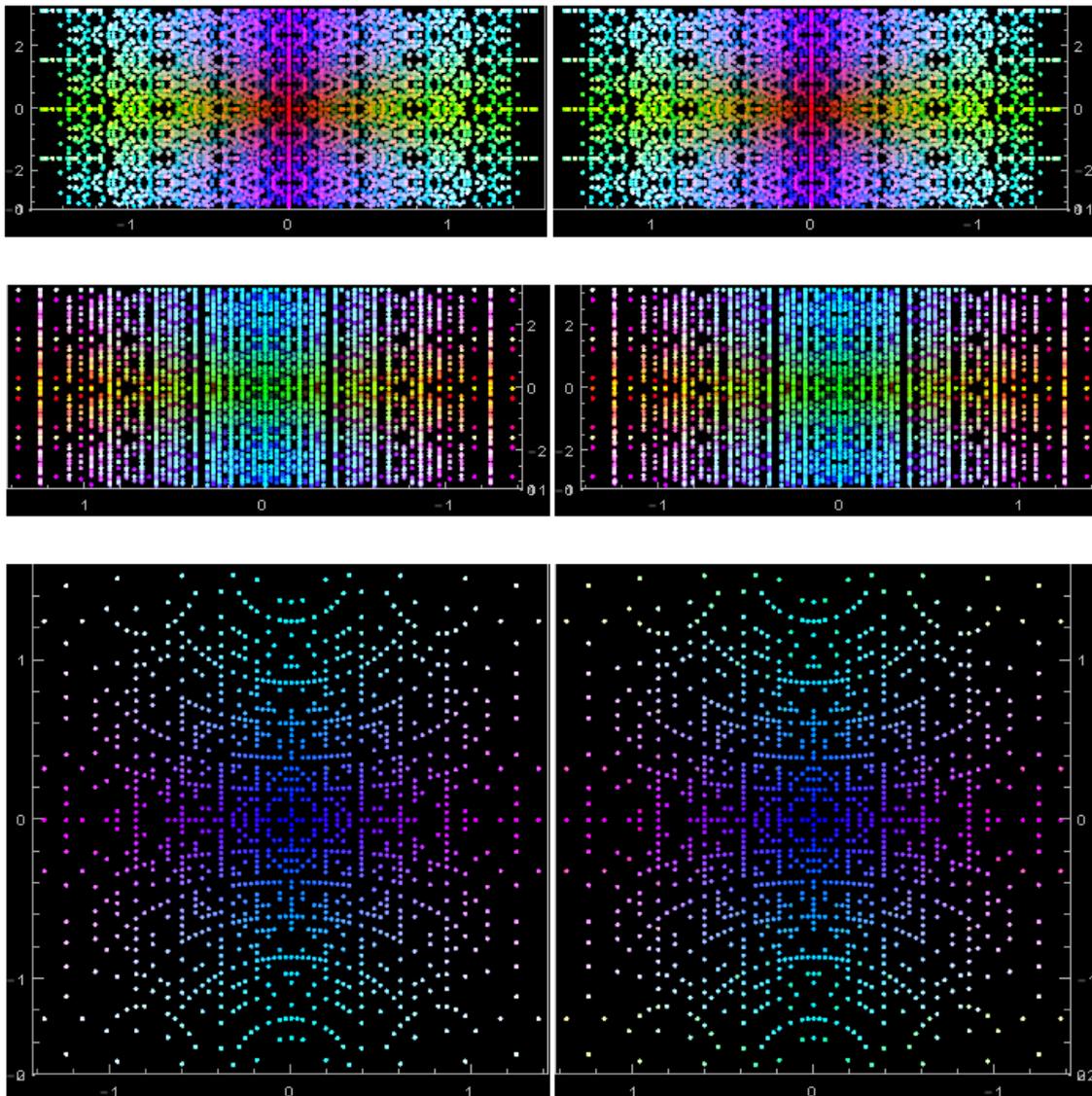


Abbildung 94: 6 Ansichten von Abbildung 93: Rechts/Links, vorne/hinten, Oben/unten

Mathematica:

```
grTable = {
Show[obj,ViewPoint->{Infinity,0,0},ImageSize->Medium],
Show[obj,ViewPoint->{-Infinity,0,0},ImageSize->Medium],
Show[obj,ViewPoint->{0,Infinity,0},ImageSize->Medium],
Show[obj,ViewPoint->{0,-Infinity,0},ImageSize->Medium],
Show[obj,ViewPoint->{0,0,Infinity},ImageSize->Medium],
Show[obj,ViewPoint->{0,0,-Infinity},ImageSize->Medium]
}
```

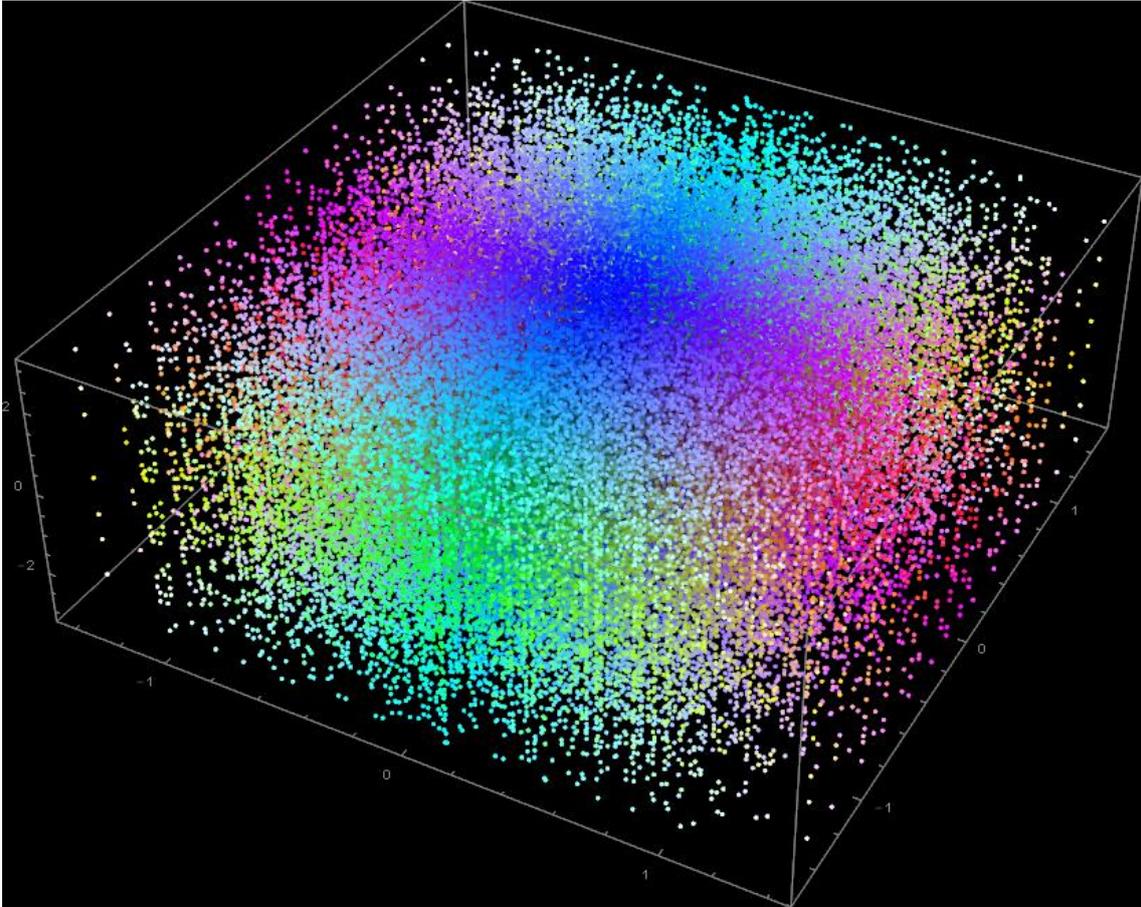


Abbildung 95: Beispiel2: Gitterpunkte auf 3-Sphere einer 4-dimensionalen Kugel mit $r^2=10007$

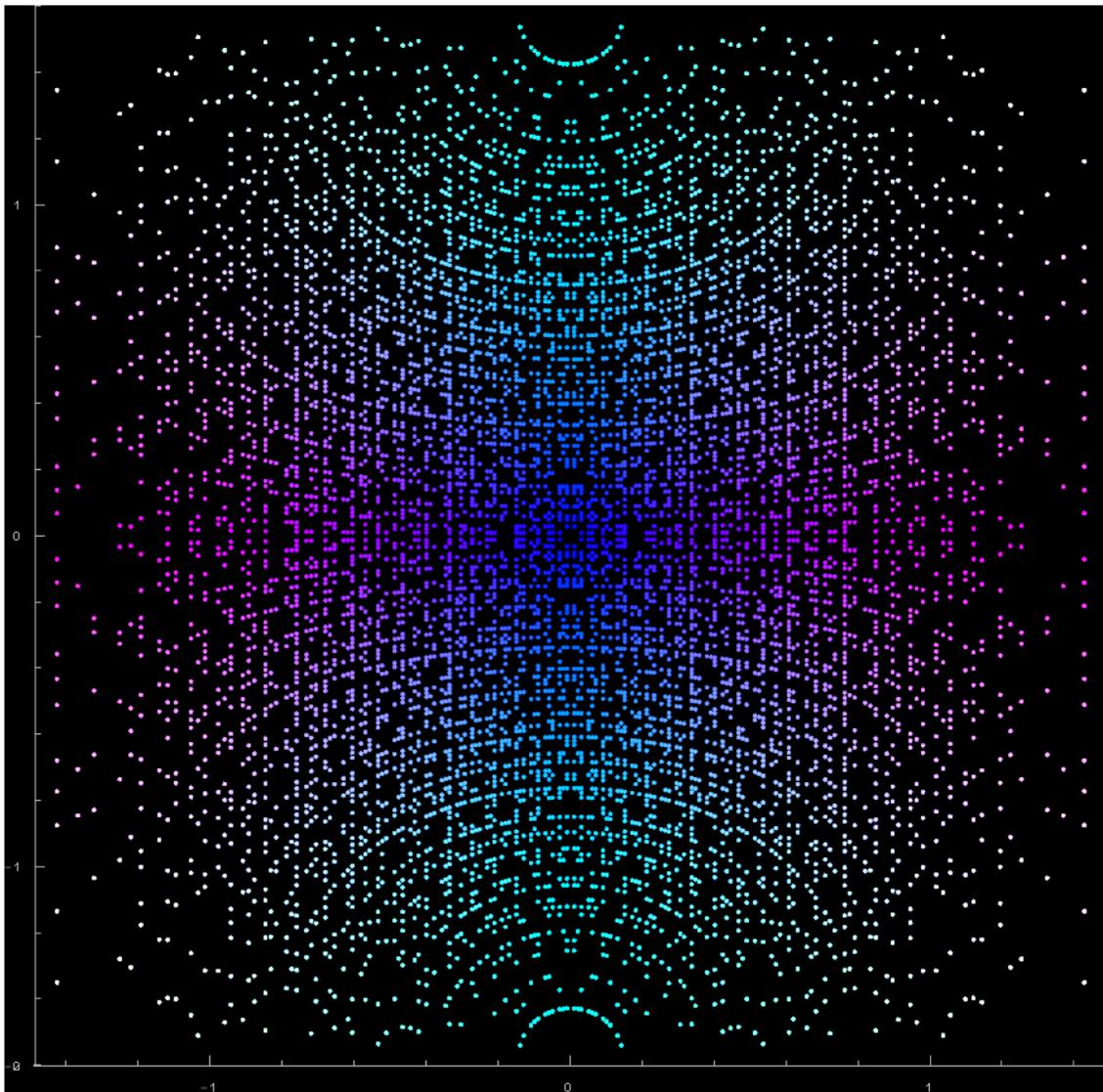
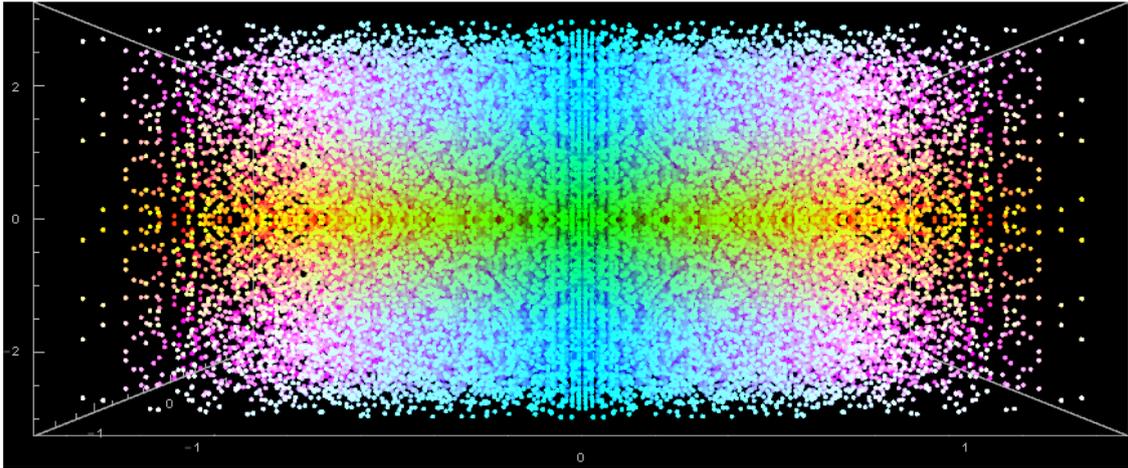


Abbildung 96: 2 Ansichten von Abbildung 95: Beispiel2: Gitterpunkte auf 3-Sphere einer 4-dimensionalen Kugel mit $r^2=10007$ Rechts/Links, vorne/hinten, Oben/unten

Die sich ergebenden Bilder der vierdimensionalen Kugeloberflächen sind nach Ansicht des Autors am schönsten, wenn das Quadrat des Kugelradius eine Primzahl ist. Bei ihnen

ist das Verhältnis zwischen Punkthäufungen und leeren Zwischenräumen am ausgewogensten. Dies kommt auch in ABBILDUNG 90 zum Ausdruck: Die Dichte der Kugelpunkte auf 3-Spheres ist für Primzahlen immer im mittleren Bereich.

9.3.1 FORMELN UND EIGENSCHAFTEN

Wie bei den niedrigeren Dimensionen beschränken wir uns hier auf die Funktion $r_4(n)$, die die Anzahl von Gitterpunkten im 4-dimensionalen Raum, die auf einer Hyperkugeloberfläche mit Radius \sqrt{n} liegen, berechnet. Für die Funktion $r_4^*(n)$, die die Anzahl der unterschiedlichen, positiven Gitterpunkte berechnet, so dass gilt: $0 \leq n_1 \leq n_2 \leq n_3 \leq n_4$ und $n_1^2 + n_2^2 + n_3^2 + n_4^2 = n$ sei hier auf entsprechende andere Quellen verwiesen. Die erzeugende Funktion von $r_4(n)$ ist die elliptische Jacobi $\vartheta_3(n)$ -Funktion hoch 4:

$$\sum_{n=0}^{\infty} r_4(n)x^n = \vartheta_3^4(x) = 1 + 8x + 24x^2 + 32x^3 + 24x^4 + 48x^5 + \dots \quad (143)$$

Explizite Formeln:

$$r_4(n) = \begin{cases} 8\sigma_1(n), & \text{wenn } n \text{ ungerade} \\ 24\sigma^0(n), & \text{wenn } n \text{ gerade, wobei } \sigma^0(n) = \sum_{2 \nmid d, d|n} d \end{cases} \quad (144)$$

Dies lässt sich etwas leichter formulieren als:

$$r_4(n) = \begin{cases} 8\sigma_1(n), & \text{wenn } 4 \nmid n \text{ (} n \text{ nicht durch 4 teilbar)} \\ 8\sigma_1(n) - 32\sigma_1\left(\frac{n}{4}\right), & \text{sonst (} n \text{ durch 4 teilbar)} \end{cases} \quad (145)$$

Oder auch:

$$r_4(n) = 8 \sum_{d|n, 4 \nmid d} d \quad (146)$$

10 VON OCRONS UND GOCRONS: GÖDEL LÄSST GRÜßEN

10.1 WAS SIND OCRONS BZW. GOCRONS?

Die Abkürzung ‚OCRON‘ steht für „Operator Chain Representation Of Number“. Ein OCRON ist eine Darstellungsweise für natürliche Zahlen $n > 0$, die funktional arbeitet (ähnlich wie ein kleines Computerprogramm, das in einer Programmiersprache mit sehr wenigen einfachen Befehlen geschrieben wurde) und Operatoren, die von links nach rechts sequentiell abgearbeitet werden. Für die Abarbeitung der Operator-Sequenz (=OCRON) verwenden wir die sogenannte ‚Polnische Notation‘, die mit einem Stack arbeitet: In der Liste erscheinende Zahlen- und Basis-Symbole werden einfach auf den Stack ‚gepusht‘. Operatoren verknüpfen die untersten zwei Stack-Einträge und lassen die darüber liegenden Stackeinträge nachrutschen. Der Stack kann im Laufe einer OCRON-Abarbeitung beliebig groß werden, am Ende sollte jedoch nur noch ein Eintrag übrig bleiben: der Wert des OCRONS. So lässt sich jede Zahl in ein OCRON umwandeln. Ein OCRON beschreibt, im Gegensatz zu einer normalen Summendarstellung nicht nur einfach die Summe, sondern die Prozedur, wie diese Zahl entsteht.

Die Umkehrung gilt natürlich nicht, d.h. nicht jede Zeichenkette, die aus dem verwendeten Symbolvorrat besteht, gehört zu einer Zahl. Die Logiker sprechen von wohlgeformten und nicht-wohlgeformten Zeichenketten. Bei den meisten Arten von OCRONS gibt es mehr nicht-wohlgeformte OCRONS als wohlgeformte OCRONS. Weiter unten werden wir jedoch Methoden vorstellen, die es erlauben auch nicht-wohlgeformte OCRONS zu interpretieren und einer Zahl zuzuweisen.

Es gibt auch OCRON-Systeme (siehe ‚Prime OCRONS‘) die von Haus aus immer wohlgeformte Operatorsequenzen bilden. Diese höchstinteressante Systeme stellen eine bijektive Abbildung der natürlichen Zahlen auf eine Menge von Symbolen dar, die eindeutig umkehrbar. Es sei hier vorweggenommen, dass ein ‚GOCRON‘ ein ‚gödelisiertes‘ OCRON ist. Frei nach der Methode des genialen österreichischen Mathematikers Kurt Gödel, der diese Methode erfand (Gödel ordnete mathematischen Aussagen, Sätzen oder Formeln einfach natürliche Zahlen zu) werden wir jeder OCRON-Kette einen Zahlenwert zuordnen. Diesen Prozess nennen wir „Gödelisierung“. Er beschreibt einen Wechsel der Bedeutungsebene: Von einer prozeduralen Bedeutung in eine arithmetische zahlenmäßige Bedeutung. Anders als bei Gödel, bei dem diese Gödelisierung nur hypothetisch und theoretisch (für seinen Beweis des Unvollständigkeitstheorems) durchgeführt wurde, werden wir hier ganz konkret mit gödelisierten Zahlen arbeiten.

Zunächst werden wir die einfachen Zahlendarstellungen wiederholen und zeigen, dass auch sie als OCRONS gedeutet werden können.

10.1.1 SUMMENDARSTELLUNG IN ZAHLENSYSTEMEN

Zunächst einmal sei an dieser Stelle ein kurzer Überblick über Zahlendarstellungen gestattet. Die gängigen, zur Verarbeitung im Computer geeigneten Darstellungsweisen für natürliche Zahlen sind:

Die **Summendarstellung** in Zahlensystemen mit einer geeigneten Basis: Als Basis wird typischerweise 10 (Dezimalsystem), 2 (Binärsystem), 16 (Hexadezimalsystem), oder 8 (Oktalsystem) verwendet. Sei b die Basis, $z_i < b$ die ‚Ziffern‘ und N die höchste vorkommende Potenz zur Basis b der Zahlendarstellung. Dann lässt sich jede natürliche Zahl $n \geq 0$ schreiben als:

$$n = \sum_{i=0}^N z_i b^i, \quad \text{wobei } N = \lfloor (\ln(n)) / \ln b \rfloor \quad (147)$$

Sowohl die Ziffern z_i als auch die Exponenten i werden in der gleichen Summendarstellung mit der gleichen Basis b dargestellt, so dass wir insgesamt eine Darstellung haben, die mit $b + 3$ Symbolen auskommt (nämlich den b Ziffernsymbolen sowie den drei Operator-Symbolen '+', '*' und '^' (Addition, Multiplikation und Potenzierung)). Dies ist eine gemischte Darstellung, da alle drei Operatoren vorkommen. Auch die Ziffern, Basis und Exponent werden in der Regel im gleichen Zahlensystem dargestellt werden. Man kann hier jedoch zu einer reinen nur aus den Operatoren '+ und '^' bestehenden Darstellung kommen, indem die Terme $z_i b^i$ ‚ausaddiert‘ ($b^i + b^i + b^i + \dots$) werden und Terme mit ‚0‘ weggelassen werden. Dies führt dann zu einer Summendarstellung mit nur zwei Operatoren '+ und '^'. Wir verwenden hier für die Basis nicht ‚10‘ sondern das eigene Symbol ‚b‘. Das hat den Vorteil, dass das Symbol ‚0‘ in der reduzierten Darstellung, in der wirklich nur noch die einzelnen Ziffern und die Operatoren vorkommen, nicht mehr auftaucht.

Die minimale Anzahl von verschiedenen Symbolen für die Summendarstellung mit den Operatoren '+', '*' und '^' ist fünf (Binärsystem), die maximale Anzahl $b + 3$ (im b -System).

Die minimale Anzahl von verschiedenen Symbolen für die Summendarstellung mit den Operatoren '+ und '^' Operatoren ist vier (Binärsystem), die maximale Anzahl $b + 2$ (im b -System)

Beispiel: Die Zahl **1280000=1100001101010000000000**_{binär}

(Im Dezimalsystem, mit Operatoren '+', '*' und '^'):

$$8 * 10^5 + 2 * 10^6 + 10^7$$

Oder in Operator-Schreibweise (Stackmethode⁵⁰, $b=10$):

$$8b5^*2b6^*+b7^+$$

(Dezimalsystem, mit Operatoren '+ und '^'):

⁵⁰ Stackmethode: Umgekehrte polnische Notation, ‚b‘ und ‚1‘ werden auf den Stack gepusht, ‚+ und '^‘ verknüpfen die beiden untersten Stackwerte, schreiben das Ergebnis in das unterste Stack-Register und dekrementieren den Stack um 1.

$$10^5 + 10^5 + 10^5 + 10^5 + 10^5 + 10^5 + 10^5 + 10^5 + 10^6 + 10^6 + 10^7$$

Oder in Operator-Schreibweise (Stackmethode, b=10):

$$b5^{\wedge}b5^{\wedge}+b5^{\wedge}+b5^{\wedge}+b5^{\wedge}+b5^{\wedge}+b5^{\wedge}+b5^{\wedge}+b5^{\wedge}+b6^{\wedge}+b6^{\wedge}+b7^{\wedge}+$$

(Binärsystem, mit Operatoren '+', '*' und '^'):

$$1 * 2^{12} + 1 * 2^{14} + 1 * 2^{16} + 1 * 2^{17} + 1 * 2^{22} + 1 * 2^{23} \text{ (dezimal)}$$

$$= 1 * 10^{1*10^{11}+1*10^{10}} + 1 * 10^{1*10^{11}+1*10^{10}+1*10^1} + 1 * 10^{1*10^{100}} + 1 * 10^{1*10^{100}+1} + 1 * 10^{1*10^{100}+1*10^{10}+1*10^1} + 1 * 10^{1*10^{100}+1*10^{10}+1*10^1+1} \text{ (binär)}$$

Offensichtlich ist die Multiplikation mit 1 redundant, so dass wir schließlich bekommen:

(Binärsystem, mit Operatoren '+' und '^'):

$$10^{10^{10}+1+10^{10}} + 10^{10^{10}+1+10^{10}+10} + 10^{10^{10}+10} + 10^{10^{10}+1} + 10^{10^{10}+10^{10}+10} + 10^{10^{10}+10^{10}+10+1}$$

Oder in Operator-Schreibweise (Stackmethode, b=10):

$$bbb1^{\wedge}bb^{\wedge}+^{\wedge}bbb1^{\wedge}bb^{\wedge}+b^{\wedge}+^{\wedge}bbbb^{\wedge}+^{\wedge}+^{\wedge}bbbb^{\wedge}1^{\wedge}+^{\wedge}+^{\wedge}bbbb^{\wedge}bb^{\wedge}+b^{\wedge}+^{\wedge}+^{\wedge}bbbb^{\wedge}bb^{\wedge}+b+1+^{\wedge}+$$

Die Methode, die Beschreibung einer Zahl auf ganz wenige Symbole (z. B. 1,2, '+', '*', '^') zu reduzieren wurde bereits 1944 von dem britischen Mathematiker Reuben Louis Goodstein⁵¹ im Zusammenhang mit der nach ihm benannten Goodstein-Folge⁵² untersucht. Diese Folge hat interessante Eigenschaften, da ihre Glieder unvorstellbar große Werte erreichen und (so behauptet es der Satz von Goodstein) nach endlich vielen Schritten wieder den Wert 0 erreicht. Manche Mathematiker behaupten, dieser Satz gehöre in Gödels Kategorie der wahren, jedoch nicht beweisbaren Aussagen.

Man beachte, dass in diesem Beispiel die 0 gar nicht mehr vorkommt, so dass wir im Fall des Binärsystems eine reine Summendarstellung einer Zahl mit nur 2 Operatoren (',' und '^') und den Symbolen 1 und b haben, somit nur 4 Symbole. Diese Summendarstellung ist natürlich mehrdeutig: Wegen der Kommutativität der Operatoren '+', '*' kann an vielen Stellen der Sequenz die Reihenfolge vertauscht werden. Fassen wir der Vollständigkeit halber zusammen, was ohnehin trivial ist:

Die Summendarstellung innerhalb eines Zahlensystems mit einer Basis b und deren Potenzen kann als Sequenz von Operatoren und Symbolen geschrieben werden. Die Sequenz ist umso länger, je weniger Symbole verwendet werden und je kleiner die Basis des Zahlensystems ist. Es sind Darstellungen mit zwei (',' und '^') und drei (',' , '*' und '^') Operatoren möglich.

10.1.2 PRODUKTDARSTELLUNG MIT PRIMFAKTOREN

Die **Produktendarstellung** arbeitet mit Hilfe der Primfaktorenzerlegung. Jede natürliche Zahl $n > 1$ lässt sich schreiben als Produkt von Primfaktoren p_{n_i} , die in der Potenz e_i vorkommen. Sei hier N die Anzahl der verschiedenen vorkommenden Primfaktoren:

⁵¹ R.L. Goodstein(1945), „Function Theory in an Axiom-Free Equation Calculus“. *Proceedings of the London Mathematical Society*

⁵² <https://de.wikipedia.org/wiki/Goodstein-Folge>

Was sind OCRONs bzw. GOCRONs?

$$\mathbf{n} = \prod_{i=1}^N \mathbf{p}_{n_i}^{e_i}, \text{ wobei } N = \omega(\mathbf{n}) \quad (148)$$

Hinweis: $\omega(n)$ verhält sich asymptotisch wie $\approx \ln \ln n$ und kann berechnet werden:

$$\omega(n) = \ln \ln n + B_1 + \sum_{k=1}^{\infty} \left(-1 + \sum_{j=0}^{k-1} \frac{\gamma_j}{j!} \right) \frac{(k-1)!}{(\ln n)^k} \quad (149)$$

wobei B_1 die Mertens-Konstante 0.2614972128 ist und γ_j die Stieltjes Konstanten. Mit B'_1 läßt sich auch die Varianz $\text{var}(\omega(n))$ berechnen:

$$\begin{aligned} \text{var}(\omega(n)) &= \ln \ln n + B'_1 + \sum_{k=1}^{\infty} \frac{c_k}{(\ln n)^k} \\ B'_1 &= B_1 - t - \frac{\pi^2}{6} = 1.83568427, \\ t &= P(2) = \sum_{k=1}^{\infty} \frac{1}{p_k^2} \text{ (Primzetafunktion)} = 0.452247 \end{aligned} \quad (150)$$

Mit $c_1 = 1.0879488865$, und $c_2 = 3.3231293098$

In Mathematica sind $\omega(n)$ und $\Omega(n)$ als zahlentheoretische Funktion `PrimeNu[n]` und `PrimeOmega[n]` implementiert.

Für die Darstellung der p_{n_i} und e_i wiederum können wir wählen: In der Regel werden die p_{n_i} und e_i in der Summendarstellung eines Zahlensystems zu einer Basis b dargestellt werden. So haben wir eine gemischte Zahlendarstellung: Z. B. p_{n_i} und e_i im Dezimalsystem als Summendarstellung, die Gesamtzahl n jedoch als Produktdarstellung. Man kann jedoch auch hier zu einer ‚reinen‘ Darstellung kommen (wobei wir hier als ‚rein‘ verstehen, dass in der Darstellung nur der ‚*‘ und der ‚^‘ Operator vorkommen sollen, nicht jedoch der ‚+‘ Operator. Das führt uns wiederum zu der Idee der ‚OCRONs‘. Nehmen wir an, wir beschränken uns auf die ersten N Primzahlen. Der nächste Schritt besteht darin, auch die n_i (nicht die p_{n_i} !) und e_i in Produktdarstellung anzugeben. Hier kommt dann allerdings Rekursivität ins Spiel, da die Produktdarstellung von n_i oder e_i auch wieder p_{n_i} und e'_i enthalten kann, die in normaler Summendarstellung oder auch Produktdarstellung dargestellt werden können. Der Rekursive Prozess der Umwandlung von Summendarstellung in Produktdarstellung kann solange fortgesetzt werden, bis nur noch die ersten N Primzahlen vorkommen. Dann haben wir eine reine Produktdarstellung einer Zahl, in der nur die ersten N Primzahlen vorkommen (auch in den Primzahlpotenzen).

Beispiel: Die Zahl **12800000**

(In Produktdarstellung unter Verwendung der ersten 3 Primzahlen ($p_1 = 2, p_2 = 3, p_3 = 5$), mit Operatoren ‚*‘ und ‚^‘:

$$\mathbf{12800000} = 2^{12} * 5^5 = 2^{2^2*3} * 5^5 = p_1^{p_1^{p_1*p_2}} * p_3^{p_3}$$

Oder in Operator-Schreibweise (Stackmethode):

$$222^3 * 55^*$$

(In Produktdarstellung unter Verwendung der ersten 2 Primzahlen ($p_1 = 2, p_2 = 3$), mit Operatoren ' * ' und ' ^ ' :

$$12800000 = p_1^{p_1^{p_1 * p_2}} * p_3^{p_3} = p_1^{p_1^{p_1 * p_2}} * p_{p_2}^{p_{p_2}}$$

Oder in Operator-Schreibweise (Stackmethode):

$$222^3 * (p_3) (p_3) ^*$$

Diese Idee, die Anzahl von benötigten Primzahlen zur Darstellung immer weiter zu reduzieren, führt uns im nächsten Kapitel zu den OCRONS mit Prim-Operator. Die Darstellung mit Hilfe von Indices wie z.B. p_{p_3} ist verwirrend und unübersichtlich, deshalb führen wir einen sogenannten Prim-Operator P ein, der einfach die n -te Primzahl liefert, wenn er auf n angewendet wird.

10.2 OCRONS MIT PRIM-OPERATOR

Wir führen die Idee der Produktdarstellung weiter und ersetzen alle in den Basen und Exponenten vorkommenden Werte rekursiv solange durch kleinere, einfachere Primfaktor-Zerlegungen bzw. Indices von (Indices von ... usw.) Primzahlen, bis wir bei der letzten einfachsten Primzahl $p_1 = 2$ angelangt sind, die nicht weiter reduziert werden kann. Diese letzte ‚irreduzible‘ Primzahl 2 nennen wir den ‚2‘-Operator.

Führen wir das letzte Beispiel im letzten Kapitel weiter:

$$12800000 = p_1^{p_1^{p_1 * p_2}} * p_3^{p_3} = p_1^{p_1^{p_1 * p_2}} * p_{p_2}^{p_{p_2}} = p_1^{p_1^{p_1 * p_{p_1}}} * p_{p_{p_1}}^{p_{p_{p_1}}} = 2^{2^{*p_2}} * p_{p_2}^{p_{p_2}}$$

Oder in Operator-Schreibweise (Stackmethode, mit Operatoren 2, P, * und ^):

$$12800000 = 222^2P * ^2PP * 2PP^*$$

Die Operator-Schreibweise ist viel einfacher. Man beachte, dass der ‚2‘-Operator nichts anderes macht, als die ‚2‘ auf den Stack zu ‚pushen‘; der P -Operator berechnet einfach die x -te Primzahl (mit x als aktuellen Stackwert). Die ‚*‘ und ‚^‘-Operatoren arbeiten wie gewohnt und verknüpfen die beiden untersten Stackinträge, schreiben das Ergebnis in die unterste Stack-Zelle und lassen alle darüber liegenden Stack-Einträge ‚nachrutschen‘.

Hier noch einmal ein einfaches Beispiel anhand der Zahl 1763:

1763 ist das Produkt der Primzahlen 41 und 43. Wir benutzen den P -Operator in Funktionsschreibweise: $P(n)$ liefert die n -te Primzahl. Statt $1763 = 41 * 43$ schreiben wir: $1763 = P(13) * P(14)$. Nun wissen wir, dass 13 die 6-te Primzahl ist und $14 = 2 * 7 = 2 * P(4)$.

Daher können wir schreiben $P(13) = P(P(6))$ und $P(14) = P(2 * P(4))$ usw. ... (Evtl. Exponenten zerlegen wir auf die gleiche Weise wie die Basis...). Also ergibt sich:

$$1763 = 41 * 43 = P(13) * P(14) = P(P(6)) * P(2 * P(4)) = P(P(2 * P(2))) * P((2 * P(2^2))) =$$

$P(P(2 * P(2))) * P((2 * P((2)^2)))$, oder in Operator-Schreibweise mit umgekehrter polnischer Notation:

$$1763 = 22P * PP222^P * P *$$

Diese Operator-Sequenzen haben von ihrer ‚Einfachheit‘ her eine gewisse Ähnlichkeit mit der Programmiersprache ‚Brain-Fuck‘.⁵³

Unter den OCRON-Sequenzen gibt es ‚wohlgeformte‘ und ‚nicht-wohlgeformte‘ Sequenzen. Die wohlgeformten lassen sich problemlos abarbeiten. Die nicht-wohlgeformten, z. B. $^{^}p * 222$ ergeben (zunächst noch) keinen Sinn.

Hinweis: OCRONS vom Typ 3 bis 5 (mit * und ^-Operatoren) können redundant und trotzdem wohlgeformt sein. Die Redundanz kommt daher, dass in arithmetischen Darstellungen eine gewisse Vieldeutigkeit herrscht. OCRONS, die nicht mehr verkürzt werden können, nennen wir ‚minimale‘ OCRONS. Beispiel:

Redundant (arithmetisch): $2 * 2 * 5 * 5 * 2$. Redundant (OCRON:): $22*52^*2*$

Minimal (arithmetisch): $2^3 * 5^2$. Minimal (OCRON:): 23^52^*

OCRONS sind nicht eindeutig. Sie können unterschiedliche Elemente in unterschiedlicher Reihenfolge haben, dennoch den gleichen Wert ergeben. Diese Eigenschaft nennen wir „Entartung“. Dies kommt von der Kommutativität der Rechenarten. OCRONS können leicht multipliziert werden, indem die OCRON-Ketten einfach aneinander gehängt werden und mit einem ‚*‘-Operator ergänzt werden:

Beispiel $5 * 7 = 35$

$$2PP \cdot 22^P = 2PP22^P *$$

Beispiel $6 * 12 = 72$

$$2P2 * \cdot 2P22^* = 2P2 * 2P22^* ** = 2P2^22P^* *$$

Hinweis: Das ‚Umformen‘ des redundanten OCRONS "2P2 * 2P22^*" in das minimale OCRON „2P2^22P^*“ gestaltet sich jedoch schwierig und stellt ein noch ungelöstes Problem dar. Näheres dazu im Kapitel 10.3.

10.2.1 OCRONS MIT PRIM-"P" UND "*" -OPERATOR

Das einfachste OCRON, das sich aus der Primfaktorenzerlegung einer Zahl gewinnen lässt, enthält drei Operatoren: 2, P, *. Wie im letzten Kapitel besprochen liefert eine rekursive Zerlegung der auftretenden Basen und Exponenten ein OCRON, das aus 3 Symbolen besteht. Wir nennen das den OCRON-Typ ‚3‘. Jede wohlgeformte Sequenz beginnt mit einer ‚2‘ und endet mit ‚*‘ oder ‚P‘ (d.h. man kann einem Typ3-OCRON anhand des letzten Zeichens sofort ansehen, ob es eine Primzahl oder eine zusammengesetzte Zahl darstellt). Hier ein Beispiel: die ersten 50 natürlichen Zahlen in OCRON-Typ3-Darstellung:

⁵³ <https://en.wikipedia.org/wiki/Brainfuck>

Tabelle 13: Die Zahlen 2 bis 50 in OCRON-Typ3-Darstellung

n	OCRON Typ 3	n	OCRON Typ 3
1	-	26	2P2*P2*
2	2	27	2P2P*2P*
3	2P	28	22*P2*2*
4	22*	29	2PP2*P
5	2PP	30	2PP2P*2*
6	2P2*	31	2PPPP
7	22*P	32	22*2*2*2*
8	22*2*	33	2PPPP2P*
9	2P2P*	34	22*PP2*
10	2PP2*	35	22*P2PP*
11	2PPP	36	2P2P*2*2*
12	2P2*2*	37	2P2*2*P
13	2P2*P	38	22*2*P2*
14	22*P2*	39	2P2*P2P*
15	2PP2P*	40	2PP2*2*2*
16	22*2*2*	41	2P2*PP
17	22*PP	42	22*P2P*2*
18	2P2P*2*	43	22*P2*P
19	22*2*P	44	2PPPP2*2*
20	2PP2*2*	45	2PP2P*2P*
21	22*P2P*	46	2P2P*P2*
22	2PPPP2*	47	2PP2P*P
23	2P2P*P	48	2P2*2*2*2*
24	2P2*2*2*	49	22*P22*P*
25	2PP2PP*	50	2PP2PP*2*

OCRONs vom Typ 3 besitzen keinen Potenz-Operator und sind daher nicht so interessant. Für hohe Potenzen werden OCRONs vom Typ 3 unhandlich (man denke nur an große zusammengesetzte Zahlen oder 2er Potenzen wie z. B. $2^{57885161}$, deren OCRON-Darstellung dann Millionen von Zeichen lang wäre!)

10.2.1.1 ENTARTUNG VON TYP3-OCRONS

Unter Entartung verstehen wir die Tatsache, dass es zu einer Zahl n im Allgemeinen mehrere OCRON-Darstellungen gibt. Die Umkehrung gilt natürlich nicht, d.h. zu einem OCRON gibt es nur eine eindeutige Zahl n . Diese Entartung wächst sehr schnell mit n , wie folgende Grafik zeigt:

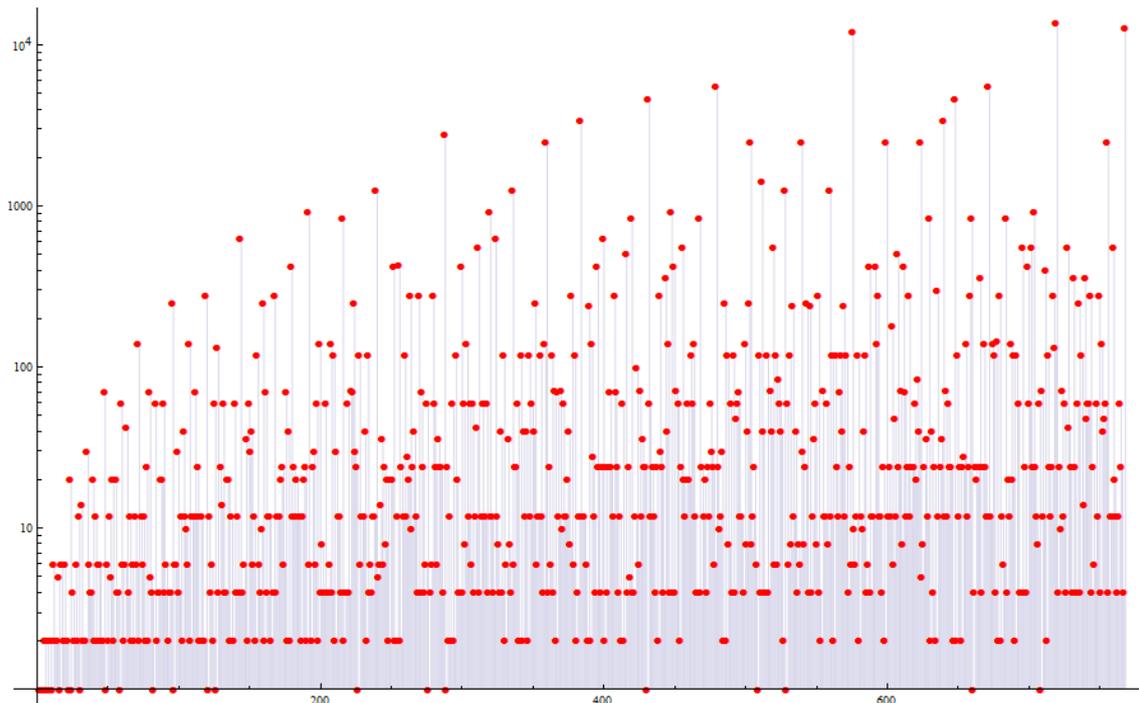


Abbildung 97: Entartung von wohlgeformten OCRON3s bis $n=768$ (logarithmisch dargestellt)

```

Mathematica:
data =
Import["Primzahlen/data/ocron3_wohlgeformt_Degeneration_gut_bis_768.tx
t", "CSV"]
ListLogPlot[data, PlotStyle->Red, AxesLabel->Automatic, Filling->
Axis, PlotMarkers->Automatic, PlotRange->All]
    
```

10.2.2 OCRONS MIT PRIM-"P", "*" - UND "^"-OPERATOR

Diesem OCRON-Typ wollen wir die meiste Aufmerksamkeit widmen. Wir nennen dieses OCRON ‚Typ4-OCRON‘, da es die 4 Operatoren: 2, P, *, ^ enthält. Beim Typ4-OCRON haben wir zusätzlich einen Potenz-Operator. **Es spiegelt die Primfaktor-Zerlegung einer Zahl wider.** Jede wohlgeformte Sequenz beginnt mit einer ‚2‘ und endet mit ‚*‘, ‚^‘ oder ‚P‘ (d.h. man kann einem OCRON anhand des letzten Zeichens sofort ansehen, ob es eine Primzahl, eine zusammengesetzte Zahl oder eine Potenzzahl darstellt). Hier ein Beispiel: die ersten 50 natürlichen Zahlen in OCRON Typ4-Darstellung:

Tabelle 14: Die Zahlen 2 bis 49 in OCRON Typ4-Darstellung

n	OCRON Typ 4	n	OCRON Type 4
2	2	26	222P*P*
3	2P	27	2P2P^
4	22^	28	22^22^P*
5	2PP	29	22PP*P
6	22P*	30	22P*2PP*
7	22^P	31	2PPPP
8	22P^	32	22PP^
9	2P2^	33	2P2PPP*
10	22PP*	34	222^PP*
11	2PPP	35	2PP22^P*
12	22^2P*	36	22^2P2^*
13	22P*P	37	22^2P*P
14	222^P*	38	222P^P*
15	2P2PP*	39	2P22P*P*
16	222^^	40	22P^2PP*
17	22^PP	41	22P*PP
18	22P2^*	42	22P*22^P*
19	22P^P	43	222^P*P
20	22^2PP*	44	22^2PPP*
21	2P22^P*	45	2P2^2PP*
22	22PPP*	46	22P2^P*
23	2P2^P	47	2P2PP*P
24	22P^2P*	48	222^^2P*
25	2PP2^	49	22^P2^

OCRONs vom Typ 4 liefern eine kompakte Darstellung auch sehr großer Werte. Als Beispiel hier eine Tabelle der ersten Mersenne-Zahlen:

Tabelle 15: Mersenne-Zahlen, sowie die Exponenten in OCRON Typ4-Darstellung

n	Mersenne Prim-Exponent p	Mersenne Zahl $M_p = 2^p - 1$	OCRON4(p)
1	2	3	2
2	3	7	2P
3	5	31	2PP
4	7	127	22^P
5	11	2047	2PPP
6	13	8191	22P*P
7	17	131071	22^PP
8	19	524287	22P^P
9	23	8388607	2P2^P
10	29	536870911	22PP*P
11	31	2147483647	2PPPP
12	37	137438953471	22^2P*P
13	41	2199023255551	22P*PP
14	43	8796093022207	222^P*P
15	47	140737488355327	2P2PP*P
16	53	9007199254740991	222^^P

OCRONs mit Prim-Operator

17	59	576460752303423487	22^PPP
18	61	2305843009213693951	22P2^*P
19	67	147573952589676412927	22P^PP
20	71	2361183241434822606847	22^2PP*P
21	73	9444732965739290427391	2P22^P*P
22	79	604462909807314587353087	22PPP*P
23	83	9671406556917033397649407	2P2^PP
24	89	618970019642690137449562111	22P^2P*P

Tabelle 16: Mersenne-Zahlen in OCRON-Typ4-Darstellung

n	p	Mersenne Zahl $M_p = 2^p - 1$	OCRON4(M_p)
1	2	3	2P
2	3	7	22^P
3	5	31	2PPPP
4	7	127	2PPPPP
5	11	2047	2P2^P22P^2P*P*
6	13	8191	22^2PP2PPP*P*P
7	17	131071	22P^2P*22P2^*P*PP
8	19	524287	22PP*22^2P2P^*PP*P
9	23	8388607	2P2PP*P2PP2^P2P22P*P*P*P*
10	29	536870911	2P22^PP*P2PP22^2P*P*P*22^22PPP*P*P*
11	31	2147483647	2PP2P*22P^*PP2P2*P2PP*2P2^*P*2PP*P
12	37	137438953471	2P222^*P2P2^2*P22^P2*P*22^P2^*2PP2P^*2*P*
13	41	2199023255551	2P2^2*P2P2*P*2*P22^PP2PP*2P*P2PP2*P*2P2*P*2P*2*P*
14	43	8796093022207	2P2^PPP2PP2*PPP2*P*2PP22^*P2PP*P2*P*
15	47	140737488355327	22^P2PP*2*PP22^PP2P2^*22^*P*2PPPP2PP*2P*P2PPP*P2*P*
16	53	9007199254740991	2PP2*P2PP*PP22P^P2P*PP22^*P*2PP2*P2PP*22P^*P2P2^P*2P*2*P*
17	59	576460752303423487	22^P2*P22^*P222^*P2P22^P22P^*PP2P2P^*P2P*PP*2P2*P*2P2^*2*P*
18	61	2305843009213693951	2PP2P2^2*P*222^*P*22PPP*2P2*P2P*P*2P2^2*PP22^*P2P2^*P*P*P
19	67	147573952589676412927	22^22^2P*2PP*22P*P*22P2P^*P*P*P2PP2P^222^*P*PP*22P^2PP2^*22^*P*22P^*PP*P*
20	71	2361183241434822606847	2P22P^*PP*222P*P*P*22P2P2P22^*P*PP*P*P*2P2^22^*P*2P22P*P*P*2PPP22^*P*P*
21	73	9444732965739290427391	2PP22^*PP*P22PP^22P^*PP*22PPP*P*P*2P2^22PPP*P*2P2P^*2P22P^*PP*P*P*
22	79	604462909807314587353087	22P*2PP*22P*P*P22PPP*222^*P*P*2PP22^*P*P*P*22^2P*P222^2PP*P*P*22P*2P2^*P*P*P*
23	83	9671406556917033397649407	2P22P*P*P2P222^*P*P*222P2^*P*P*22P^2P2P^*P*22^2PP2P^*P*222^*P*22^2P2^*P*22P^222P*P*P*P*
24	89	618970019642690137449562111	?????

Tabelle 17: Wagstaff-Prim-Exponenten in OCRON-Typ4-Darstellung (Primzahlen in rot)

n	Wagstaff Prim-Exponent p	Wagstaff Zahl $\frac{M_p=2^p+1}{3}$	OCRON4(p)
1	2	5/3	2
2	3	3	2P
3	5	11	2PP
4	7	43	22^P
5	11	683	2PPP
6	13	2731	22P*P
7	17	43691	22^PP
8	19	174763	22P^P
9	23	2796203	2P2^P
10	29	178956971	22PP*P
11	31	715827883	2PPPP
12	37	45812984491	22^2P*P
13	41	733007751851	22P*PP
14	43	2932031007403	222^P*P
15	47	46912496118443	2P2PP*P
16	53	3002399751580331	222^^P

17	59	192153584101141163	22^PPP
18	61	768614336404564651	22P2^*P
19	67	49191317529892137643	22P^PP
20	71	787061080478274202283	22^2PP*P
21	73	3148244321913096809131	2P22^P*P
22	79	201487636602438195784363	22PPP*P

Tabelle 18: Wagstaff-Zahlen in OCRON-Typ4-Darstellung (Primzahlen in rot)

n	Wagstaff Prim-Exp. p	Wagstaff Zahl $\frac{w_p=2^p+1}{3}$	OCRON ₄ $\left(\frac{2^p+1}{3}\right)$
1	2	5/3	-
2	3	3	2P
3	5	11	2PPP
4	7	43	222^*P*P
5	11	683	22^2PPPPP*P
6	13	2731	2P22^*P*22P^*P*P
7	17	43691	22P^*22P^*22P*P*P*P
8	19	174763	222^*22P2P^*P*P*P
9	23	2796203	2PP22^*2P2^*P*2P22P^*P*P
10	29	178956971	22^*PPP222^*2P22^*2P*P2^*P*
11	31	715827883	22^*PPP22^*2P^*2PPP*22^*PP*PP*P
12	37	45812984491	2PP2^*2PPP*P22P*2PPP*2PP22PPP*P*PP*P*
13	41	733007751851	2P2^*PP22^*2P*2PP*22^*P*22P*P*22PPP*P*2P222^*P*P*P*
14	43	2932031007403	22PPP*222^*2P*P*P*P*2P2PP*P22^*P*P*P
15	47	46912496118443	22P2^*PP2PP222^*P*P*P2PP22P2^*P*P*2P2^*22P*PP*P*P*
16	53	3002399751580331	22^*22^*P*P22^*2P*22^*P*2P22^*PP*P*2P2P22^*2P*P*P*22PP2P^*P*P*P*
17	59	192153584101141163	2P2P2PPP*P*P22PPP*22P*PPP*P*2P22^*PP*22P^*22P2^*P*P*P*
18	61	768614336404564651	22^*222^*PP*PP*222^*PP*222^*P*222^*2P*22^*2P*P*P*P*P*P*P
19	67	49191317529892137643	22PP*22^*22P2P^*P*P*P*P222^*22P^*PP*2P22P^*P*222^*P*PPP*P*P*P*
20	71	787061080478274202283	22^*2P*22^*PP*2P2^*P2P2^*P*P*P222^*P*22PP*P*2PP22^*PP*P*222^*22P22P^*P*P*P*
21	73	3148244321913096809131	2P22^*P*22P*P*P22^*2P*PP22P*222PP*2PPP*P*PP*P*2P2PP*22P2^*P*2PP2PPP*P*P*P*P*
22	79	201487636602438195784363	?????

Die Längen der OCRONs bei Mersenne-Zahlen wachsen ungefähr proportional zu p:

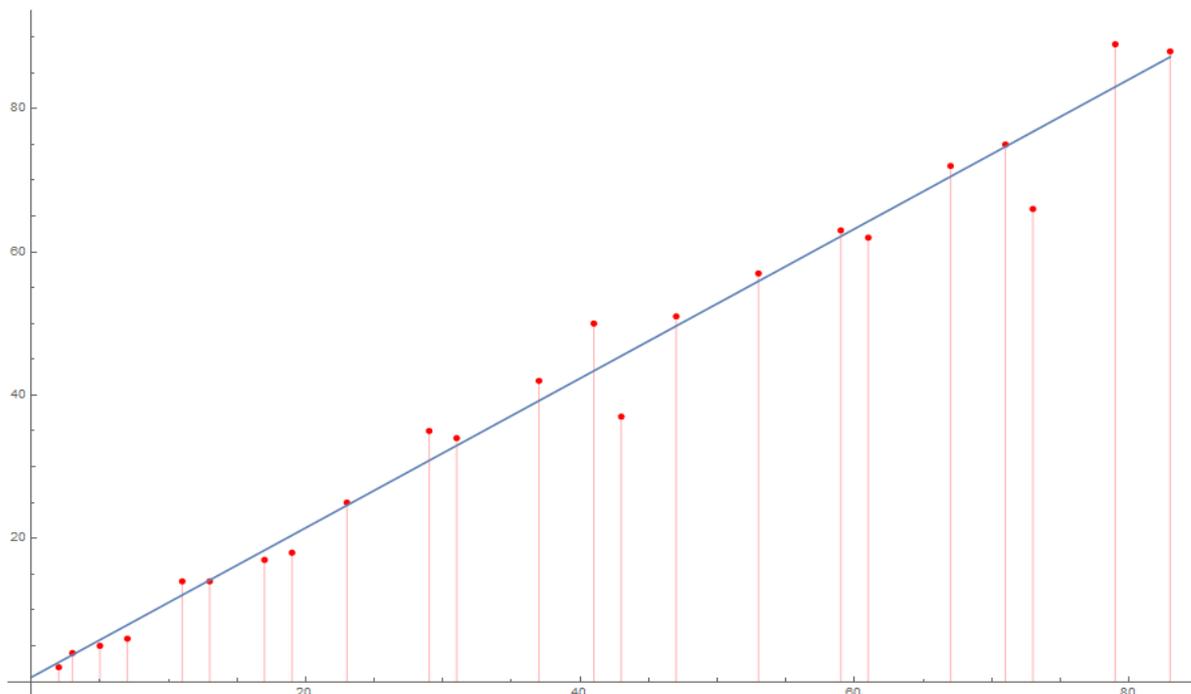


Abbildung 98: Längen der Typ4-OCRONs von Mersenne-Zahlen bis M_{83}

OCRONs mit Prim-Operator

Mathematica:

```
data={{2,2},{3,4},{5,5},{7,6},{11,14},{13,14},{17,17},{19,18},{23,25},
{29,35},{31,34},{37,42},{41,50},{43,37},{47,51},{53,57},{59,63},{61,62},
{67,72},{71,75},{73,66},{79,89},{83,88}}
line = Fit[data, {1,x},x]
Show[ListPlot[data,PlotStyle->Red, AxesLabel->Automatic,Filling->
Axis,PlotMarkers->Automatic],Plot[line,{x,0,83}]]
```

Extrapolieren wir für hohe Mersenne-Primzahlen, dann sind OCRON-Längen von einigen Millionen Zeichen zu erwarten (etwa dreimal so lang wie in Dezimaldarstellung, jedoch deutlich kürzer als in Binärdarstellung).

Die Längen der OCRONs bei Wagstaff-Zahlen wachsen ebenfalls ungefähr proportional zu p :

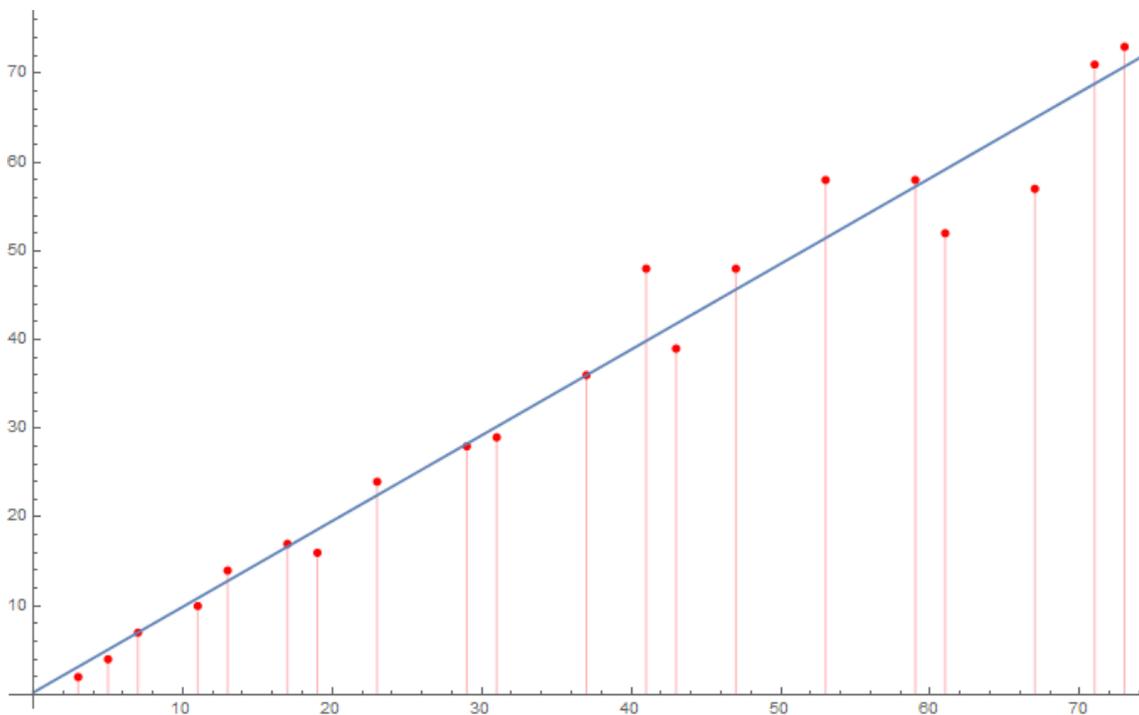


Abbildung 99: Längen der Typ4-OCRONs von Wagstaff-Zahlen bis W_{73}

Mathematica:

```
{{3,2},{5,4},{7,7},{11,10},{13,14},{17,17},{19,16},{23,24},{29,28},{31,29},
{37,36},{41,48},{43,39},{47,48},{53,58},{59,58},{61,52},{67,57},{71,71},
{73,73}}
line = Fit[data, {1,x},x]
Show[ListPlot[data,PlotStyle->Red, AxesLabel->Automatic,Filling->
Axis,PlotMarkers->Automatic],Plot[line,{x,0,73}]]
```

Extrapolieren wir für hohe Wagstaff-Primzahlen in ähnlich hohe Bereiche, wie die größten bekannten Mersenne Primzahlen, dann sind OCRON-Längen von einigen Millionen Zeichen zu erwarten.

Man beachte, dass die OCRONS bei Mersenne Primzahlen für die um 1 größere Zahl 2^p jeweils nur um 2 Zeichen länger sind als die der Primzahlexponent p selbst! Hier ein Beispiel:

Die Mersenne-Primzahl $M_{48} = 2^{57885161} - 1$ besitzt in Dezimaldarstellung 17425170 Stellen. Ihre Darstellung als Typ4-OCRON hat eine geschätzte Länge von ca. 60 Mio. Zeichen. Der Primexponent 57885161 besitzt die OCRON-Typ4-Darstellung $2PP2*P2PP*2P*P22^{\wedge}P*2P22^{\wedge\wedge}*2*P$ mit einer Länge von 29 Zeichen! $2^{57885161}$ besitzt die OCRON-Darstellung $22PP2*P2PP*2P*P22^{\wedge}P*2P22^{\wedge\wedge}*2*P^{\wedge}$ mit einer Länge von 31 Zeichen!

Noch erstaunlicher ist folgende Überlegung: Wir wollen annehmen, dass die Zahl $2^{57885161} + 1$ ($= 3 * \text{möglicher Wagstaff Kandidat}$) eine ähnliche Komplexität (mit ca. 60 Mio. Zeichen OCRON-Länge) besitzt wie sein ‚Mersenne-Nachbar‘ $2^{57885161} - 1$.

Nun kann man aufgrund der multiplikativen Eigenschaft der OCRONS die (unvorstellbar riesige) Zahl $2^{2^{57885161}+1}$ in OCRON-Typ4 -Darstellung einfach hinschreiben:

$$2^{2^{57885161}+1} = 2^{2^{57885161}} * 2 = \mathbf{222}PP2 * P2PP * 2P * P22^{\wedge}P * 2P22^{\wedge\wedge} * 2 * P^{\wedge} *$$

Mit einer OCRON-Länge von 33 Zeichen! Das könnte bedeuten, dass in der (vermutlich Millionen Zeichen langen) OCRON-Typ4-Darstellung von $2^{57885161} + 1$ sehr viel Redundanz vorhanden ist. Es könnte deshalb einen (bisher noch unbekannt) Algorithmus geben, der diese Redundanz eliminiert!

Wenn die obige Annahme über ähnlich hohe Komplexitäten richtig sein sollte, dann stellt sich schon die Frage, wieso $2^{2^{57885161}+1}$ eine um den Faktor 2 Millionen geringere Komplexität besitzt als sein scheinbar einfacherer Exponent $2^{57885161} + 1$! Damit diese Überlegung auch auf M_{48} und seine Zweierpotenz $2^{M_{48}}$ übertragen werden kann, müsste natürlich $2^{2^{57885161}-1}$ genauso einfach in $2^{2^{57885161}} * 2^{-1}$ zerlegt werden können. Dies wiederum würde eine Erweiterung des OCRON-Konzeptes auf negative ganze Zahlen bedeuten und wird Gegenstand weiterer Untersuchungen sein.

Kurioses:

Die bei den OCRONS verwendete sequentielle Operator-Darstellung ist ‚alternativlos‘. Hier ein Beispiel, wie das OCRON der 17-ten Mersenne-Primzahl M_{59} aussieht, wenn der Ausdruck ‚konventionell‘ dargestellt wird. Wir haben bis zu fünf Ebenen von verschachtelten Indices (sowohl in der Basis als auch in der Mantisse). Diese Zahl ist praktisch nicht mehr lesbar:

$$M_{59} = p_{p_2 p_{p_2 p_{p_2 p_2}} p_{p_2 p_2 p_2 p_2 p_2 p_2 (p_2)^2 (p_2)^2} p_{2 p_{p_2} p_{p_2 p_2}}$$

10.2.2.1 EIGENSCHAFTEN UND ERWEITERUNG VON TYP4-OCRONS: EOGRONS

Kommen wir zurück zu den wohlgeformten und nicht-wohlgeformten OCRONS:

Nicht alle OCRONs, die sich mit den Zeichen ' * ', '^', '2' und 'P' darstellen lassen sind ‚wohlgeformt‘ und ergeben einen Sinn, so dass sie abgearbeitet werden können (so z.B. die Sequenz $^{^p} * 222$ ist kein wohlgeformtes OCRON).

Es gibt nun eine Möglichkeit, in sinnvoller Weise diesen entarteten OCRONs auch eine Bedeutung und damit einem Zahlenwert zuzuordnen. Für die Interpretation eines OCRONs müssen nach den Regeln der ‚Polnischen‘ Notation die Symbole der OCRON-Sequenz auf den Stack ‚gepusht‘ werden bzw. abgearbeitet werden.

- Wir belegen den Stack mit unendlich vielen ‚virtuellen‘ ‚2‘-Symbolen vor, so dass im Fall von ‚unerwarteten‘ ‚*‘ oder ‚^‘ Symbolen (‚unerwartet‘ bedeutet, dass wir zum Zeitpunkt der Abarbeitung eines Operatorsymbols, das zwei Operanden benötigt, auf dem Stack weniger als 2 Werte haben) den Operator dennoch anwenden können.
- Sind am Ende der OCRON-Abarbeitung noch mehr als ein Stack-Eintrag vorhanden, so werden so viele virtuelle ‚*‘ angehängt (d.h. Multiplikationen mit links vorangestellten virtuellen ‚2‘en ausgeführt), bis der Stack nur noch einen Eintrag (das Ergebnis) hat.

Mit diesen Regeln, lassen sich nicht wohlgeformte OCRONs in normale OCRONs verwandeln. Damit bekommt jedes beliebige (auch nicht wohlgeformte) OCRON einen eindeutigen Wert. Umgekehrt lassen sich normale OCRONs verkürzen, indem wir die (immer vorhandene) führende ‚2‘ weglassen, und alle ‚*‘ am Ende entfernen, die schließlich den Stack bis auf einen Wert reduziert hätten. Dies hat den Vorteil, dass am Ende der OCRON-Abarbeitung so viele Stackeinträge übrig bleiben, wie Faktoren vorhanden sind (es sei denn unsere Zahl ist eine Potenzzahl (Zahl, die als Potenz geschrieben werden kann). Die Anzahl der Primfaktoren lässt sich aus den OCRONs auch ‚extrahieren‘, ohne auf die ‚Zahlen‘-Ebene gehen zu müssen. Wir können ohne Decodierung auf der formalen OCRON-Bedeutungsebene bleiben. Dies sieht praktisch so aus, dass wir die letzten ‚*‘s einfach entfernen (falls das Ende des OCRONs nur aus hintereinander kommenden ‚*‘ besteht; damit bleibt der Stack unverändert). Diese ‚gelöschten‘ Multiplikationen können ja hinterher (siehe oben, ‚virtuelle‘ ‚*‘s) wieder angehängt werden, damit wieder ein wohlgeformtes OCRON entsteht.

Nennen wir diese nicht wohlgeformten zusammen mit der Menge der wohlgeformten OCRONs ‚EOCRONs‘ (= Enhanced OCRONs). Beide Arten dürfen ineinander umgewandelt werden. Um aus einem nicht wohlgeformten EOCRON ein wohlgeformtes OCRON zu machen, muss es immer (nach links oder rechts) vergrößert werden.

Wir unterscheiden drei Typen von OCRONs: (Wohlgeformte OCRONs), standardisierte EOCRONs, beliebige EOCRONs

Eigenschaften oder Umformungsregeln von (wohlgeformten) OCRON4s:

- **Ein OCRON4 besteht aus einer beliebig langen Zeichenkette von OCRON4-Symbolen (2,P,^,*), die, wenn sie interpretiert werden, einen Wert ergeben.**

- **OCRON4s dürfen verkettet werden (d.h. aneinandergelängt, und am Ende mit einem ‚*‘ ergänzt. Diese Verkettung ist assoziativ und kommutativ und entspricht auf der Zahlenbedeutungsebene einer Multiplikation.**
- **Jedes OCRON4 (=wohlgeformt) beginnt mit einer ‚2‘ und endet mit einem ‚*‘, ‚^‘ oder ‚P‘. Das einzige OCRON4, das mit einer ‚2‘ enden kann ist ‚2‘ selbst.**
- **Außer der Verkettungsregel sind z. Zt. keine bedeutsamen weiteren Regeln bekannt (s. B. Additionsregel, oder Umformungsregeln)**
- **Die Zahl 1 besitzt keine Darstellung durch Typ4-OCRONs.**
- **Die Länge eines Typ4-OCRONs wächst typischerweise proportional zum Wert der entsprechenden Zahl.**
- **Der Unterschied zwischen der Maximallänge und der Minimallänge von OCRONs die zu einer Klasse von entarteten OCRONs (d.h. die denselben Zahlenwert ergeben) kann beliebig groß werden.**

Eigenschaften oder Umformungsregeln von (beliebigen) EOCRON4s:

- **Ein EOCRON4 besteht aus einer beliebig langen Zeichenkette von OCRON4-Symbolen (2,P,^,*), die beliebig angeordnet werden können.**
- **Damit ein nicht-wohlgeformtes EOCRON4 interpretiert werden kann, muss es in ein wohlgeformtes OCRON4 umgewandelt werden, dadurch tritt eine Vergrößerung auf.**
- **Es gibt ein leeres EOCRON4: Durch Umwandlung in ein OCRON4 ist dies die ‚2‘**
- **Es gibt EOCRONs, die bei Interpretation den gleichen Wert wie standardisierte EOCRONs ergeben, jedoch kürzer sind.**

Eigenschaften oder Umformungsregeln von (standardisierten) EOCRON4s:

- **Ein standardisiertes EOCRON4 besteht aus einer beliebig langen Zeichenkette von OCRON4-Symbolen (2,P,^,*).**
- **Damit ein standardisiertes EOCRON4 interpretiert werden kann, muss es in ein wohlgeformtes OCRON4 umgewandelt werden (evtl. durch Einfügen von ‚2‘-Symbolen am Anfang und Ergänzen von ‚*‘-Symbolen am Ende). Dadurch tritt eine Vergrößerung auf.**
- **Es gibt ein leeres standardisiertes EOCRON4: Durch Umwandlung in ein OCRON4 ist dies die ‚2‘**

OCRONs mit Prim-Operator

- Die Anzahl von Primfaktoren eines EOCRON4s ist einfach die Anzahl der Stack-Einträge nach Interpretation aller standardisierten EOCRON-Symbole (mit Einfügen der führenden ‚2‘ vor Umwandlung in ein wohlgeformtes OCRON)
- Jedes standardisierte EOCRON4 (=wohlgeformt) endet mit einem ‚^‘ oder ‚P‘. Es kann nie mit einem ‚*‘ oder ‚2‘ enden.
- Standardisierte EOCRONs können verkettet werden (entspricht einer Multiplikation), sind jedoch dann nicht mehr standardisiert
- Der Unterschied zwischen der Maximallänge und der Minimallänge von EOCRONs die zu einer Klasse von entarteten EOCRONs (d.h. die denselben Zahlenwert ergeben) kann beliebig groß werden.

Die letzte Eigenschaft deutet schon darauf hin, dass etwaige Umformungsregeln zwischen entarteten OCRONs oder EOCRONs nicht trivial sind!

Hier eine Typ4-EOCRON-Tabelle (erzeugt mit der Software: „kmatrix“⁵⁴, rot: ‚Minimal-EOCRONs blau hinterlegt: wohlgeformt):

Tabelle 19: Die ersten 100 Typ4-EOCRONs (in aufsteigender Reihenfolge)

Typ4-EOCRON	Wert (n)	Typ4-EOCRON	Wert (n)
*	4	^*2	16
P	3	^^^	256
2	4	^P*	14
^	4	^PP	17
P*	6	^P2	14
PP	5	^P^	128
P2	6	^2*	8
P^	8	^2P	12
2*	4	^22	16
2P	6	^2^	16
22	8	^^*	32
2^	4	^^P	53
^*	8	^^2	32
^P	7	^^^	65536
^2	8	P***	24
^^	16	P***P	37
P**	12	P**2	24
P*P	13	P**^	4096
P*2	12	P*P*	26
P*^	64	P*PP	41
PP*	10	P*P2	26
PPP	11	P*P^	8192
PP2	10	P*2*	12
PP^	32	P*2P	18
P2*	6	P*22	24
P2P	9	P*2^	36
P22	12	P*^^	128

⁵⁴ Kmatrix: <http://kmatrix.eu>

P2^	9	P*^P	311
P^*	16	P*^2	128
P^P	19	P*^^	18446744073709551616
P^2	16	PP**	20
P^^	256	PP*P	29
2**	8	PP*2	20
2*P	7	PP*^	1024
2*2	8	PPP*	22
2*^	16	PPPP	31
2P*	6	PPP2	22
2PP	10	PPP^	2048
2P2	12	PP2*	10
2P^	8	PP2P	15
22*	8	PP22	20
22P	12	PP2^	25
222	16	PP^*	64
22^	8	PP^P	131
2^*	8	PP^2	64
2^P	7	PP^^	4294967296
2^2	8	P2**	12
2^^	16	P2*P	13
^^*	16	P2*2	12
^*P	19	P2*^	64

10.2.2.2 ENTARTUNG VON TYP4-OCRONS

Entartung wurde unter 10.2.1.1 definiert. Der Grad der Entartung hängt entscheidend davon ab, wie ‚zusammengesetzt‘ die Zahl ist, d.h. wie viele Primfaktoren sie enthält. ‚Hoch‘ zusammengesetzte besitzen eine hohe OCRON-Entartung, Primzahlen dagegen oft eine sehr kleine. Manche Primzahlen haben den Entartungswert 1 (und somit keine Entartung). Hier eine kleine Tabelle der Entartungswerte der ersten 100 Typ4-OCRONS:

Tabelle 20: Entartungswerte der ersten 100 Typ4 OCRONS

n	Entart.	n	Entart.	n	Entart.	n	Entart.
1	0	26	4	51	4	76	40
2	1	27	5	52	16	77	4
3	1	28	16	53	18	78	24
4	2	29	2	54	34	79	2
5	1	30	12	55	2	80	156
6	2	31	1	56	68	81	18
7	2	32	57	57	10	82	4
8	5	33	2	58	4	83	2
9	2	34	4	59	2	84	144
10	2	35	4	60	72	85	4
11	1	36	46	61	8	86	8
12	8	37	8	62	2	87	4
13	2	38	10	63	16	88	34
14	4	39	4	64	220	89	34
15	2	40	34	65	4	90	72
16	18	41	2	66	12	91	8
17	2	42	24	67	5	92	16
18	8	43	4	68	16	93	2

OCRONs mit Prim-Operator

19	5	44	8	69	4	94	4
20	8	45	8	70	24	95	10
21	4	46	4	71	8	96	714
22	2	47	2	72	244	97	2
23	2	48	156	73	4	98	28
24	34	49	6	74	16	99	8
25	2	50	8	75	8	100	46

Eine Tabelle von Type4-OCRONs der ersten 25 natürlichen Zahlen, inklusive entarteter Ocronen sowie weitere Tabellen über Ocronen finden sich im Anhang.

Die Entartung wächst sehr schnell mit n (wenn auch nicht ganz so schnell wie bei Typ3-OCRONs), wie folgende Grafik zeigt:

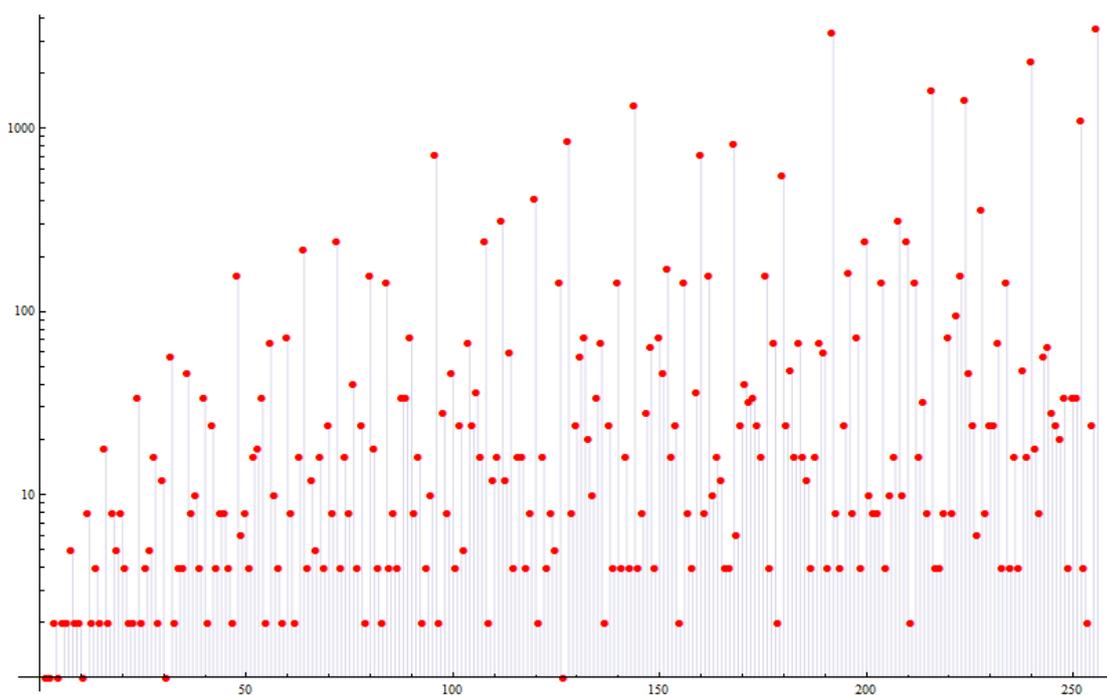


Abbildung 100: Entartung von wohlgeformten Ocron4s bis $n=256$ (logarithmisch dargestellt)

Mathematica:

```
data =  
Import["Primzahlen/data/ocron4_wohlgeformt_Degeneration_gut_bis_256.txt", "CSV"]  
ListLogPlot[data, PlotStyle->Red, AxesLabel->Automatic, Filling->  
>Axis, PlotMarkers->Automatic, PlotRange->All]
```

10.2.2.3 STANDARDISIERUNG VON TYP4-OCRONs UND -EOCRONs

Wegen des hohen Entartungsgrades dieser Ocron-Typen wollen wir aus den vielen möglichen (E)OCRON-Darstellungen den sogenannten Standard-Typ herausgreifen, der folgenden Ocron-Regeln entspricht:

- Die standardisierte Form sollte der Primfaktorenzerlegung entsprechen (d.h. jede Primzahl darf für eine Zerlegung nur einmal zusammen mit seinem Exponenten vorkommen).
- Wann immer Produkte auftauchen, so gilt die Regel der aufsteigenden Sortierung (zuerst die kleinen Faktoren, dann die großen Faktoren).
- ‚^‘ hat Priorität gegenüber dem ‚*‘-Operator, d.h.wann immer es möglich ist, wird statt dem ‚*‘ Operator der ‚^‘-Operator genommen (Beispiel: ‚22^‘ statt ‚22 * ‘
- Die Standardisierung sollte in reduzierter, minimaler Form sein (als minimales EOCCRON), in dem die Primfaktorenzerlegung aus den Stackwerten einfach abgelesen werden kann.

Bevor OCRONs zu EOCCRONs konvertiert werden, sollten sie in die standardisierte Form gebracht werden.

Hinweis: Die Umformung von beliebigen Typ4-OCRONS in standardisierte Typ4-OCRONS einzig allein auf Symbol-Ebene (d.h. ohne Auswertung als Zahl) ist ein schwieriges und ungelöstes Problem!

10.2.2.4 DIE GÖDELISIERUNG VON TYP4-OCRONS

Unter ‚Gödelisierung‘ verstehen wir einen Wechsel der Bedeutungsebene von einer funktionellen Betrachtungsweise (jedes OCRON stellt ein kleines Computerprogramm dar, mit dem sein Wert berechnet werden kann) in eine statische, ‚wert-definierte‘ Betrachtungsweise. Wir ordnen jedem OCRON (das zunächst einmal nur aus einer Kette von formalen Symbolen besteht) einen positiven, ganzzahligen Wert n zu. Diese Vorgehensweise stammt ursprünglich von dem Mathematiker Kurt Gödel, dem es mit dieser Methode gelang, sein berühmtes Unvollständigkeitstheorem zu beweisen.

Diese Zuordnung ist willkürlich und kann auf verschiedene Art und Weise erfolgen. Gemeinsam ist jedoch allen Methoden, dass jedem OCRON-Symbol (z.B. dem ‚^‘-Symbol) ein eindeutiger ganzzahliger Wert, der sogenannte Gödel-Code zugeordnet wird (z.B. der Wert 3 für das Symbol ‚^‘). Die gesamte OCRON-Kette erhält danach einen Gesamt-Zahlenwert, der aus einer arithmetischen Verknüpfung der einzelnen OCRON-Symbole entsteht. Auch hier haben wir völlige Freiheit, was die Wahl eines geeigneten Algorithmusses betrifft, der die einzelnen Gödel-Codes der OCRON-Symbole zu einem Gesamtwert zusammenfasst. Die Wahlfreiheit wollen wir etwas einschränken, indem wir fordern, dass die entstehenden Gödel-Werte möglichst klein werden, damit wir ihre evtl. vorhandenen arithmetischen Gesetze möglichst einfach untersuchen können. Bei der von Gödel verwendeten Gödelisierung entstehen astronomische hohe Werte, die für eine weitere arithmetische Untersuchung unbrauchbar sind. Nähere Informationen dazu findet der Leser in Hofstadters Buch ‚Gödel-Escher-Bach‘ (Hofstadter, 1991 / 1985)⁵⁵. Der Wechsel der Bedeutungsebene von formalen Symbolen (man spricht von ‚typogra-

⁵⁵ S. 288: Der Bumerang: Gödel-Numerierung von TNT

phisch') in die Welt der Zahlen hat es in sich (wir zitieren Hofstadter aus seinem berühmten Buch):

„Aus einem rein typographischen System in ein anderes isomorphes System überzuwechseln ist nicht sehr aufregend, während der Schritt aus dem typographischen Bereich in einen isomorphen Teil der Zahlentheorie gewisse noch unerforschte Möglichkeiten bietet. Es ist wie wenn jemand sein ganzes Leben Partituren gemacht hat – aber rein visuell – und ihn dann ganz unvermutet jemand darauf aufmerksam macht, dass Noten in der Partitur Abbilder von Tönen sind. Welch reiche neue Welt! Und dann ist es wieder so, wie wenn jemand sein ganzes Leben lang mit Fadenfiguren vertraut gewesen war, aber nur mit Fadenfiguren ohne Bedeutung – und dann, ganz plötzlich, jemand ihn auf die Entsprechung zwischen Geschichten und Fäden hingewiesen hätte. Was für eine Erleuchtung! Die Entdeckung der Gödel-Numerierung ist mit Descartes Entdeckung, dass zwischen Kurven in einer Ebene und Gleichungen mit zwei Variablen Isomorphie besteht, gleichgesetzt worden – unglaublich einfach, wenn man es einmal erkannt hat- aber eine Öffnung in eine unermessliche neue Welt.“

Hier ein paar denkbare Möglichkeiten der Gödelisierung:

Die einfache Primzahl-Gödelisierung⁵⁶:

Hierfür brauchen wir für jedes OCRON o der Länge l die ersten l Primzahlen, also z.B. zur Gödelisierung der 7 Zeichen langen Kette $,22^{\wedge}P2^*P^{\circ}$ die ersten 7 Primzahlen $P_n = 2,3,5,7,11,13,17$, sowie Gödel Codes $gc(\text{symbol})$ für die OCRON-Symbole (z.B. 1 für $,^*$, 2 für $,2^{\circ}$, 3 für $,P^{\circ}$ und 4 für $,^{\wedge}$). Die Gödel-Codes müssen einen ganzzahligen Wert > 0 besitzen. Der Wert 0 ist nicht erlaubt. Der Wert entsteht dann dadurch, dass die Faktoren $P_n^{gc(\text{symbol})}$ mit einander multipliziert werden (wobei n bis zur OCRON-Länge $l - 1$ läuft):

Beispiel: Das OCRON $,22^{\wedge}P2^*P^{\circ}$ (entspricht dem Zahlenwert 43) hat

- An erster Stelle den Wert $P_1^{gc(2)} = 2^2 = 4$
- An zweiter Stelle den Wert $P_2^{gc(2)} = 3^2 = 9$
- An dritter Stelle den Wert $P_3^{gc(^{\wedge})} = 5^4 = 625$
- An vierter Stelle den Wert $P_4^{gc(P)} = 7^3 = 343$
- An fünfter Stelle den Wert $P_5^{gc(2)} = 11^2 = 121$
- An sechster Stelle den Wert $P_6^{gc(^{\circ})} = 13^1 = 13$
- An siebter Stelle den Wert $P_7^{gc(P)} = 17^3 = 4913$

Der Gesamtwert $g(o)$ (Gödelnummer von $22^{\wedge}P2^*P^{\circ}$) ergibt sich dann zu:

$$g(o) = g(22^{\wedge}P2^*P^{\circ}) = 4 * 9 * 625 * 343 * 121 * 13 * 4913 = 59641989907500$$

Oder ganz allgemein (mit l =Länge des OCRONs)

$$g(o) = \prod_{n=1}^l P_n^{gc(\text{OCRON}[n-1])} \quad (151)$$

⁵⁶ <https://de.wikipedia.org/wiki/Gödelnummer>

Aus der Primfaktorzerlegung von 59641989907500 läßt sich eindeutig das OCRON $,22^{\wedge}P2*P^{\prime}$ und damit schliesslich die ursprüngliche Zahl 43 rekonstruieren.

Der Vorteil dieser Methode ist, die kleine Anzahl der ‚Freiheitsgrade‘ (nämlich 4 für die Wahl der Gödel-Codes der OCRONs) und die Unabhängigkeit in der Darstellung von einem etwaigen Stellenwertsystem (z.B. Dezimalsystem oder Binärsystem). Der Nachteil ist klar: Wir bekommen unhandlich große Zahlen für die Gödelnummern, die zudem noch schwer zu dekodieren sind. Der entscheidende Nachteil ist jedoch, dass es sehr viele Zahlen gibt, die keiner Gödelnummer entsprechen (und daher nicht in ein OCRON umgewandelt werden können), nämlich alle Zahlen, die eine Primfaktorenzerlegung besitzen, die nicht der lückenlosen Reihenfolge der ersten n -Primzahlen entspricht, oder Zahlen in deren Zerlegung eine Primfaktorenpotenz vorkommt, die grösser ist als alle vorkommenden Gödel-Codes unserer OCRONs, so wäre beispielsweise $32 = 2^5$ kein gültiger Gödel-Code.

Die einfache Primzahl-Gödelisierung stellt eine injektive Abbildung der Menge der OCRONs auf die Menge der positiven natürlichen Zahlen \mathbb{N}_+ dar.

Die differenzielle Primzahl-Gödelisierung

Bei dieser Methode brauchen wir mehr als die ersten l Primzahlen, wobei l für die OCRON-Länge steht, sowie die festen Gödel-Codes 1,2,3 und 4 für die Symbole $,*^{\prime},^{\prime}2^{\prime},^{\prime}P^{\prime},^{\prime}\wedge^{\prime}$ (wobei die Zuordnung willkürlich ist, so dass wir hier auch $4! = 24$ mögliche Code-Zuordnungen haben). Mit welchem Algorithmus lässt sich nun aus einer beliebig langen OCRON-Kette, die aus den Symbolen $,*^{\prime},^{\prime}2^{\prime},^{\prime}P^{\prime},^{\prime}\wedge^{\prime}$ mit den entsprechenden Gödel-Codes (z.B. 1,2,3,4) besteht, eine eindeutige Gödel-Nummerierung konstruieren?

Indem wir einen Gödel-Code als Offset von Indices in der Primzahl-Tabelle behandeln. Für das obige Beispiel ergibt sich hiermit:

Das OCRON $,22^{\wedge}P2*P^{\prime}$ hat (mit leicht veränderter Gödel-Code Zuordnung $* \rightarrow 0, P \rightarrow 1, 2 \rightarrow 2, \wedge \rightarrow 3$):

- An erster Stelle den Wert $P_{0+gc(2)} = P_2 = 3$
- An zweiter Stelle den Wert $P_{2+gc(2)} = P_4 = 7$
- An dritter Stelle den Wert $P_{4+gc(\wedge)} = P_7 = 17$
- An vierter Stelle den Wert $P_{7+gc(P)} = P_8 = 19$
- An fünfter Stelle den Wert $P_{8+gc(2)} = P_{10} = 29$
- An sechster Stelle den Wert $P_{10+gc(*)} = P_{10} = 29$
- An siebter Stelle den Wert $P_{10+gc(P)} = P_{11} = 31$

Der Gesamtwert $g(o)$ (Gödelnummer von $22^{\wedge}P2*P$) ergibt sich dann zu:

$$g(o) = g(22^{\wedge}P2*P) = 3 * 7 * 17 * 19 * 29^2 * 31 = 176839593$$

Das sieht schon etwas besser aus, diese Methode hat jedoch noch immer die bei der letzten Methode beschriebenen Nachteile.

Gödelisierung mit Hilfe von Stellenwertsystemen

Hier ersetzen wir einfach die Symbole der OCRON-Ketten durch die jeweiligen Gödel-Codes, und erhalten z.B. aus $22^{\wedge}P2^*P$ die Gödelnummer ‚2231201‘ die am besten im Vierer-System repräsentiert wird: 2231201_4 . Diese Codierung ist in beiden Richtungen (OCRON->Gödelnummer und Gödelnummer->OCRON) leicht durchzuführen, und sie hat einen entscheidenden Vorteil: Sie ist bijektiv, in beiden Richtungen eindeutig, d.h. zu jeder beliebigen Gödelnummer existiert auch ein eindeutiges EOCRON (wir schreiben hier extra EOCRON), da die Konvertierung oft nicht-wohlgeformte OCRONs (=EOCRONs) ergibt, die nach der Methode wie in 10.2.2.1 beschrieben leicht in wohlgeformte OCRONs umgewandelt werden können. **Mit dieser Gödelisierungsmethode haben wir nun eine Möglichkeit, beliebige Zahlen mit Hilfe dieser „Gödel-Transformation“ in eine Gödel-Nummer (in der irgendwie das Konstruktionsprinzip dieser Zahl steckt) zu transformieren. Eine Transformation, die uns in eine andere Zahlenwelt führt, die einen Bedeutungswechsel der Zahlen beinhaltet!**

Hier ein paar Tabellen, damit man eine Vorstellung von den abstrakten Beschreibungen bekommt (mit $o(n) = \text{OCRON}(n)$ und $g(n) = g(\text{OCRON}(n)) = \text{Gödelnummer}$):

Eigenschaften von EGOCRONs:

Man beachte, dass die Gödelnummern („EGOCRONs“) fast immer größer sind als ihre ursprünglichen Zahlenwerte, von denen sie stammen und außerdem immer ungerade sind.

Tabelle 21: Typ4-EOCRONs (Standarddarstellung) und Gödelnummern von 2 bis 100.
GödelCodes: (“*”=0, “P”=1, “2”=2, “^”=3)

n	o(n)	g(n)	N	o(n)	g(n)
2	(Leer)	(-)	51	P22^PP	1717
3	P	1	52	2^22P*P	11921
4	2^	11	53	22^^P	701
5	PP	5	54	2P2P^	615
6	2P	9	55	PP2PPP	1429
7	2^P	45	56	2P^22^P	10157
8	2P^	39	57	P22P^P	1693
9	P2^	27	58	22PP*P	2641
10	2PP	37	59	2^PPP	725
11	PPP	21	60	2^2P2PP	11877
12	2^2P	185	61	2P2^*P	2481
13	2P*P	145	62	2PPPP	597
14	22^P	173	63	P2^22^P	7085
15	P2PP	101	64	22P*^	659
16	22^^	175	65	PP22P*P	5777
17	2^PP	181	66	2P2PPP	2453
18	2P2^	155	67	2P^PP	629
19	2P^P	157	68	2^22^PP	11957
20	2^2PP	741	69	P2P2^P	1645
21	P22^P	429	70	2PP22^P	9645
22	2PPP	149	71	2^2PP*P	11857
23	P2^P	109	72	2P^2P2^	10139

24	2P^2P	633	73	P22^P*P	6865
25	PP2^	91	74	22^2P*P	11153
26	22P*P	657	75	P2PP2^	1627
27	P2P^	103	76	2^22P^P	11933
28	2^22^P	2989	77	2^P2PPP	11669
29	2PP*P	593	78	2P22P*P	9873
30	2P2PP	613	79	2PPP*P	2385
31	PPPP	85	80	22^^2PP	11237
32	2PP^	151	81	P22^^	431
33	P2PPP	405	82	22P*PP	2629
34	22^PP	693	83	P2^PP	437
35	PP22^P	1453	84	2^2P22^P	47533
36	2^2P2^	2971	85	PP22^PP	5813
37	2^2P*P	2961	86	222^P*P	10961
38	22P^P	669	87	P22PP*P	6737
39	P22P*P	1681	88	2P^2PPP	10133
40	2P^2PP	2533	89	2P^2P*P	10129
41	2P*PP	581	90	2P2^2PP	9957
42	2P22^P	2477	91	2^P22P*P	46737
43	22^P*P	2769	92	2^2P2^P	11885
44	2^2PPP	2965	93	P2PPPP	1621
45	P2^2PP	1765	94	2P2PP*P	9809
46	2P2^P	621	95	PP22P^P	5789
47	P2PP*P	1617	96	2PP^2P	2425
48	22^^2P	2809	97	PP2^P	365
49	2^P2^	731	98	22^P2^	2779
50	2PP2^	603	99	P2^2PPP	7061

Eigenschaften von inversen EGOCRONs:

Man beachte, dass die inversen Zahlenwerte der Gödelnummern („inverse EGOCRONs“) fast immer kleiner sind als ihre ursprünglichen Gödelnummern, von denen sie stammen.

Tabelle 22: Typ4-EOCRONs (Inverse Gödelisierung von g=0 bis 99)

Gödelzahl g	EOCRON zu g	n	Gödelzahl g	EOCRON zu g	n
0	*	4	50	^*2	16
1	P	3	51	^^^	256
2	2	4	52	^P*	14
3	^	4	53	^PP	17
4	P*	6	54	^P2	14
5	PP	5	55	^P^	128
6	P2	6	56	^2*	8
7	P^	8	57	^2P	12
8	2*	4	58	^22	16
9	2P	6	59	^2^	16
10	22	8	60	^^*	32
11	2^	4	61	^^P	53
12	^*	8	62	^^2	32
13	^P	7	63	^^^	65536
14	^2	8	64	P***	24
15	^^	16	65	P**P	37
16	P**	12	66	P**2	24

OCRONs mit Prim-Operator

17	P*P	13	67	P**^	4096
18	P*2	12	68	P*P*	26
19	P*^	64	69	P*PP	41
20	PP*	10	70	P*P2	26
21	PPP	11	71	P*P^	8192
22	PP2	10	72	P*2*	12
23	PP^	32	73	P*2P	18
24	P2*	6	74	P*22	24
25	P2P	9	75	P*2^	36
26	P22	12	76	P**^*	128
27	P2^	9	77	P**^P	311
28	P^*	16	78	P**^2	128
29	P^P	19	79	P**^^	18446744073709551616
30	P^2	16	80	PP**	20
31	P^^	256	81	PP*P	29
32	2**	8	82	PP*2	20
33	2*P	7	83	PP*^	1024
34	2*2	8	84	PPP*	22
35	2*^	16	85	PPPP	31
36	2P*	6	86	PPP2	22
37	2PP	10	87	PPP^	2048
38	2P2	12	88	PP2*	10
39	2P^	8	89	PP2P	15
40	22*	8	90	PP22	20
41	22P	12	91	PP2^	25
42	222	16	92	PP^*	64
43	22^	8	93	PP^P	131
44	2^*	8	94	PP^2	64
45	2^P	7	95	PP^^	4294967296
46	2^2	8	96	P2**	12
47	2^^	16	97	P2*P	13
48	^**	16	98	P2*2	12
49	^*P	19	99	P2*^	64

Man mag sich fragen, wie die Häufigkeiten der “*”, “P“, “2“ und “^“-Symbole ist. Eine statistische Auswertung der EOCRONs vom Typ 4 für die natürlichen Zahlen von 3 bis 10000 ergibt folgende Häufigkeiten:

Gesamtanzahl der Symbole: 123952
davon P-Symbole: 52664 (entspricht 42,487 %)
davon 2-Symbole: 42794 (entspricht 34,525 %)
davon ^-Symbole: 16711 (entspricht 13,482 %)
davon *-Symbole: 11783 (entspricht 9,506 %)

Folgende Grafik zeigt eine logarithmische Darstellung der Gödelnummern für die ersten 5000 natürlichen Zahlen bei Verwendung der GödelCodes “*”=0, “P“=1, “2“=2, sowie “^“=3. Bei der Gödelisierung wurde das Vierer-Stellenwertsystem verwendet. Man kann deutlich eine ‚Bänderstruktur‘ erkennen.

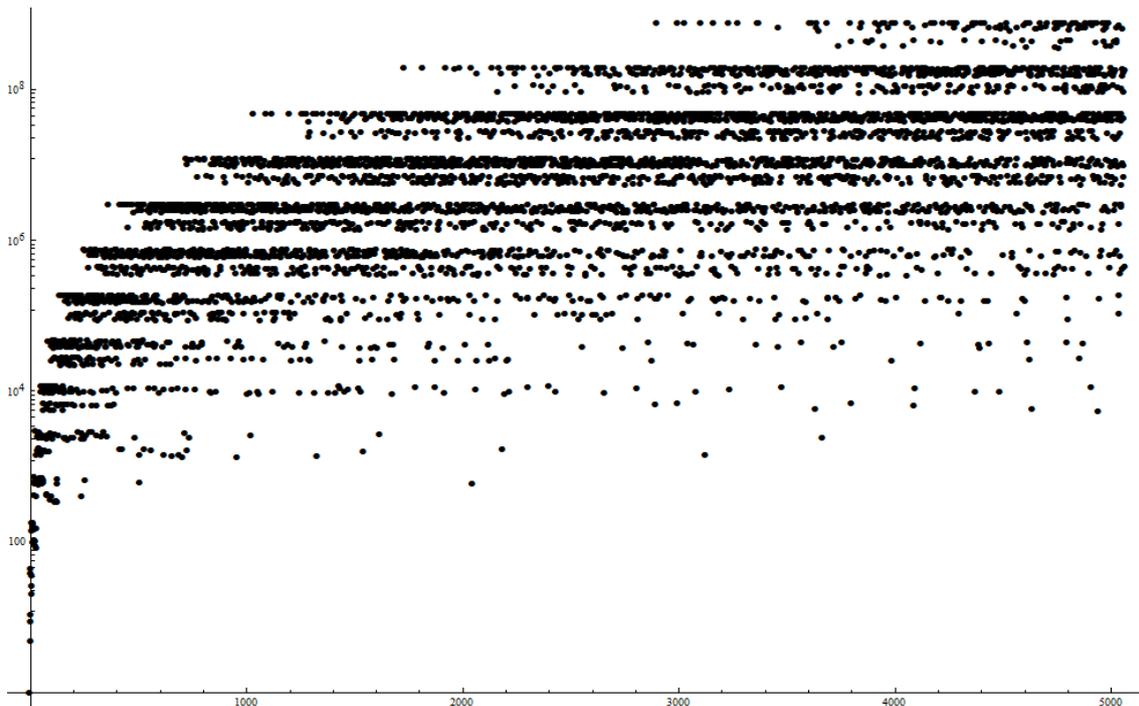


Abbildung 101: Logarithmische Darstellung der Gödelnummern für die ersten 5000 natürlichen Zahlen bei Verwendung der GödelCodes $"*" = 0$, $"P" = 1$, $"2" = 2$, sowie $"^" = 3$

Mathematica:

```
data=Import["Primzahlen/data/EGOCRONSTyp8_3.txt",{"Data",All,{1}}];
ListLogPlot[{data,{All}{1}},PlotStyle->Black,PlotMarkers->Automatic,
AxesLabel->Automatic,PlotRange->All,ImageSize->Large]
```

Und hier im Bereich zwischen 1 und 200:

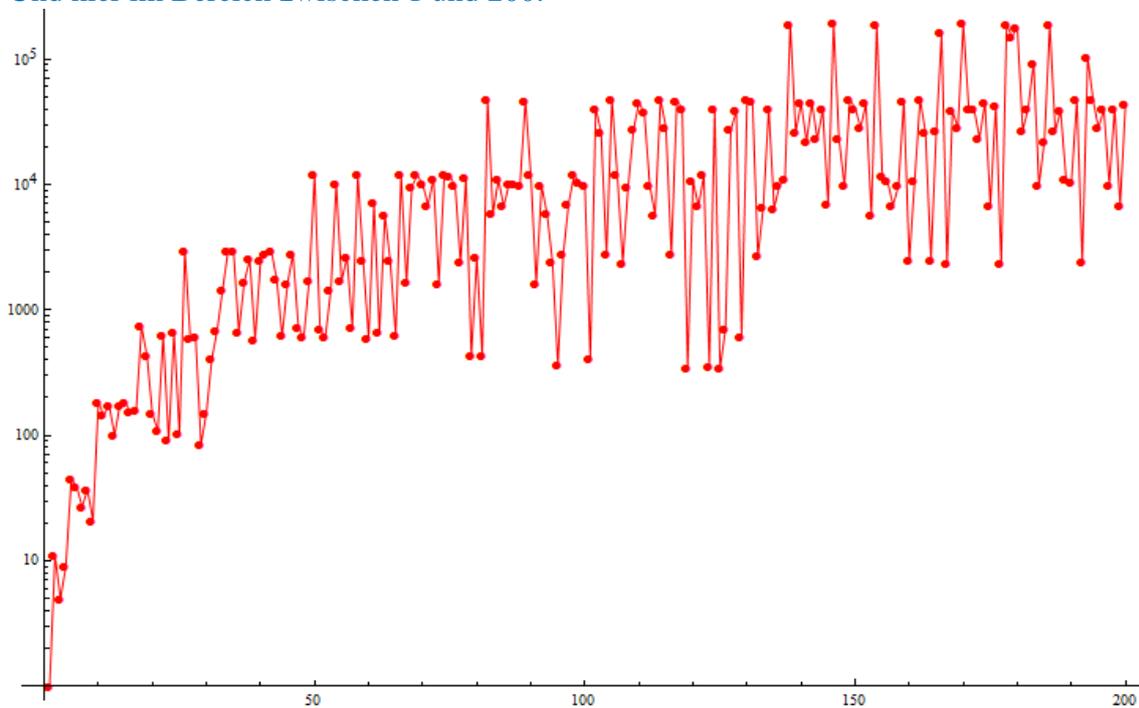


Abbildung 102: Logarithmische Darstellung der Gödelnummern für die ersten 200 natürlichen Zahlen bei Verwendung der GödelCodes $"*" = 0$, $"P" = 1$, $"2" = 2$, sowie $"^" = 3$

10.2.3 OCRONS MIT PRIM-"P", "*"-"^"- UND "Q"-OPERATOR

Der Q-Operator ersetzt mehrfach hintereinander vorkommende ‚P‘-Operatoren durch die Symbolkette <n>Q, wobei *n* in der entsprechenden OCRON-Codierung dargestellt wird und die Anzahl von hintereinanderfolgenden ‚P‘'s enthält.

10.2.3.1 ENTARTUNG VON TYP5-OCRONS

Entartung wurde unter 10.2.1.1 definiert. Die Entartung wächst sehr schnell mit *n* wie folgende Grafik zeigt:

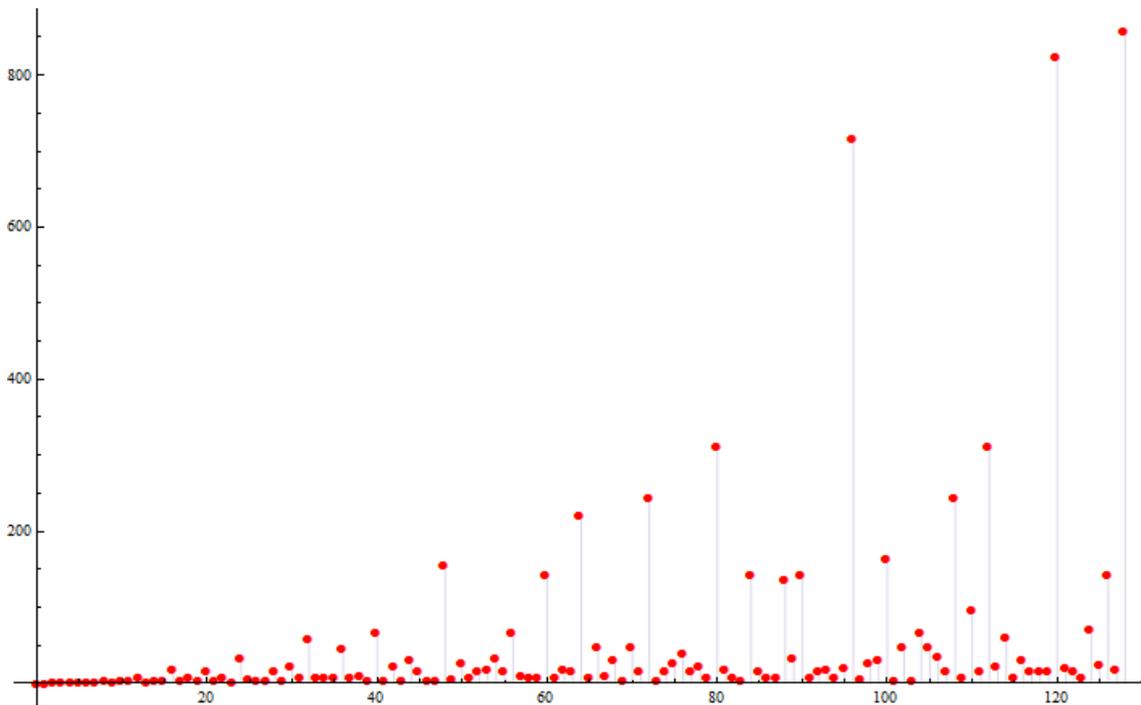


Abbildung 103: Entartung von wohlgeformten OCRON5s bis n=128

```
Mathematica:  
data =  
Import["Primzahlen/data/ocron5_wohlgeformt_Degeneration_gut_bis_128.tx  
t", "CSV"]  
ListPlot[data, PlotStyle->Red, AxesLabel->Automatic, Filling->  
>Axis, PlotMarkers->Automatic, PlotRange->All]
```

10.2.4 OCRONS MIT PRIM- UND NICHT-PRIM-OPERATOR

Dieser Typ von OCRON kommt mit nur zwei Operatoren aus: Dem Primzahl-Operator ‚P‘ und dem Nicht-Primzahl-Operator, den wir der Einfachheit halber auch mit ‚*‘ bezeichnen wollen (nicht zu verwechseln mit dem Multiplikationsoperator ‚*‘ der OCRON-Typen 3 bis 5. Die Interpretation der Operatoren ist die gleiche wie bei den

Typ3, Typ4 und Typ5-Operatoren: Sei n der aktuelle Zahlenwert, der aktuell in der untersten ‚Stack-Schublade‘ steht. Der ‚P‘-Operator berechnet die n -te Primzahl und überschreibt damit den Stackwert. Der ‚*‘-Operator berechnet die n -te Nicht-Primzahl und überschreibt damit ebenfalls den untersten Stackwert. Auf diese Weise läßt sich jede Zahl $n \geq 1$ als Folge von ‚*‘- und ‚P‘-Operatoren schreiben. Man beachte, dass kein Multiplikationsoperator oder Potenzoperator mehr vorkommt! Der neue Satz von OCRONs bekommt den Typ 6. Zur Berechnung der OCRONs ist nur noch wichtig, dass zu Beginn der Stack mit dem Wert 1 vorbelegt wird, so dass bei der Abarbeitung ‚*‘ den Wert 1 ergibt (erste Nicht-Primzahl) und ‚P‘ den Wert 2 ergibt (erste Primzahl). Die Null besitzt keine Entsprechung in der ‚P *‘-OCRON-Codierung. Prime OCRONs besitzen eine Reihe von sehr interessanten, bemerkenswerten Eigenschaften:

- 1) Die ‚P *‘-Darstellung ist ein-eindeutig (bijektiv), d.h. zu jeder Zahl gibt es eine eindeutige ‚P *‘-Darstellung und umgekehrt! Mit Hilfe der ‚P *‘-OCRON Darstellung läßt sich die Menge der natürlichen Zahlen umordnen, und das auf eindeutige Weise.
- 2) Eine direkte Folge ist, dass es keine Entartung mehr gibt, und dass die entsprechenden Gödelnummern überschaubar klein bleiben.

Zum Verdeutlichen hier die ersten 100 P-* -OCRONs, zusammen mit ihren Gödelnummern, basierend auf den Gödelcodes ‚*‘=0 und ‚P‘=1; für die Gödelnummer GN(g) verwenden wir auch die Bezeichnung ‚GOCRON‘ (=Gödelized OCRON).

Tabelle 23: Prime-OCRONs (P und*-Operator) mit Gödelnummern (GCodes P=1, *=0)

N	G=OCRON6(N)	GN(g)	N	G=OCRON6(N)	GN(g)
0	(-)	(-)	50	PPP*****	224
1	*	0	51	PP*****	192
2	P	1	52	P*P*P*	42
3	PP	3	53	PP***P	49
4	P*	2	54	P*PP***	88
5	PPP	7	55	PPPP*****	240
6	PP*	6	56	P**P***	72
7	P*P	5	57	PP*PP*	54
8	P**	4	58	PP**P**	100
9	PPP*	14	59	P*PPP	23
10	PP**	12	60	P***P*	34
11	PPPP	15	61	PPPP*P	61
12	P*P*	10	62	P*P*****	160
13	PP*P	13	63	PPPPP**	124
14	P***	8	64	PP*P*****	208
15	PPP**	28	65	PPP**P*	114
16	PP***	24	66	P*****	128
17	P*PP	11	67	P**PP	19
18	PPPP*	30	68	PPP*P***	232
19	P**P	9	69	PPP*****	448
20	P*P**	20	70	PP*****	384
21	PP*P*	26	71	P*P**P	41
22	P****	16	72	P*P*P**	84
23	PPP*P	29	73	PP*P*P	53
24	PPP***	56	74	PP***P*	98
25	PP*****	48	75	P*PP*****	176
26	P*PP*	22	76	PPPP*****	480
27	PPPP**	60	77	P**P****	144

OCRONs mit Prim-Operator

28	P**P*	18	78	PP*PP**	108
29	PP**P	25	79	P****P	33
30	P*P***	40	80	PP**P***	200
31	PPPPP	31	81	P*PPP*	46
32	PP*P**	52	82	P***P**	68
33	P*****	32	83	PPP*PP	59
34	PPP*P*	58	84	PPPP*P*	122
35	PPP*****	112	85	P*P*****	320
36	PP*****	96	86	PPPPP***	248
37	P*P*P	21	87	PP*P*****	416
38	P*PP**	44	88	PPP**P**	228
39	PPPP***	120	89	PPP***P	113
40	P**P**	36	90	P*****	256
41	PP*PP	27	91	P**PP*	38
42	PP**P*	50	92	PPP*P****	464
43	P***P	17	93	PPP*****	896
44	P*P****	80	94	PP*****	768
45	PPPPP*	62	95	P*P**P*	82
46	PP*P***	104	96	P*P*P***	168
47	PPP**P	57	97	PP***P	97
48	P*****	64	98	PP*P*P*	106
49	PPP*P**	116	99	PP***P**	196

Mathematica (Berechnung $n \rightarrow \text{GOCRON}(n)$):
Bitte den Autor kontaktieren.

Man beachte, dass in der Binärdarstellung der Gödelnummern $\text{GN}(g)$ Primzahlen immer mit einer ‚1‘ enden und zusammengesetzte Zahlen immer mit einer ‚0‘! Oder in der Dezimalschreibweise: Primzahlen haben immer eine ungerade Gödelnummer, zusammengesetzte Zahlen immer eine gerade Gödelnummer! Die entstehenden Gödelnummern sind nicht ganz unbekannt; auf der Internetseite <https://oeis.org> tauchen sie unter der Nummerierung A071574 und A237739 auf.⁵⁷

⁵⁷ <https://oeis.org/A071574>

Hier ein logarithmischer Plot mit Prime GOCRONs:

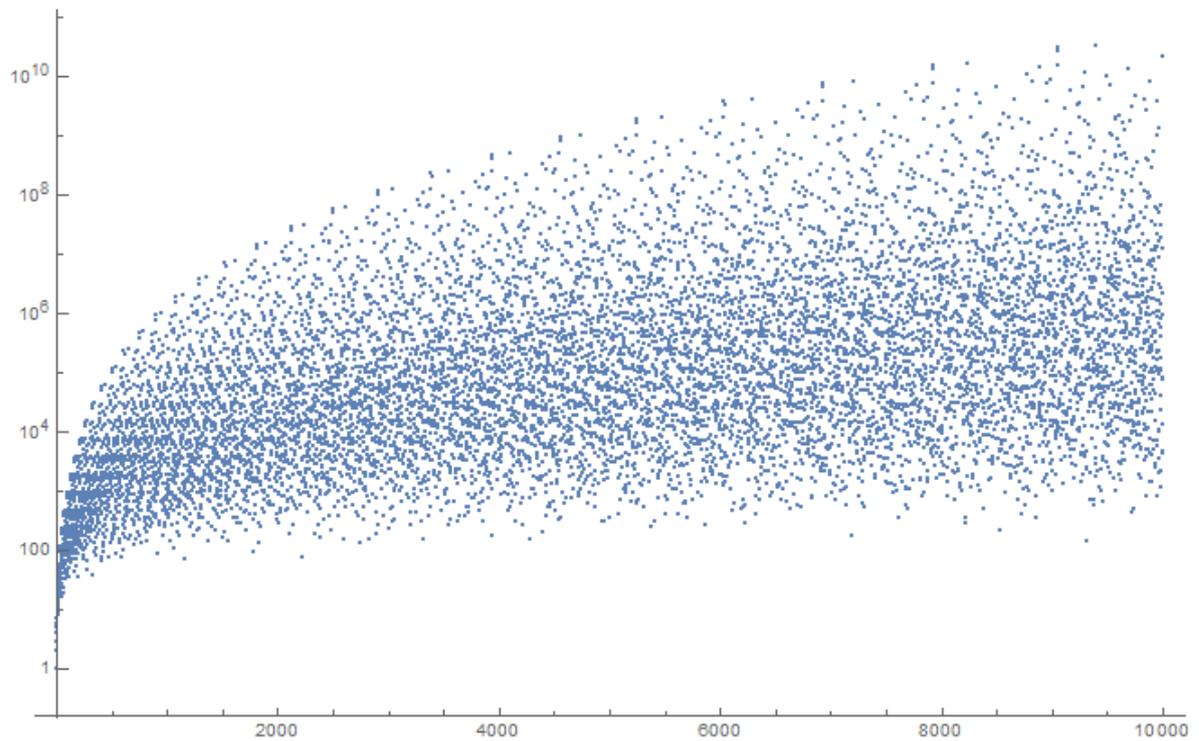


Abbildung 104: Prime-GOCRONs Typ6 ($n \rightarrow GOCRON[n]$) von 1 bis 10000

OCRONs mit Prim-Operator

Die folgende Tabelle stellt die inverse Tabelle zu Tabelle 23 dar.

Tabelle 24: Prime GOCRONs, OCRONs und dazu inverse Zahlen von 0 bis 99

GOCRON	OCRON	N	GOCRON	OCRON	N
0	*	1	50	PP**P*	42
1	P	2	51	PP**PP	109
2	P*	4	52	PP*P**	32
3	PP	3	53	PP*P*P	73
4	P**	8	54	PP*PP*	57
5	P*P	7	55	PP*PPP	179
6	PP*	6	56	PPP***	24
7	PPP	5	57	PPP**P	47
8	P***	14	58	PPP*P*	34
9	P**P	19	59	PPP*PP	83
10	P*P*	12	60	PPPP**	27
11	P*PP	17	61	PPPP*P	61
12	PP**	10	62	PPPPP*	45
13	PP*P	13	63	PPPPPP	127
14	PPP*	9	64	P*****	48
15	PPPP	11	65	P*****P	137
16	P****	22	66	P****P*	106
17	P***P	43	67	P****PP	401
18	P**P*	28	68	P***P**	82
19	P**PP	67	69	P***P*P	281
20	P*P**	20	70	P***PP*	244
21	P*P*P	37	71	P***PPP	1153
22	P*PP*	26	72	P**P***	56
23	P*PPP	59	73	P**P**P	173
24	PP***	16	74	P**P*P*	141
25	PP**P	29	75	P**P*PP	587
26	PP*P*	21	76	P**PP**	121
27	PP*PP	41	77	P**PP*P	467
28	PPP**	15	78	P**PPP*	411
29	PPP*P	23	79	P**PPPP	2221
30	PPPP*	18	80	P*P****	44
31	PPPPP	31	81	P*P***P	113
32	P*****	33	82	P*P**P*	95
33	P****P	79	83	P*P**PP	353
34	P***P*	60	84	P*P*P**	72
35	P***PP	191	85	P*P*P*P	239
36	P**P**	40	86	P*P*PP*	203
37	P**P*P	107	87	P*P*PPP	919
38	P**PP*	91	88	P*PP***	54
39	P**PPP	331	89	P*PP**P	163
40	P*P***	30	90	P*PP*P*	133
41	P*P**P	71	91	P*PP*PP	547
42	P*P*P*	52	92	P*PPP**	110
43	P*P*PP	157	93	P*PPP*P	419
44	P*PP**	38	94	P*PPPP*	345
45	P*PP*P	101	95	P*PPPPP	1787
46	P*PPP*	81	96	PP*****	36
47	P*PPPP	277	97	PP****P	97
48	PP****	25	98	PP***P*	74
49	PP**P*	53	99	PP**PP	241

Mathematica-Programm (Berechnung GOCRON->n (invers): Bitte den Autor kontaktieren.

Hier ein logarithmischer Plot der inversen Prime GOCRONs (Typ 6):

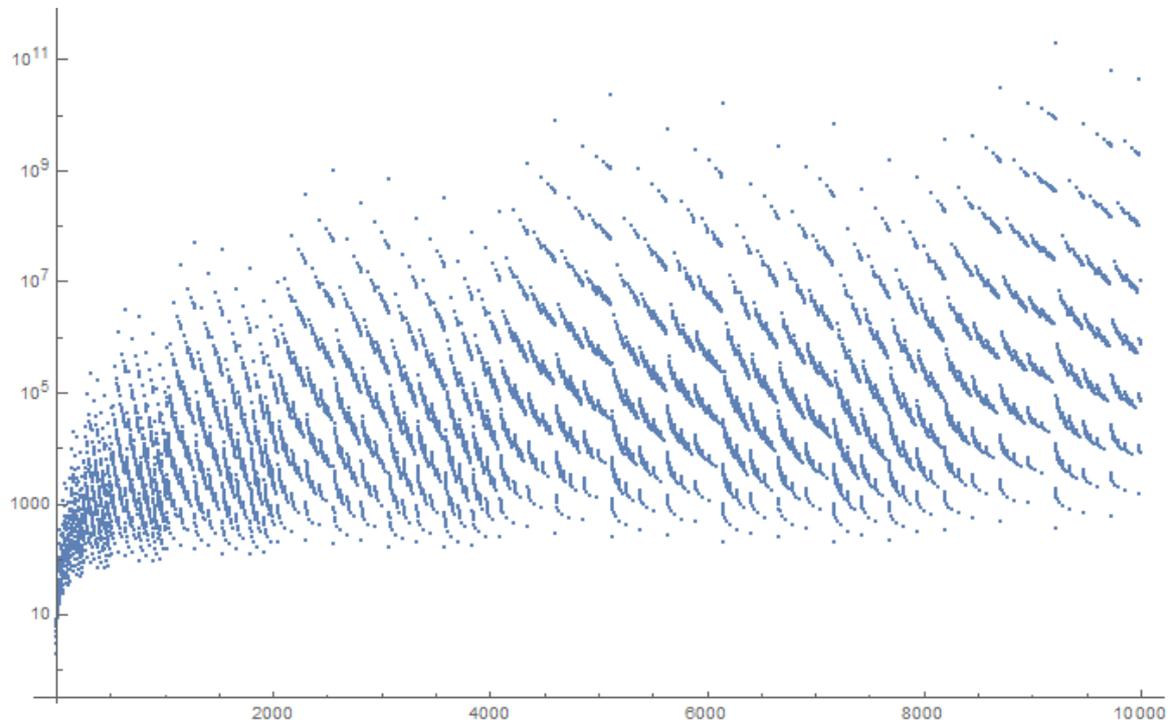


Abbildung 105: Inverse Prime-GOCRONs Typ6 (GOCRON->n) von 1 bis 10000

10.3 DIE WELT DER OCRON-WESEN UND MATHEMATISCHER SPRENGSTOFF

Hinweis: Um dieses Kapitel richtig zu verstehen, sollte der Leser sich wenigstens ein bisschen mit den Typ4-OCRONS (siehe Kapitel 10.2.2) vertraut machen.

Wir wollen jetzt einen Ausflug in die Welt der OCRON-Wesen machen. Hierbei handelt es sich um ein Gedankenexperiment, da wir nicht sicher wissen, ob diese Welt zusammen mit ihren Bewohnern, die wir 'OCRONianer' nennen wollen, tatsächlich existiert. Gedankenexperimente sind in der Wissenschaft ein erfolgreiches Mittel, um sich Zusammenhänge klarer zu machen. Es ist bekannt, dass Einstein die entscheidende Idee für seine Allgemeine Relativitätstheorie hatte, indem er sich mit Gedankenexperimenten beschäftigte. Eines dieser Gedankenexperimente war, dass er sich vorstellte, sich in einem riesigen fallenden Fahrstuhl zu befinden, der groß genug ist, dass darin ein ganzes Labor mit allen möglichen Messgeräten Platz hätte, und der keinen Kontakt nach außen hätte. Er verglich diese Situation mit einem abgeschlossenen Raum (ebenfalls mit allen möglichen Messgeräten und ohne Kontakt zur Außenwelt), der (heutzutage im Raumfahrtzeitalter kann man sich das leichter vorstellen) sich mit gleichförmiger Geschwindigkeit fernab von einem Planeten oder anderen Himmelskörpern durchs Weltall bewegt. Ein Physiker, der sich in dem einen oder dem anderen Raum befindet kann alle Messungen durchführen, die er möchte. Alle Arten von Messgeräten stehen in den beiden Räumen zur Verfügung. Er kann jedoch durch Messungen nur innerhalb des jeweiligen Raumes nicht feststellen, ob er sich in einem fallenden Fahrstuhl (man könnte auch sagen, im Einfluss eines Gravitationsfeldes), oder in einem Raumschiff befindet, dass sich mit konstanter Geschwindigkeit fernab im Weltall bewegt.

Ein ähnliches Gedankenexperiment wäre, den einen abgeschlossenen Raum, samt der Messinstrumente einfach auf der Oberfläche der Erde stehen zu lassen, den anderen Raum auf die Spitze einer Rakete zu setzen, die ihre Triebwerke eingeschaltet hat, und die sich mit einer konstanten Beschleunigung durchs Weltall bewegt. Wieder kann ein Physiker durch Messungen innerhalb des Raumes nicht unterscheiden, in welcher Situation er sich befindet.

Der logische Schluss war, dass die beiden jeweiligen Situationen nicht nur identisch erscheinen, sondern sogar identisch sind. Mit dem nötigen mathematischen Handwerkszeug kann man zwischen den jeweiligen betrachteten Situationen eine Brücke bauen. Im Fall der Allgemeinen Relativitätstheorie war es die Idee eines 'gekrümmten Raumes' der schließlich die Äquivalenz der beiden Situationen herstellte.

Unser Gedankenexperiment führt uns in eine andere Welt, und zwar in eine Welt, die so völlig anders ist, dass wir uns sie kaum vorstellen können. Die Kosmologen sprechen ja auch oft von anderen Welten. Es gibt den Begriff des Multiversums, das viele oder auch unendlich viele Universen einer bestimmten Art beinhaltet. Das Universum, in das wir uns begeben wollen, stammt aus einem Multiversum, das einen Namen hat: Max Tegmark⁵⁸ nennt es in seinem Buch 'Unser mathematisches Universum' das sogenannte 'Ebene IV' -Multiversum. Tegmark spricht von verschiedenen Paralleluniversen, die eine vierstufige Hierarchie bilden, wobei jedes Multiversum ein einzelnes Element unter vielen auf der darüber liegenden Ebene ist. Nach seiner Theorie entstanden die Ebene I und II Universen physikalisch nach dem

⁵⁸ Max Tegmark: Unser mathematisches Universum, Ullstein Buchverlage GmbH, Berlin

Urknall bei der sogenannten inflationären Phase. In Ebene I hat jedes Universum dieselben Naturgesetze und Naturkonstanten, dieselbe Mathematik, jedoch unterschiedliche Anfangsbedingungen. In Ebene II hat jedes Universum die gleichen Naturgesetze und gleiche Mathematik, jedoch unterschiedliche Naturkonstanten und Teilchen. Dieser Typ von Multiversum kann auch höhere räumliche Dimensionalität haben. Das Ebene III Multiversum entspricht dem Ebene II Multiversum, besteht jedoch aus unendlich vielen einzelnen Universen, die sich ständig neu bilden, und zwar immer dann, wenn irgendjemand irgendetwas (oder irgendjemand anderen) wahrnimmt. Es trägt der quantenmechanischen Natur unserer Welt Rechnung. Mit „Wahrnehmen“ ist die allgemeinste Form der Wahrnehmung gemeint, dazu zählt z.B. das 'Betrachten' oder 'Anschauen', ein Physiker, der eine physikalische Größe misst, aber auch ganz abstrakt eine Wechselwirkung eines komplexen Systems mit einem anderen.

Auf atomarer und subatomarer Ebene werden alle physikalischen Vorgänge durch die Quantenmechanik und durch eine, dem Vorgang beschreibende Wellenfunktion berechnet. Sie hat nur den Nachteil, dass alle potentiell messbaren Größen als eine unendliche Überlagerung aller möglichen diskreten Zustände existieren. Zumindest solange keiner 'hinschaut', d. h. solange keine Messung gemacht wird, denn im Falle einer Messung muss sich das quantenmechanische System für einen dieser unendlich vielen möglichen Werte entscheiden. Die Physiker nennen dies den Zusammenbruch der Wellenfunktion nach der 'Kopenhagener Deutung' und sind bis heute nicht glücklich darüber. Eine denkbare Alternative, die diesen Zusammenbruch der Wellenfunktion vermeidet, ist die Viele-Welten-Hypothese der Quantenmechanik, die besagt, dass sich unser gesamtes Universum aufspaltet in mehrere Universen, je nachdem welcher Vorgang betrachtet wird. Dies führt zum Ebene-III Multiversum. Philosophisch betrachtet kann man sagen, dass sich in einem solchen Multiversum alles ereignet (in irgendeinem der unendlich vielen Ebene III Universen), was sich jemals ereignen kann.

Im Ebene VI-Multiversum fällt auch die Beschränkung der einheitlichen Gleichungen der Physik weg. Jedes Universum enthält seinen eigenen Satz von mathematischen Strukturen. Viele dieser Typ VI Universen werden langweilig sein, viele werden jedoch komplex und mächtig genug sein, um eigene Welten innerhalb dieses Ebene-VI Universums zu erschaffen.

Warum holen wir so weit aus? Um zu zeigen, dass die Welt, in die wir uns begeben wollen so unvorstellbar anders ist als unsere Welt, da sie eine Ebene VI Welt (nach Max Tegmark) ist, in der eine völlig andere Mathematik wie in unserer Welt herrscht.

Jetzt kommen wir endlich zu unseren OCRON-Wesen. Nehmen wir an, die Mathematik in dieser Welt ist vielseitig und mächtig genug, um eine Welt mit einer ähnlichen Komplexität und Vielfalt wie unsere hervorzubringen. Dort in dieser Welt gibt es auch Leben: Die Bewohner sind die oben erwähnten OCRONianer. Sie leben dort in Gemeinschaften, sind intelligent, und sie betreiben aus Neugier auch Wissenschaften, unter anderem auch Mathematik. Sie können auch Maschinen, Computer und andere Geräte bauen, für die sie als Handwerkszeug Mathematik benötigen. Ihre eigene Mathematik unterscheidet sich grundlegend von unserer Mathematik. Die OCRONianer können nämlich nicht Addieren, sie können nur Multiplizieren und Potenzieren (und auf irgendeine geheimnisvolle Weise auch Logarithmieren zur Basis 2). Sie rechnen auch nicht mit Zahlen, denen ein eindeutiger Wert zugeordnet werden

kann, sondern sie rechnen nur mit OCRONs: den vier verschiedenen formalen Symbolen $*$, $'2'$, $'^'$, und $'P'$. Sie können auch nicht sagen, wie groß eine Zahl ist, denn die Begriffe 'Größe' oder Wert einer Zahl (in unserem Sinn) existiert in ihrer Welt nicht.

Der Begriff der Addition ist ihnen fremd, ja nicht nur fremd, denn er existiert in ihrer Welt einfach nicht. Sie können von kleinster Kindheit an selbst riesige Zahlen (in ihrer Welt natürlich OCRONs) mühelos multiplizieren. Auch Primfaktoren sind für sie kein Problem: Sie sehen einer Zahl in einem Bruchteil einer Sekunde an, ob sie eine Primzahl ist, oder aus welchen Primfaktoren sie besteht. Auch ihre Computer arbeiten ohne Addition, denn Zahlen werden bei ihnen nicht als Summendarstellung gespeichert, sondern immer nur als OCRON-Darstellungen (was letztlich auf ein Produkt von Primfaktoren hinausläuft). Zum Programmieren und den unvermeidlichen Berechnungen, die beim Konstruieren von Maschinen auftreten müssen sie jedoch auch Zahlen vergleichen können, und feststellen können wenn zwei Zahlen gleich „groß“ sind und sie müssen evtl. auch Operationen der Art ausführen, die wir in unserer Welt „Addition“ nennen (was in ihrer Welt zwar geht, aber nur sehr umständlich). Obwohl bei ihnen auch der Begriff des „Wertes“ einer Zahl (eines OCRONs) nicht existiert, besitzen sie jedoch eine Fähigkeit, ohne einen Begriff von „Größe“ oder „Wert“ zu haben dennoch festzustellen, welche von zwei Zahlen größer ist, oder ob sie gleich groß sind (MATHOrakel-Befragung, s. unten). Sie können das MATHOrakel auch nach einer zweiten Operation, die sie in ihren Berechnungen verwenden, befragen: Der ORAKELLog-Operation (wir in unserer Welt nennen dies den Logarithmus zu Basis 2).

„Additionen“ (diesen Begriff kennen nur wir) müssen die OCRONianer auf Grund ihrer physikalischen Gleichungen auch durchführen, aber sie wissen natürlich nicht, dass wir diesen Vorgang in unserer Welt „Addition“ nennen und dass wir eine viel einfachere Methode dafür haben. Sie verwenden dafür eine ziemlich komplizierte Methode: um zwei OCRONs o_1 und o_2 zu „addieren“ schreiben sie nämlich (das Ergebnis wollen wir o_3 nennen):

$$o_3 = \text{ORAKELLog}[2o_1 \wedge 2o_2 \wedge *].$$

In unserer Mathematik-Sprache lautet dies so: $o_3 = \text{Log}_2(2^{o_1} \cdot 2^{o_2})$.

Das Symbol 'ORAKELLog' nennen die OCRONianer den Orakellogarithmus. 'ORAKELLog' liefert entweder ein OCRON oder gar nichts zurück. Zusammen mit der Funktion 'ORAKELWert' (die nur einen der drei Werte ‚kleiner‘ ‚größer‘ oder ‚gleich‘ liefern kann, handelt es sich um die zwei mathematischen Operationen für die die OCRONianer das MATHOrakel befragen können.)

Die Methoden ‚ORAKELWert‘ und ‚ORAKELLog‘ versteht eigentlich kein OCRONianer so richtig, aber allen ist diese Methode intuitiv geläufig. Sie können innerhalb eines Sekundenbruchteiles das 'MATHOrakel' befragen, das ihnen immer die richtige Antwort auf diese Fragen geben wird. Jeder OCRONianer hat von überall aus zu jedem Zeitpunkt auf geheimnisvolle Art und Weise Zugriff auf das MATHOrakel. Auch die OCRONianer-Computer besitzen diesen Zugriff. Nun stellen die Mathematiker unter den OCRONianern fest, dass es durchaus verschiedene OCRONs gibt, die bei der MATHOrakel-Befragung mit ‚ORAKELWert‘ das gleiche Ergebnis liefern. Durch Herumprobieren und ‚ORAKELWert-Befragungen‘ finden Sie alle möglichen OCRONs heraus, die den gleichen Wert ergeben. Die schlauesten Mathematiker unter den OCRONianern haben sich deshalb viele Jahre mit dem

Problem befasst, ob es statt dem zufälligen 'Herumprobieren' auf der Suche nach 'gleichwertigen' OCRONs ein methodisches, auf einem Algorithmus basierendes Vorgehen gibt, mit dem man 'gleichwertige' OCRONs ineinander umformen könnte (und dadurch z.B. auch vereinfachen könnte). Das hätte die Arbeit der OCRONianer-Ingenieure natürlich wesentlich erleichtert, denn sie hätten dann statt zufälligem Herumprobieren eine schnelle Methode für das Umformen von OCRONs gefunden, außerdem müssten sie dann das MATHOrakel (zumindest für ihre komplizierte ‚Addition‘) nicht mehr mit ‚ORAKELWert‘ befragen. Schlaue OCRONianer-Mathematiker fanden außerdem heraus, dass dann auch die Befragung nach ‚ORAKELLog‘ hinfällig wäre, vorausgesetzt, sie besäßen einen Algorithmus, der durch Umformen alle ‚gleichwertigen‘ OCRONs erzeugen könnte.

Für das Finden eines solchen Algorithmus, wurde in der OCRONianer-Welt ein Preis ausgesetzt.

Hier ein Beispiel: Die Multiplikation von $8 * 4$ sieht in unserer Welt so aus: $8 * 4 = 32$.

In der OCRONianer-Welt: $22P^{\wedge}$ mal 22^{\wedge} ergibt $22P^{\wedge}22^{\wedge} *$ (OCRONianer multiplizieren, in dem sie OCRONs einfach verketteten und ein ‚*‘ hinten anfügten).

Die OCRONianer sind nun in der Lage durch zufälliges Herumprobieren und ORAKELWert-Befragungen festzustellen, dass z. B. $22P^{\wedge}22^{\wedge} *$ den gleichen Wert hat wie $22PP^{\wedge}$.

Sie besitzen (oder besäßen...) jedoch keinen Algorithmus der aus $22P^{\wedge}22^{\wedge} *$ das gleichwertige OCRON $22PP^{\wedge}$ erzeugt.

In unserer Welt sprechen Mathematiker und Logiker von einem (typographischen) formalen System. OCRONianer müssen mühsam so einfache Sachen wie Addieren durch Hantieren und zufälligen ‚Würfeln‘ mit Symbolen lösen. Ihnen ist der Schritt in die ‚höhere‘ logische Welt, in der es eine Addition gibt, verwehrt.

Arme OCRONianer! Wie geht die Geschichte weiter? Wird es den schlauesten OCRONianern gelingen, einen solchen Algorithmus zu finden?

Wir wollen uns zwei mögliche (erfundene) Szenarien einer Geschichte ausdenken.

Szenario 1: Ein OCRONianer-Mathematiker findet einen Algorithmus zum Umformen gleichwertiger OCRONs. Er wird gefeiert und bekommt dafür die OCRONianer-Fields-Medaille. Das MATHOrakel muss für die ‚Additionen‘ nicht mehr befragt werden. Die Lösung ist zwar kompliziert (zur Berechnung von Umformungen müssen Lösungen mit Hilfe von relativ komplizierten rekursiven Regeln und Regeln, die wiederum neue Regeln erfinden, gefunden werden), aber es ist immerhin eine Methode, die MATHOrakel-Befragung zu umgehen. Außerdem ist die Methode mit der ‚ORAKELWert‘-Befragung immer auch mit langwierigen zufälligem ‚Ausprobieren‘ verbunden, so dass die komplizierte Methode der rekursiven Regeln für lange OCRONs unschlagbar besser als die Zufallsmethode funktioniert.

Szenario 2: Ein schlauer OCRONianer Namens ‚Gocrödel‘ findet einen Beweis, dass das Axiomensystem der OCRONianischen Mathematik einfach zu ‚schwach‘ und zu wenig ‚mächtig‘ ist, um das Umformungsproblem zu lösen. Er behauptet, dass die Aussage:

" $22P^{\wedge}22^{\wedge} *$ ist äquivalent mit $22PP^{\wedge}$ " zwar tatsächlich wahr ist, jedoch nicht mit der OCRONianischen Mathematik bewiesen werden kann. Das impliziert natürlich,

dass kein solcher Algorithmus gefunden werden kann. Dafür bekommt Gocrödel zwar auch den OCRONianer-Fields-Medaille, aber so richtig freuen können sich die OCRONianer darüber nicht.

Der aufmerksame Leser wird vermutlich erkennen, welcher ‚Sprengstoff‘ sich im Szenario 1 verbirgt:

Wäre das Szenario 1 wahr, dann könnten wir von den (oben als ‚arm‘) bezeichneten OCRONianern lernen: Wir könnten ihren ‚Umform-Algorithmus‘ übernehmen und hätten im Handumdrehen eine Methode zum schnellen Faktorisieren von Zahlen: Wir müssten die zu faktorisierende Zahl einfach in eine Summe von zwei Zahlen zerlegen, deren Primfaktorzerlegung wir kennen (genauer: deren OCRON-Darstellungen wir kennen). Dann setzen wir an den Anfang der jeweiligen OCRONs ein ‚2‘-Symbol, an das Ende der beiden OCRONs ein ‚^‘-Symbol, verketteten die beiden neuen OCRONs (OCRON-Multiplikation) und hängen ein ‚*‘-Symbol hinten an. Ist der zweite Summand ‚1‘ dann ist die Sache noch einfacher: das zweite OCRON ist dann einfach das Symbol ‚2‘. Schließlich formen wir das verkettete OCRON mit Hilfe des sagenhaften Algorithmus in ein gleichwertiges OCRON um, so dass am Ende des OCRONs ein ‚^‘-Symbol steht, bilden dann problemlos den Logarithmus zur Basis 2 (durch ‚Weglassen‘ der führenden ‚2‘ und dem letzten ‚^‘, und, ‚voila‘: Wir haben eine Produktdarstellung (die ist ja in jedem OCRON impliziert vorhanden) der zu faktorisierenden Zahl! Wenn das resultierende OCRON am Ende kein ‚*‘ Symbol oder ‚^‘-Symbol besitzt, sondern ein ‚P‘ Symbol, dann ist unsere zu faktorisierende Zahl eine Primzahl. Wir hätten das Faktorisierungsproblem auf rein typographischer Ebene durch Anwenden von typographischen Umformungsregeln gelöst.

Hier ein paar Beispiele:

Wir untersuchen die Zahl **37**. Zerlegung: **37 = 36 + 1**:

In OCRON-Schreibweise:

$36 = 22^2 P 2^2 * \rightarrow$ (Klammern sind nur der Übersichtlichkeit halber eingefügt)

$(222^2 2 P 2^2 * ^)(2) * \rightarrow$ (MATHOrakelbefragung) $222^2 2 P * P^$

(Logarithmus: 2 und ^ weglassen) $\rightarrow 22^2 2 P * P$, Primzahl!

Somit haben wir mit Hilfe von OCRONs nachgewiesen, dass 37 eine Primzahl ist.

Wir untersuchen die Zahl **37**. Zerlegung: **37 = 16 + 21**:

In OCRON-Schreibweise:

$16 = 222^{\wedge\wedge}, 21 = 2P22^{\wedge} P * \rightarrow$

$(2222^{\wedge\wedge})(2P22^{\wedge} P * ^) * \rightarrow$ (MATHOrakelbefragung) $222^{\wedge} 2P * P^$

(Logarithmus: 2 und ^ weglassen) $\rightarrow 22^{\wedge} 2P * P$, Primzahl!

Wir untersuchen die Zahl **143**. Zerlegung: **143 = 71 + 72**:

In OCRON-Schreibweise:

$71 = 22^{\wedge} 2 P P * P, 72 = 22 P^{\wedge} 2 P 2^{\wedge} * \rightarrow$

$(222^{\wedge} 2 P P * P^)(2 22 P^{\wedge} 2 P 2^{\wedge} * ^) * \rightarrow$ (Orakelbefragung) $22 P P P 22 P * P * ^$

(Logarithmus: 2 und ^ weglassen) $\rightarrow 2 P P P 22 P * P * = (2 P P P)(22 P * P) *$

Ergibt die Faktoren **2 P P P (= 11) und 22 P * P (= 13)**

Somit haben wir 143 mit Hilfe von OCRONs in die Faktoren 11 und 13 zerlegt.

Hinweis: Die Umformung ohne MATHOrakel-Befragung gestaltet sich auch in unserer Welt schwierig, dafür müssen wir den Wert von $222^{2PP} * P^2 22P^{2P2} * ^ * ^ *$ explizit ausrechnen:

11150372599265311570767859136324180752990208, um anschließend diesen Wert wieder in das (umgeformte) OCRON zu konvertieren.

Diese Beispiele zeigen, dass wir mit Hilfe von Typ4-OCRONs Zahlen durch rein typographische Manipulation von Symbolen faktorisieren können, vorausgesetzt wir haben Zugriff auf das MATHOrakel!

Bleibt nur noch das ‚kleine‘ Problem, wie wir die MATHOrakel -Befragung umgehen können und den sagenhaften Algorithmus finden!

Bevor man sich auf die Suche nach diesem Algorithmus macht, müsste erstmal geklärt werden, ob das ganze Problem nicht in die Kategorie der ‚unbeweisbaren‘ Aussagen gehört. In diesem Fall wäre es gänzlich unmöglich einen solchen Algorithmus zu finden (zumindest innerhalb der ‚OCRONianer-Mathematik‘).

(Nachtrag:)

Hierfür hat der Autor einen wahrhaft wunderbaren Algorithmus gefunden. Der Rand ist jedoch zu schmal, ihn zu fassen...

-Ende des Gedankenexperimentes-

11 PRIMZAHLEN UND DIE „MATRIX“-SOFTWARE: EINE GESETZMÄßIGKEIT?

11.1 REGELN FÜR DIFFERENZEN N-TER ORDNUNG

In diesem Kapitel werden Zahlenfolgen mit Hilfe der Matrix-Software⁵⁹ untersucht. Deshalb sei hier kurz diese Software und ihre Funktionsweise vorgestellt.

Matrix ist eine Software, die es erlaubt, aus vorgegebenen Daten, die irgendwie kausal geordnet sind, und die eine Folge von Zuständen eines Systems repräsentieren, ein Regelwerk (d.h. eine Menge von Regeln) zu erstellen, das die einzelnen Übergangswahrscheinlichkeiten des Systems von einem Zustand in den anderen beschreibt. Mit Hilfe dieser Übergangswahrscheinlichkeiten kann die Matrix eine beliebig lange Folge von Zuständen dieses Systems erzeugen. Im Prinzip handelt es sich um eine verallgemeinerte Markow-Kette mit Übergangswahrscheinlichkeiten. Bei der Berechnung der Übergangswahrscheinlichkeiten kann jedoch die ganze 'Historie' eines Prozesses mit einfließen. Außerdem kann die Matrix auch 'Beendigungswahrscheinlichkeiten' (d.h. die Wahrscheinlichkeit dass eine Folge von Zuständen endet) liefern, sowie das Gegenteil, d. h. die Erschaffung eines Zustandes ('ex nihilo') beim Start einer neuen Zustandsfolge. In ihrer einfachsten Form kann die Matrix auch einfach dazu benutzt werden, hochdimensional strukturierte Daten abzuspeichern (daher der Name 'Matrix', da es sich im Grunde genommen um nichts anderes als um eine hoch dimensionale Pointer-Matrix (mit variabler Dimensionslänge) handelt. Die Matrix kann auf fast alle Systeme angewendet werden. Die Voraussetzungen für die Anwendbarkeit sind sehr allgemein: - Der Zustand des Systems an einem bestimmten Startpunkt muss durch eine Menge von ganzen (auch negativen) Zahlen beschreibbar sein. - Es sollte genug Material über das Verhalten des Systems vorhanden sein. Ist die Matrix einmal mit Daten über ein System gefüttert worden, so kann man aus der Matrix wie aus einem unendlichen ‚Stream‘ lesen. Bei geeigneter Wahl der Parameter wird dieser ‚Stream‘ immer auch neuartige Übergänge liefern, d.h. es handelt sich beim Lesen aus der Matrix durchaus um einen kreativen Prozess. In der Matrix selbst werden keine Folgen von Zuständen gespeichert sondern ausschließlich Regeln, die sehr viel kürzer sind. Beim Lesen aus der Matrix kann die Intelligenz der Matrix eingestellt werden. Bei hoher Intelligenz wird das Ergebnis nahe am Original sein, mit dem die Matrix gefüttert wurde; bei niedriger Intelligenz fließen immer mehr Zufallselemente mit ein. Das Ergebnis des Outputs beim Lesen wirkt sehr viel intelligenter als es die Einfachheit der Regeln vermuten ließe. Irgendwie scheint die Matrix das Wissen über das Verhalten eines Systems nicht nur lokal in den Regeln zu speichern sondern holistisch in der Gesamtheit aller Regeln. Wenn man z. B. einen Teil der Regeln entfernen wird, so wird sich das Ergebnis beim Lesen nicht sprunghaft ändern.

Machen wir einen ersten Test: Wir stellen die maximal zu untersuchende Regellänge auf 20 und füttern die Matrix mit den Differenz-Sequenzen 1. Ordnung der Primzahlen und schauen uns die Häufigkeiten der berechneten Regellängen an:

Für die ersten 1000 Primzahlen (2-7919):

⁵⁹ <http://www.kmatrix.eu>

Die Matrix findet 2581 Regeln, davon 1093 eindeutige Regeln, die Maximas der Häufigkeiten liegen bei Regellänge 5, die längste Regellänge beträgt 10. Die Größe der Matrix-Datendatei beträgt 107 KByte.

Für die ersten 10000 Primzahlen (2 – 104729)

Die Matrix findet 25092 Regeln, davon 10590 eindeutige Regeln, die Maxima der Häufigkeiten liegen zwischen Regellänge 5 und 6, die längste Regellänge beträgt 15. Die Größe der Matrix-Datendatei beträgt 1,00 MByte.

Für die ersten 100000 Primzahlen (2 – 1299709)

Die Matrix findet 245731 Regeln, davon 104032 eindeutige Regeln, das Maximum der Häufigkeiten liegt bei Regellänge 6, die längste Regellänge beträgt 15.

Die Größe der Matrix-Datendatei beträgt 9,78 MByte.

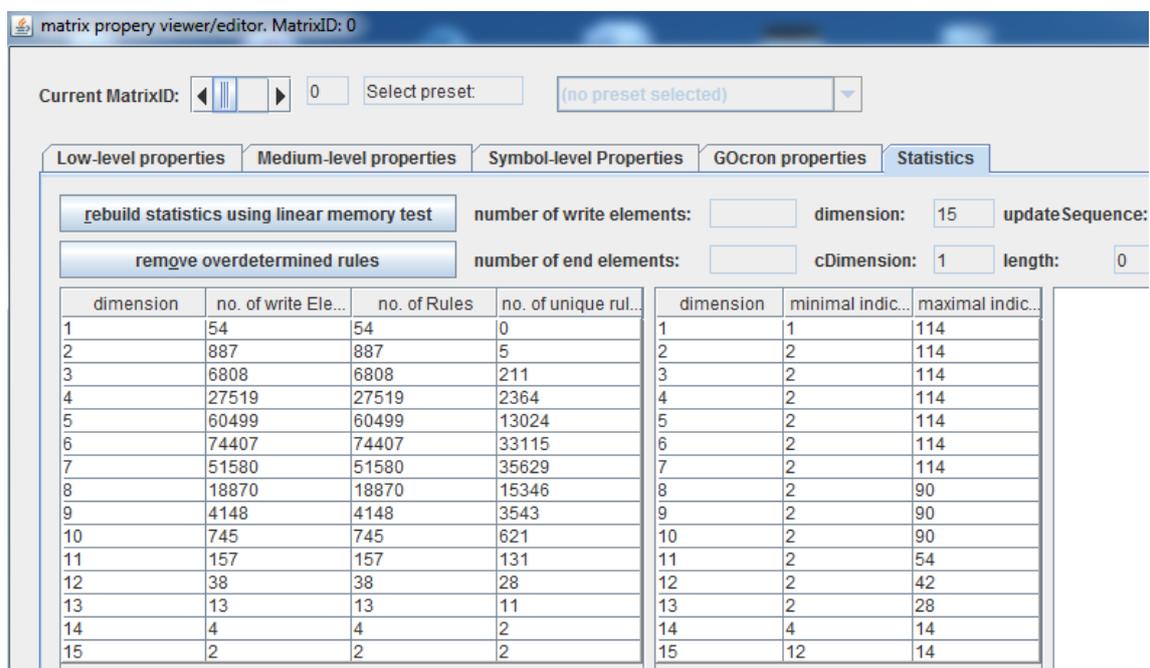


Abbildung 106: Matrix: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 100000 Primzahlen

Regeln für Differenzen n-ter Ordnung

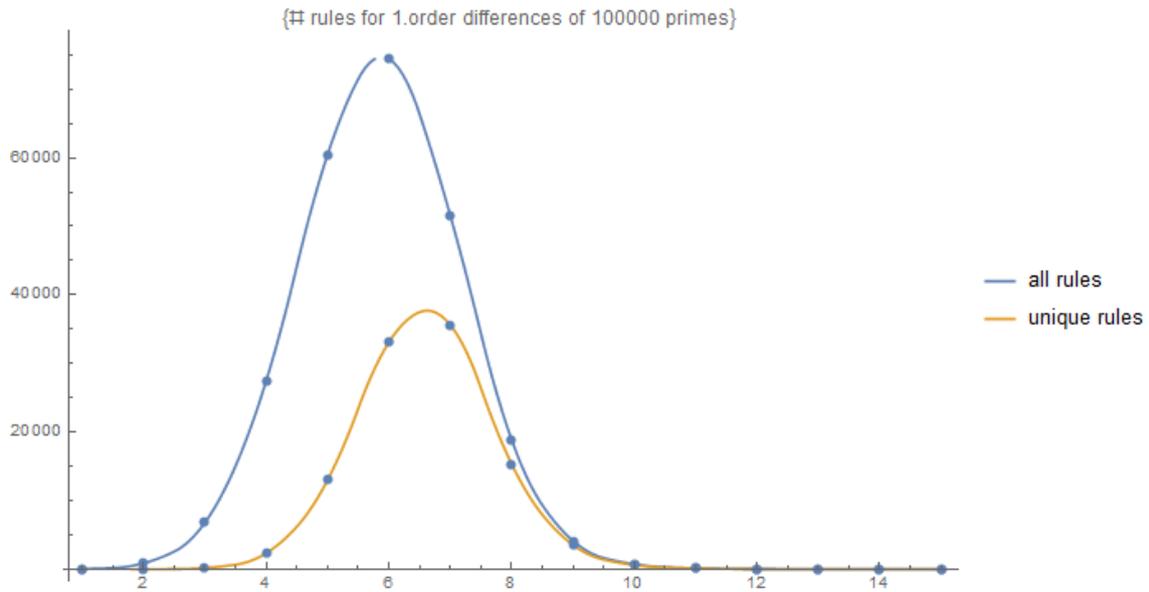


Abbildung 107: Diagramm: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 100000 Primzahlen

Mathematica:

```
data=Import["/Primzahlen/data_and_Docs/StatisticsPrimesRulesFrom100000PrimesPr  
eprocl.txt",{"Data",All,{1,2,3,4}}];  
ListLinePlot[{Transpose[data][[2]],Transpose[data][[4]]},AxesLabel-  
>Automatic,PlotRange->All,Mesh->Full,InterpolationOrder->2,PlotLegends->{"all  
rules","unique rules"},ImageSize->Large]
```

Für die ersten 1000000 Primzahlen (2 – 15485863)

Die Matrix findet 2422245 Regeln, davon 1030290 eindeutige Regeln, die Maximas der Häufigkeiten liegen zwischen Regellänge 6 und 7, die längste Regellänge beträgt 15. Die Größe der Matrix-Datendatei beträgt 96,2 MByte.

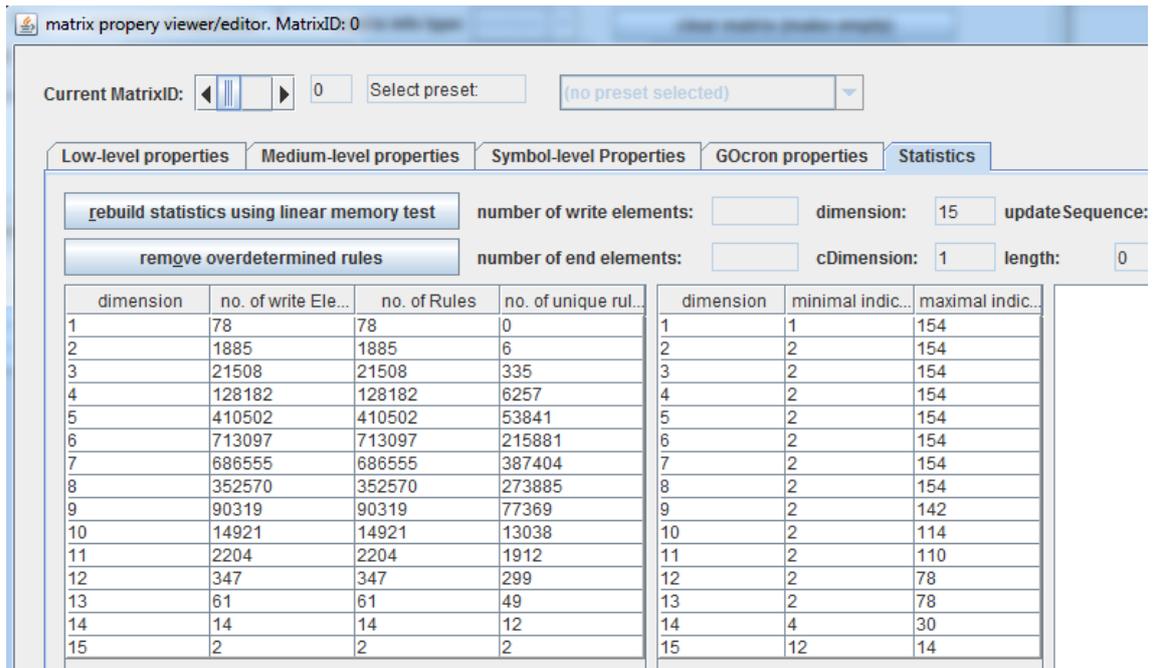


Abbildung 108: Matrix: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 1000000 Primzahlen

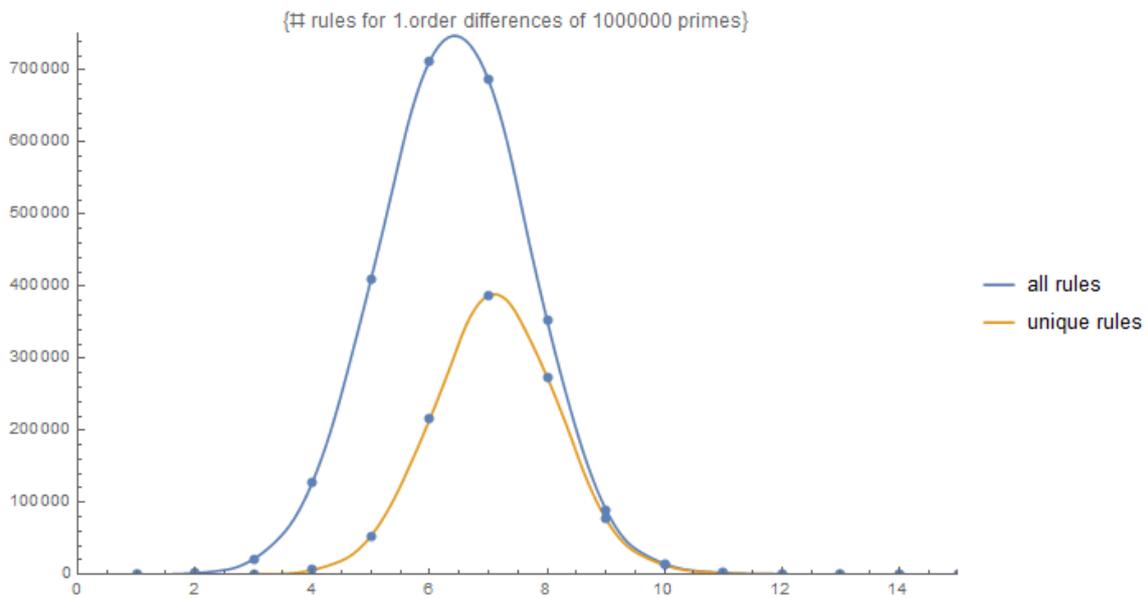


Abbildung 109: Diagramm: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 1000000 Primzahlen

Es scheint, als ob eine Sequenz von 15 aufeinander folgenden Primzahlen genügt um die 16-te darauf folgende Primzahl mit Hilfe des Regelnetzwerkes zu berechnen.

Allerdings wächst mit zunehmendem Zahlenbereich auch die Anzahl der Regeln ungefähr linear an:

Regeln für Differenzen n-ter Ordnung

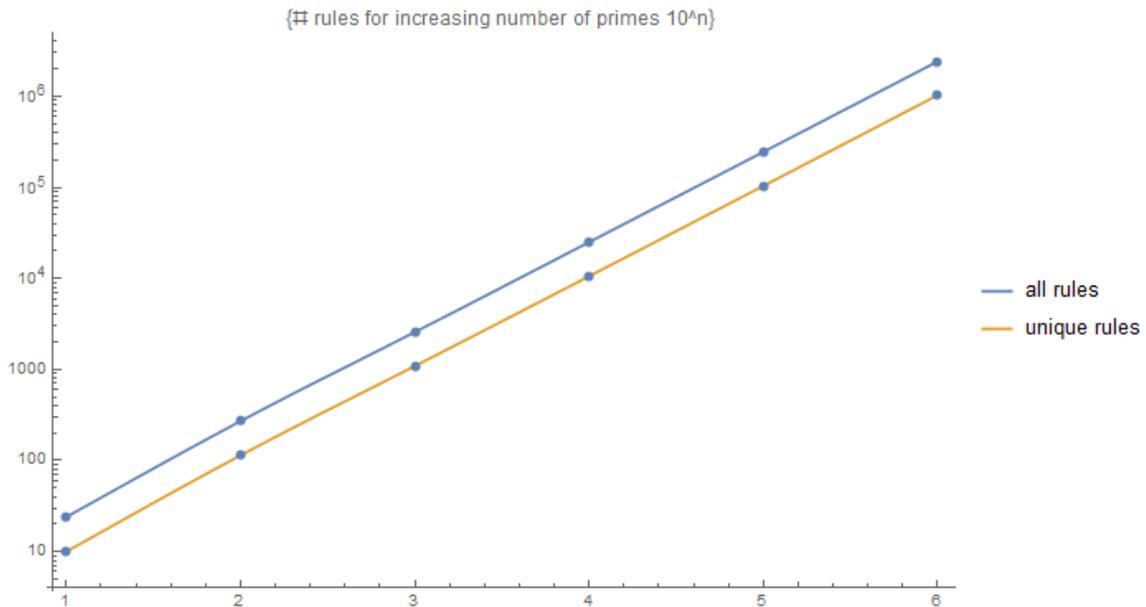


Abbildung 110: Anzahl der Regeln erstellt aus der Differenzen-Primzahlfolge 1. Ordnung in Abhängigkeit des Zahlenbereichs 10^n

Mathematica:

```
data={{1,24,10},{2,271,114},{3,2581,1093},{4,25092,10590},
{5,245731,104032},{6,2422245,1030290}};
ListLogPlot[Transpose[data][[2]],Transpose[data][[3]],AxesLabel->Automatic,PlotRange->All,Mesh->Full,Joined->True,InterpolationOrder->2,PlotLegends->{"all rules","unique rules"},PlotLabel->{"# rules for increasing number of primes 10^n"},ImageSize->Large]
```

Bei Differenzen höherer Ordnung beobachten wir folgendes Verhalten bei verschiedenen Ordnungen:

Ordnung Differenz	Anzahl Primzahlen	Längste Regel	Anzahl Regeln	# eindeutige Regeln	Dateigröße (KByte)	Minimaler Wert	Maximaler Wert
1	10	6	24	10			
	100	10	271	114			
	1000	10	2581	1093	107	1	34
	10000	15	25092	10590	1003	1	72
	100000	15	245731	104032	9777	1	114
2	1000000	15	2422245	1030290	96206	1	154
	100	9	254	111			
	1000	9	2506	1074			
	10000	14	24431	10468			
	100000	14	241138	103273			
4	1000000	14	2386607	1024963	95478	-148	144
	100	7	233	101			
	1000	7	2327	1031			
	10000	12	23092	10183			
	100000	12	229102	101283	9064	-332	304
8	1000000	12	2282543	1009733	90192	-448	460
	100	3	189	91			
	1000	4	2247	991			
	10000	8	21252	10002			
	100000	8	211227	100051			
10	1000000	8	2225543	1000473	88671	-5962	5638
	100	3	182	89		-1538	1606
	1000	4	2138	989		-5000	4608

	10000	6	22301	9992		-9488	10238
	100000	6	206803	99993	8512	-14476	15640
	1000000	6	2057953	1000029	85678	-20520	22450
14	100	3	172	85			
	1000	3	1995	985			
	10000	3	20946	9985			
	100000	4	226154	99985	8960	-206992	221570
	1000000	4	2094308	999985	81947	-298794	323414
15	100	2	169	84			
	1000	3	1976	984			
	10000	3	20498	9984	847	-277842	284414
	100000	4	222665	99984	9066	-428562	415348
	1000000	4	2151843	999985	84230	-622208	613240
16	100	2	167	83			
	1000	3	1970	983			
	10000	3	20247	9983			
	100000	3	215514	99983	8843	-843910	790698
	1000000	4	2222938	999984	87189	-1235448	1147684
20	100	2	159	79			
	1000	3	1960	979		-3837740	3855400
	10000	3	19985	9979		-8272220	7732656
	100000	3	201490	99979	8655	-12428154	11690554
	1000000	3	2101677	999979	89894	-18210894	17092050

Abbildung 111: Wie verhalten sich Regellängen bei Primzahl Differenz-Sequenzen n-ter Ordnung?

matrix property viewer/editor. MatrixID: 0

Current MatrixID: 0 Select preset: (no preset selected)

Low-level properties Medium-level properties Symbol-level Properties GOcron properties **Statistics**

rebuild statistics using linear memory test number of write elements: dimension: 4 updateSequence:

remove overdetermined rules number of end elements: cDimension: 1 length: 0

dimension	no. of write Ele...	no. of Rules	no. of unique rul...	dimension	minimal indic...	maximal indic...
1	51155	51155	0	1	-206992	221570
2	99984	99984	24972	2	-206992	221570
3	75013	75013	75011	3	-128698	136810
4	2	2	2	4	23856	32462

Abbildung 112: Matrix: Häufigkeit der Regellängen bei 14. Ordnung Differenzsequenzen der ersten 100000 Primzahlen

Interessant ist auch die Abhängigkeit der maximalen Regellänge von der Ordnung der berechneten Primzahl-Differenz-Sequenzen. Die Regellänge kann nicht kleiner sein als 2 (der ‚ex nihilo‘-Regel und der einfachsten Regel, die für jeden Wert einen Nachfolger berechnet). Für hohe Ordnungen der Differenzen-Sequenzen konvergiert dieser Wert gegen 2, was der Tatsache entspricht, dass jeder Wert in der betrachteten Differenzen-Sequenz höchstens einmal vorkommt. Das ist nicht überraschend und war auch so zu erwarten.

Hier ein Diagramm, das diese Abhängigkeit im Bereich der ersten 1000000 Primzahlen beschreibt (es wurden Primzahl-Differenz-Sequenzen bis zur Ordnung 20 ausgewertet):

Regeln für Differenzen n-ter Ordnung

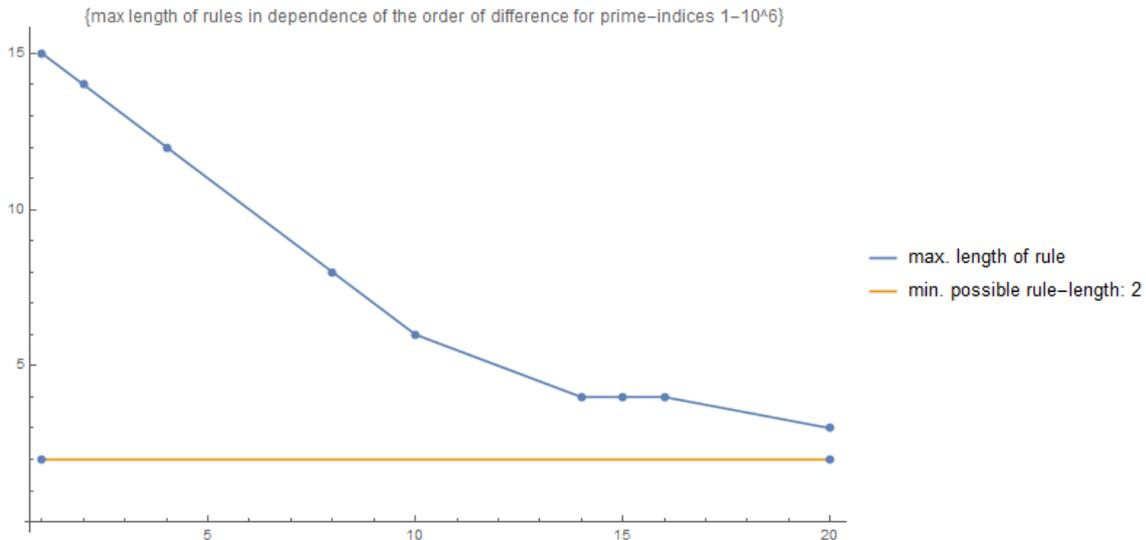


Abbildung 113: Max. Regellänge in Primzahl-Differenz-Sequenzen der Ordnung n für die ersten 10⁶ Primzahlen

Hier noch ein paar Statistiken bei Sequenzen von Primzahl-Differenzen n -ter Ordnung. (Mathematica-Programme finden sich im Anhang).

Die Differenzen n -ter Ordnung lassen sich mit Mathematica leicht berechnen.

Hier z. B. die Differenzensequenz der ersten 100000 Primzahlen für die Ordnung 1:

```
range=100000; order=1;
data=Differences[Prime[Range[range]],order];
```

Von den 99999 Werten gibt es 54 verschiedene Werte:

```
different=Union[data]
{1,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,64,66,68,70,72,74,76,78,80,82,84,86,88,90,92,94,96,98,100,106,112,114}
```

Von den 99999 Werten gibt es 49 Werte, die mindestens zweimal vorkommen:

```
doubles=With[{sData=Sort@data},DeleteDuplicates@sData[[SparseArray[Unionize@Differences@sData,Automatic,1] ["AdjacencyLists"]]]]
{2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,64,66,68,70,72,74,76,78,80,82,84,86,88,90,92,96,98,100}
```

Von den 99999 Werten gibt es 5 Werte, die ein einziges Mal vorkommen:

```
{1,94,106,112,114}
```

Bei der Differenzensequenz der ersten 100000 Primzahlen für die Ordnung 20 sieht die Sache schon komplett anders aus:

Von den 99999 Werten gibt es 98426 verschiedene Werte:

```
different=Union[data]
{-12428154,-10525630,-10259274,..., 10253734,11413498,11690554}
```

Von den 99999 Werten gibt es 1531 Werte, die mindestens zwei mal vorkommen:

```
doubles=With[{sData=Sort@data},DeleteDuplicates@sData[[SparseArray[Uni-
tize@Differences@sData,Automatic,1] ["AdjacencyLists"]]]]
{-5979490,-5554652,-5075372,...,4158040,5065004,6712100}
```

Von den 99999 Werten gibt es 96895 Werte, die ein einziges Mal vorkommen. Diese Tendenzen drücken sich in folgenden Diagrammen aus (die sortierten Differenzen n-ter Ordnung). Dort, wo es viele dicht beieinander liegende Werte gibt, ist die Kurve flach; dort wo die auftretenden Werte weit auseinander liegen, wird die Kurve steil. Dies ist typischerweise der Fall wenn die Beträge der Werte groß werden.

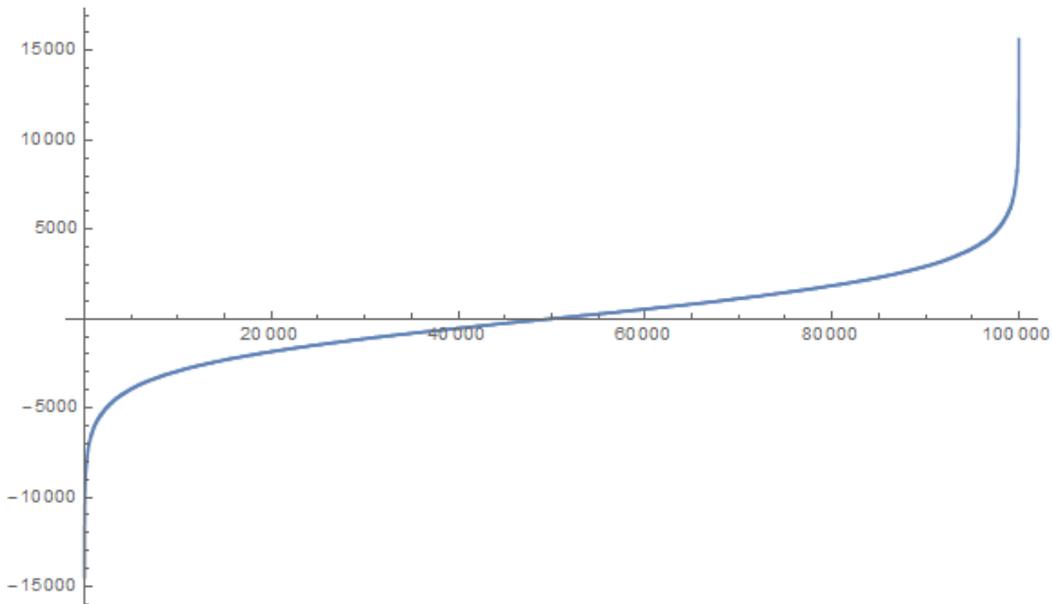


Abbildung 114: Sortierte Primzahl-Differenz-Werte der Ordnung 10 der ersten 100000 Primzahlen

```
Mathematica:
range=100000; data=Sort[Differences[Prime[Range[range]],10]];
ListLinePlot[data,AxesLabel->Automatic,PlotRange->All,InterpolationOrder->0,ImageSize->Large]
```

Regeln für Differenzen n-ter Ordnung

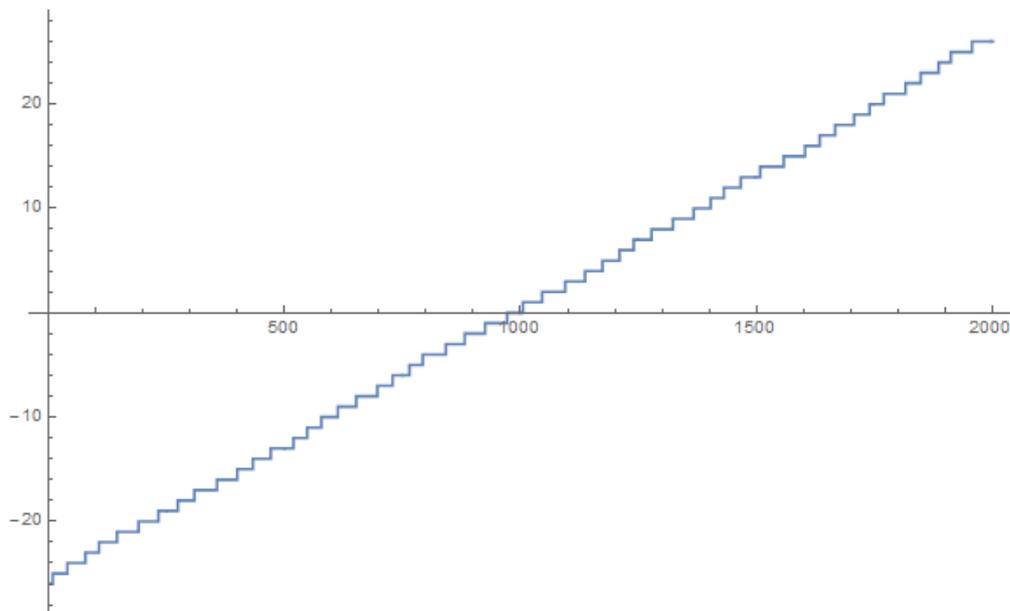


Abbildung 115: : Sortierte Primzahl-Differenz-Werte der Ordnung 10 im mittleren Bereich (2000 Werte) der ersten 100000 Primzahlen

```
Mathematica:  
range=100000;  
data=Sort[Differences[Prime[Range[range]],10]];  
ListLinePlot[data[[range/2-1000;;range/2+1000]]/2, AxesLabel-  
>Automatic, PlotRange->All, InterpolationOrder->0, ImageSize->Large]
```

Es ist zu erkennen, dass im mittleren Bereich die Werte dicht liegen, im äußeren Bereich immer dünner werden.

12 DIE ABC-VERMUTUNG

12.1 ALLGEMEINES

Die ‚abc-Vermutung‘ zählt zu den Top-Ten der ungelösten mathematischen Vermutungen. Viele Mathematiker sind der Meinung, sie sei gegenwärtig das wichtigste ungelöste Problem der Zahlentheorie. Sie ist deshalb so interessant, weil sie versucht, die beiden Welten der Addition und der Multiplikation zu verquicken.

Die einfachste Formulierung der abc-Vermutung lautet etwa so:

Sei $a + b = c$ mit $a, b, c \in \mathbb{N}$.

Seien außerdem a und b zueinander teilerfremd (auch relativ prim oder ‚koprim‘ genannt). Dafür gibt es mehrere Schreibweisen:

$a \perp b$ oder $\text{ggT}(a, b) = 1$ oder auf Englisch: $\text{gcd}(a, b) = 1$

Aufgrund der Additionsbeziehung sind auch a und c sowie b und c zueinander teilerfremd. Die abc-Vermutung sagt nun aus, dass für solche additive Tripel die multiplikative Struktur der Tripel aufgrund ihres additiven Zusammenhanges stark eingeschränkt ist:

Das Produkt aller auftretenden Primfaktoren, potenziert mit einem Exponenten, der beliebig nahe bei 1 liegt, $(\text{rad}(abc))^{1+\varepsilon}$ ist fast immer größer oder gleich als die größte Zahl des Tripels (c), mit beliebig kleinem ε .

‚Fast immer‘ bedeutet in der Mathematik: Alle, bis auf endlich viele.

Die sogenannte starke abc-Vermutung sagt nun, dass es nur endlich viele ‚Ausnahmen‘ gibt, so dass $(\text{rad}(abc))^{1+\varepsilon} < c$, mit einem beliebig kleinen Wert ε .

Diese Ausnahmen werden abc-Treffer genannt. Beispiele für solche abc-Treffer sind die Tripel:

$\{1, 8, 9\}$, $\{5, 27, 32\}$, $\{32, 49, 81\}$, usw.

Gelegentlich werden auch abc-Tripel, für die die etwas schwächere Bedingung $\text{rad}(abc) \leq c$ gilt (d.h. mit $\varepsilon = 0$), als abc-Treffer bezeichnet. In diesem Fall ist jedoch bewiesen worden, dass es unendlich viele Treffer gibt.

Je nachdem, wie stark diese Treffer von der Voraussage der abc-Vermutung abweichen, werden sie bewertet mit dem Wert

$$q = \frac{\ln c}{\ln \text{rad}(abc)} \quad (152)$$

Dieser Wert q wird in der Literatur auch als ‚Qualität‘, ‚Potenz‘ oder ‚abc-ratio‘ bezeichnet. Anschaulich stellt er ein Maß für das Anwachsen von c mit dem Primzahlinhalt $(\text{rad}(abc))$ des Tripels dar, da q die Lösung von $[\text{rad}(abc)]^q = c$ darstellt. Bisher (Stand: Feb. 2016) sind nur 237 abc-Tripel mit einer ‚Potenz‘ $q > 1.4$ entdeckt worden. Es gibt auch noch andere Bewertungen dieser abc-Treffer (siehe⁶⁰).

Ein abc-Treffer heißt ‚unbeaten‘ (unübertroffen), wenn jeder bekannte abc-Treffer mit einem größeren c jedoch eine kleinere Qualität (oder Potenz) aufweist.

Der Weltrekord (abc-Treffer mit höchster Qualität) lautet (Stand Feb. 2016):

⁶⁰ https://de.wikipedia.org/wiki/Abc-Vermutung#Weitere_Bewertungen_eines_abc-Treffers

$$\{abc\} = \{2, 6436341, 6436343\} = \{2, 109 \cdot 3^{10}, 23^5\},$$

mit $\text{rad}(abc) = 2 \cdot 23 \cdot 109 = 15042$

Wenn sich die abc-Vermutung als richtig herausstellen sollte, folgen aus ihr eine ganze Reihenfolge von wichtigen zahlentheoretischen Sätzen (so würde sich z. B. der Beweis der berühmten Fermatschen Vermutung auf ein paar Zeilen reduzieren).

Der Japaner Shinichi Mochizuki hat bereits 2012 einen Beweis der abc-Vermutung veröffentlicht. Der Status seines Beweises innerhalb der Mathematiker-Gemeinde ist immer noch unentschieden. Der 500-seitige Beweis ist selbst für Spezialisten auf diesem Gebiet sehr schwer zu verstehen.⁶¹ Die Kommentare von Mathematikern zu seinem Beweis reichen von ‚Papier aus der Zukunft‘ bis zu ‚außerirdisch‘.

Hier noch eine ‚mathematisch präzise‘ Formulierung der abc-Vermutung:

Für jedes beliebig kleine $\varepsilon > 0$ existiert eine Konstante C_ε so dass für jedes beliebige Tripel aus zueinander teilerfremden natürlichen Zahlen a, b, c , die der Gleichung $a + b = c$ genügen, folgende Ungleichung gilt (abc-Vermutung):

$$\max(a, b, c) \leq C_\varepsilon \prod_{p|abc} p^{1+\varepsilon} \quad (153)$$

Hinweis: Z.Zt. (Stand Feb. 2016) sind 33.18 Millionen abc-Treffer bekannt.

Weltweit gibt es nur etwa 5 Mathematiker, die von sich behaupten, den ganzen Beweis von Mochizuki gelesen zu haben.⁶²

Zur Veranschaulichung hier ein paar graphische Darstellungen.
(Erstellt mit folgendem Mathematica-Programm):

Mathematica:

```
rad[n_] := Times@@First/@FactorInteger[n];
isABC[a_, b_, c_] := (If[a+b!=c || GCD[a,b] != 1, Return[0]]; r=rad[a*b*c]; If[r<
c, Return[1], Return[0]]);
isC[c_] := (For[a=1, a<=Floor[c/2], a++, If[isABC[a, c-a, c] != 0, Return[1]]]; Return[0]);
tab=Select[Range[10000], isC[#]==1 &]
ListLinePlot[tab, InterpolationOrder->0, PlotStyle->Black, PlotLabel->
>"abc-conjecture: possible c-values"]
```

⁶¹ <http://www.nature.com/news/the-biggest-mystery-in-mathematics-shinichi-mochizuki-and-the-impenetrable-proof-1.18509>

⁶² https://en.wikipedia.org/wiki/Abc_conjecture

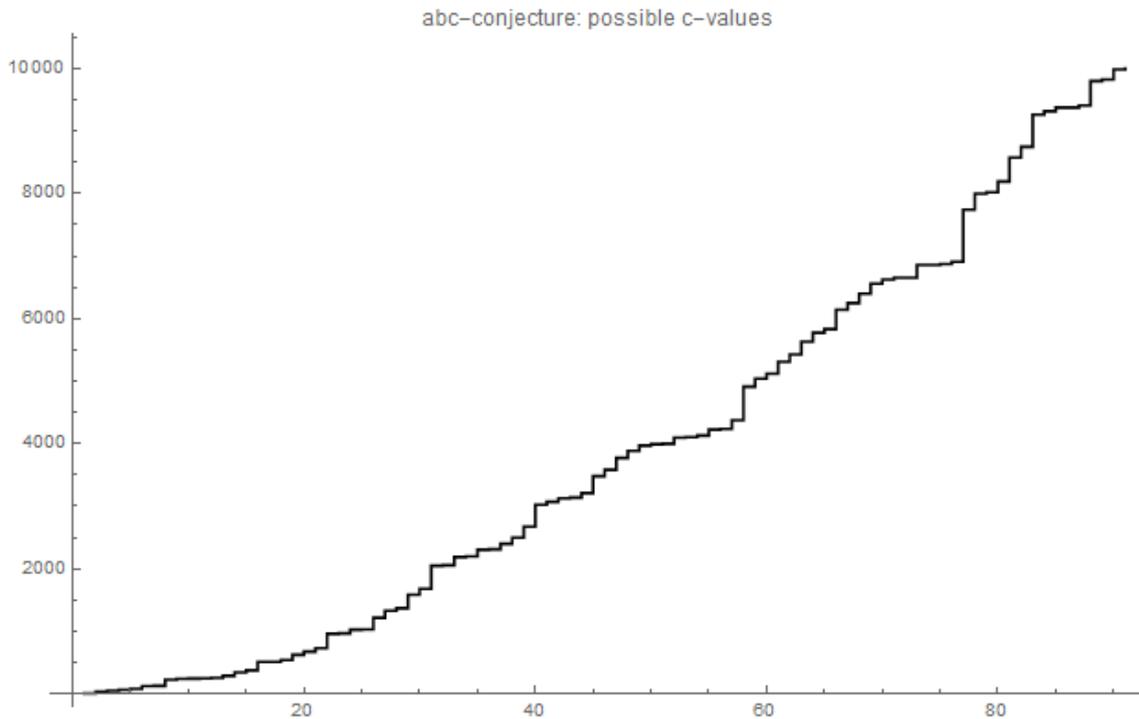


Abbildung 116: abc-Treffer: die ersten 91 möglichen c-Werte (9-10000)

abc-Treffer sind sehr selten. Unter 15,2 Millionen möglichen abc-Tripeln bis $n=10000$ gibt es nur 120 Treffer, davon 91 verschiedene Treffer.

Bisher (Stand Feb. 2016) sind erst 237 abc-Treffer mit einer Qualität $q > 1.4$ entdeckt worden.

Hier die abc-Treffer bis 10000 (nur mögliche c-Werte, ohne mehrfach-Treffer):

```
{2, 9, 32, 49, 64, 81, 125, 128, 225, 243, 245, 250, 256, 289, 343, 375, 512, 513, 539, 6
25, 676, 729, 961, 968, 1025, 1029, 1216, 1331, 1369, 1587, 1681, 2048, 2057, 2187, 2
197, 2304, 2312, 2401, 2500, 2673, 3025, 3072, 3125, 3136, 3211, 3481, 3584, 3773, 3
888, 3969, 3993, 4000, 4096, 4107, 4131, 4225, 4235, 4375, 4913, 5041, 5120, 5312, 5
427, 5632, 5776, 5832, 6144, 6250, 6400, 6561, 6625, 6655, 6656, 6859, 6860, 6875, 6
912, 7744, 8000, 8019, 8192, 8576, 8748, 9261, 9317, 9375, 9376, 9409, 9801, 9826, 9
984, 10000}
```

Allgemeines

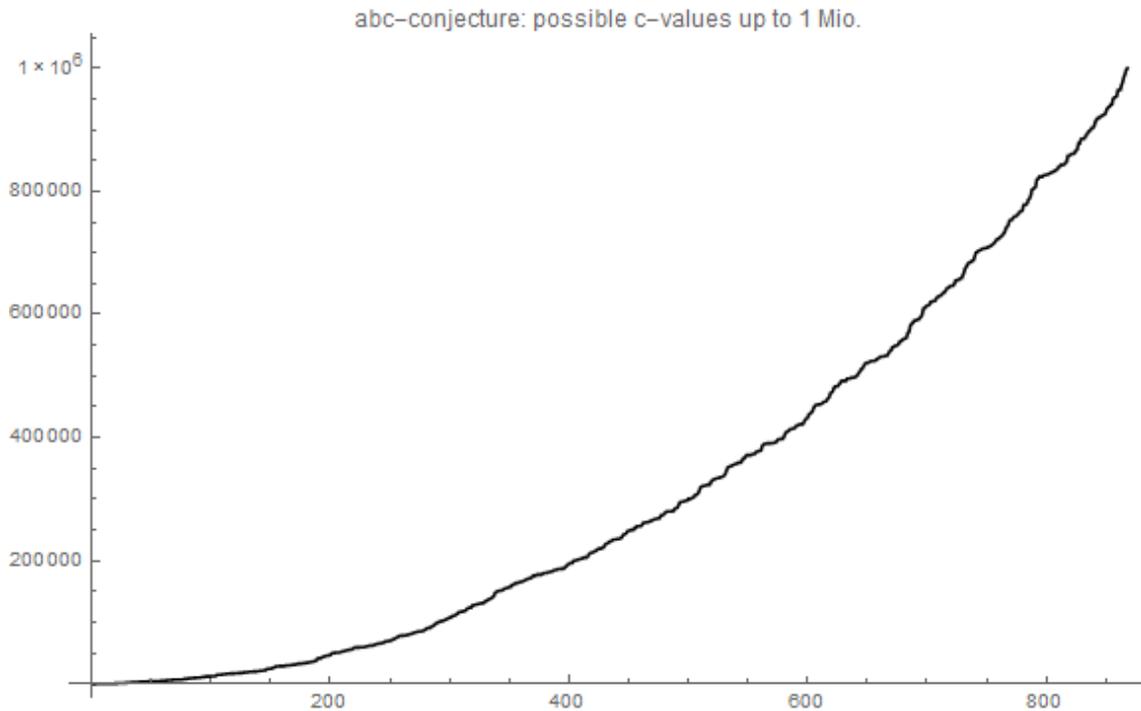


Abbildung 117: abc-Treffer: die ersten 868 möglichen c-Werte (9-1000000)

Man sieht deutlich, dass die abc-Treffer mit wachsender Größe immer seltener werden.

Unter den 380 Millionen möglichen abc-Tripeln unterhalb von 50000 gibt es 276 abc-Treffer.

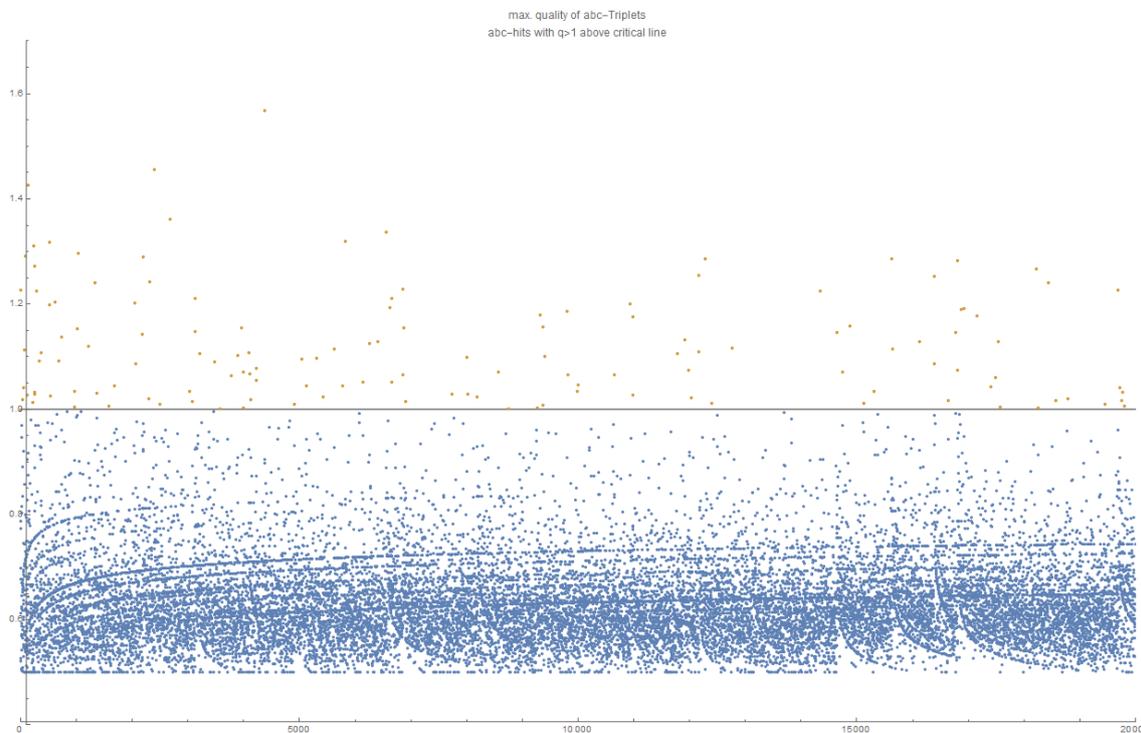


Abbildung 118: max. 'Qualität' von abc-Tripeln in Abhängigkeit von c (mit 138 abc-Treffern) im Bereich bis 20000

Man sieht deutlich, dass abc-Treffer sehr selten sind. Im ‚unkritischen‘ Bereich mit $q < 1$ sind deutliche Strukturen zu erkennen. Man wird das Gefühl nicht los, dass da noch einige Überraschungen warten...

Die ersten abc-Treffer im Bereich bis 2000 zusammen mit ihren Qualitäten (bei mehreren abc-Treffern für ein c wurde jeweils die maximale Qualität genommen) lauten:

```
{2,1.}, {9,1.22629}, {32,1.01898}, {49,1.04124}, {64,1.11269}, {81,1.29203}, {125,1.0272}, {128,1.42657}, {225,1.0129}, {243,1.3111}, {245,1.02883}, {250,1.03261}, {256,1.27279}, {289,1.22518}, {343,1.09175}, {375,1.10844}, {512,1.19875}, {513,1.31757}, {539,1.02512}, {625,1.20397}, {676,1.09219}, {729,1.13667}, {961,1.0048}, {968,1.03443}, {1025,1.1523}, {1029,1.29721}, {1216,1.1194}, {1331,1.24048}, {1369,1.02991}, {1587,1.00607}, {1681,1.04391}}
```

Mathematica-Programm: Bitte den Autor kontaktieren.

12.2 ABC-VERMUTUNG UND GOCRONS: EINE VERBINDUNG?

Die abc-Vermutung verknüpft die Welt der Addition mit der Welt der Multiplikation, indem sie vorhersagt, dass additive Verknüpfungen auch einen gewissen Einfluss auf die multiplikativen Strukturen der betrachteten Objekte hat. Was liegt deshalb näher als die abc-Vermutung mit solchen Objekten zu untersuchen, die multiplikative Eigenschaften perfekt repräsentieren: den OCRONS und GOCRONS (siehe 10.2.2). Wir wollen uns hier auf die in diesem Kapitel eingeführten Typ-4 OCRONS sowie deren Erweiterung (EOCRONS) beschränken. Da wir quantitative Aussagen machen wollen werden wir nicht Zeichenketten (also OCRONS) sondern deren ‚gödel-numerierete‘ Verwandte, die GOCRONS dafür verwenden.

Zunächst unsere additive Struktur:

Wir betrachten die Menge aller natürlichen Zahlen a und b , deren Summe einen festen Wert c ergibt: $a + b = c$, $a, b \geq 1$ und $a, b < c$ und insbesondere deren GOCRON4-Werte, die wir der Einfachheit halber $g(a)$, $g(b)$ und $g(c)$ nennen wollen, z.B. $g(a) = \text{nToEGOCRON4}(a)$ (mit der Funktion $\text{nToEGOCRON4}()$ aus der OCRON-Mathematica Bibliothek, die im Anhang zu finden ist). Den Einfluss der zusätzlichen Randbedingung der abc-Vermutung, dass a und b zueinander teilerfremd (‚koprim‘) sein sollen, werden wir auch mit untersuchen.

Wir interpretieren die Werte $\{g(a), g(b)\}$ als Punkte in der Ebene und ordnen ihnen einen Funktionswert $f[g(a), g(b)]$ zu, der der ganzen Sache eine multiplikative Struktur aufdrängt: $f[g(a), g(b)] = g(a \cdot b)$.

Da GOCRON-Werte schnell sehr groß werden können, betrachten wir lieber die logarithmierten Werte (was der Natur der Sache eher angemessen ist). Somit lässt sich unsere Aufgabenstellung wie folgt präzise beschreiben. Wir suchen nach einer Menge von ganzzahligen ‚abc-Tripeln‘, bei denen zwischen a und b eine additive Struktur existiert, dem dritten Wert jedoch ein multiplikativer Wert zugeordnet wird:

$$M_{abc} = \{\ln g(a), \ln g(b), \ln g(a \cdot b)\},$$

mit $g(x)$: Gödelcodes der OCRONs
sowie den Randbedingungen: $a + b = c$ und $a \perp b$

(154)

Schauen wir uns die Struktur der Menge M_{abc} an, dann erleben wir eine kleine Überraschung, denn die dreidimensionalen Punkte von M_{abc} liegen (mit einer Abweichung von etwa 2 bis 3 %) alle auf einer Ebene mit der unglaublich einfachen Ebenengleichung $z = x + y + const$, wobei der Wert $const$ nur von c abhängt!

$$M_{abc} \text{ wird näherungsweise beschrieben durch: } z=x+y+const \quad (155)$$

Diese scheint für alle $c \in \mathbb{N}$ zu gelten (bisher hat der Autor noch kein Gegenbeispiel gefunden). Wird die Randbedingung $a \perp b$ weggelassen, dann liegt M_{abc} bei beliebigen ganzzahligen c nicht auf einer Ebene. Die Strukturen scheinen für diesen allgemeinen Fall wesentlich komplizierter zu sein. Für den Fall, dass c eine Primzahl ist, liegt M_{abc} ebenfalls auf einer Ebene, auch wenn wir die Randbedingung $a \perp b$ weglassen, da diese Bedingung dann automatisch erfüllt wird (die zwei Summanden einer Primzahl sind automatisch teilerfremd).

Leider liegen die Punkte von M_{abc} nicht exakt auf einer Ebene, sondern mit einem ‚Rauschen‘ von ca. 2 bis 3 Prozent. Würde die Beziehung exakt gelten, dann hätten wir eine Methode gefunden die Faktorisierung einer Zahl (in unserem Fall c) allein aus den Werten a und b (bzw. $\ln g(a), \ln g(b)$) zu berechnen (oder umgekehrt aus c allein durch Projektion auf die $x - y$ -Achse), da der dritte Wert c mit Hilfe der Ebenengleichung $z=x+y+const$ nur durch eine einfache Addition zu lösen wäre (die ‚Rücktransformation‘ aus dem Gödelcode der faktorisierten Zahl in den ‚normalen‘ Zahlenbereich ist sehr einfach und ohne großen Aufwand möglich). Ganz allgemein könnte man sich auch eine Methode vorstellen, die einzig aus der Ebenengleichung allein nach ganzzahligen Werten a und b sucht, da die Bestimmung von $g(a)$ und $g(b)$ bei großen a und b sehr rechenintensiv sein kann.

Es besteht die Vermutung, dass bei einer anderen, besseren Wahl eines geeigneten GOCRON-Systems (bei der Berechnung und bei der Gödelnummerierung existiert ja eine gewisse Willkür bei der Wahl von Parametern), das beschriebene ‚Rauschen‘ kleiner wird, oder sogar zu Null wird. Hier gibt es noch viel zu tun!

Hier ein paar Plots von M_{abc} :

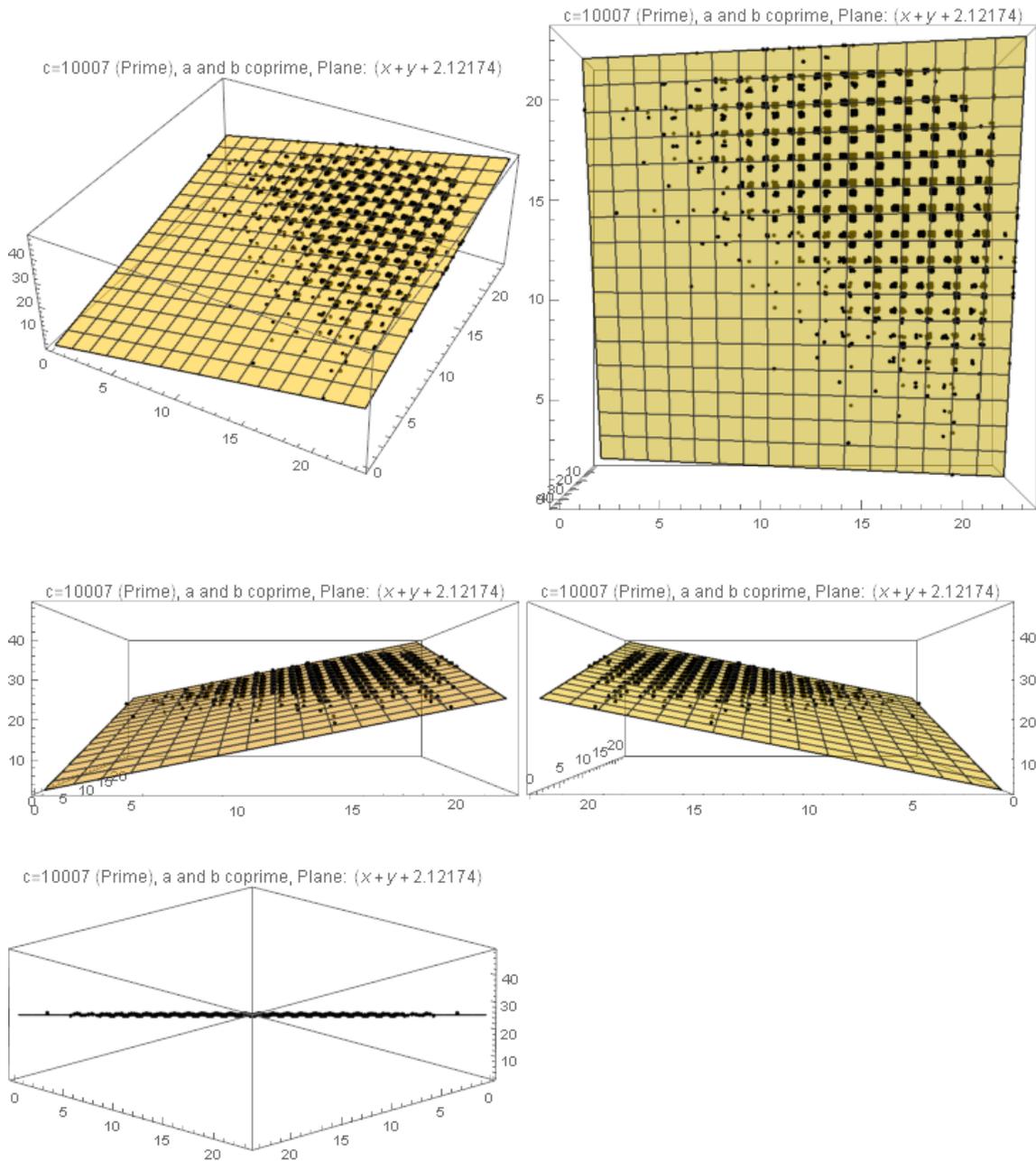


Abbildung 119: M_{abc} : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10007$ (Primzahl), a und b sind teilerfremd (verschiedene Ansichten). Die 10002 Punkte liegen etwa auf einer Ebene

Hinweis: der Plot in Abbildung 119 verändert sich nicht, wenn wir die Nebenbedingung $a \perp b$ weglassen, da 1007 eine Primzahl ist.

Mathematica-Programm: Bitte den Autor kontaktieren.

ABC-Vermutung und GOCRONs: Eine Verbindung?

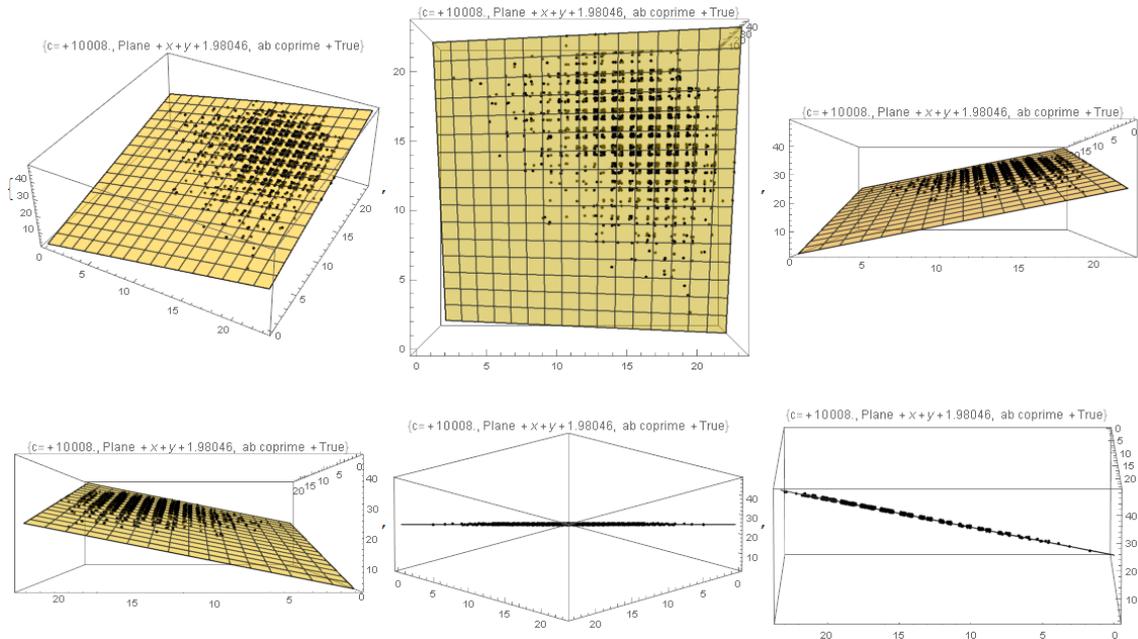


Abbildung 120: M_{abc} : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10008$, a und b sind teilerfremd (verschiedene Ansichten). Die 3310 Punkte liegen etwa in einer Ebene.

(Mathematica-Programme zur Berechnung im Anhang)

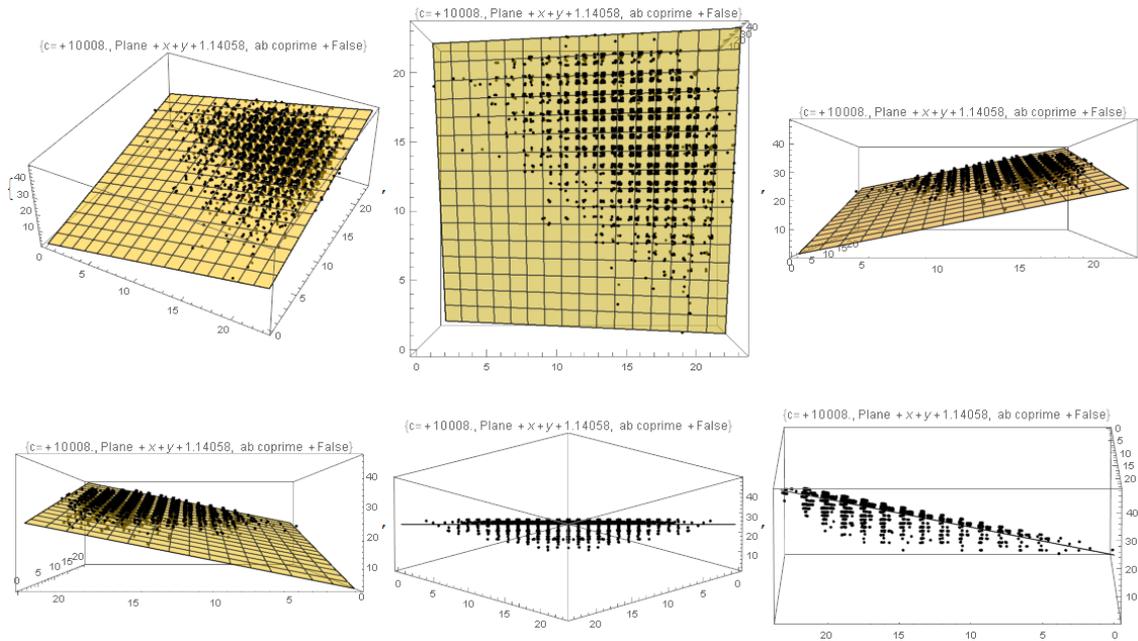


Abbildung 121: M_{abc} : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10008$, (verschiedene Ansichten). Die 10004 Punkte sind räumlich verteilt

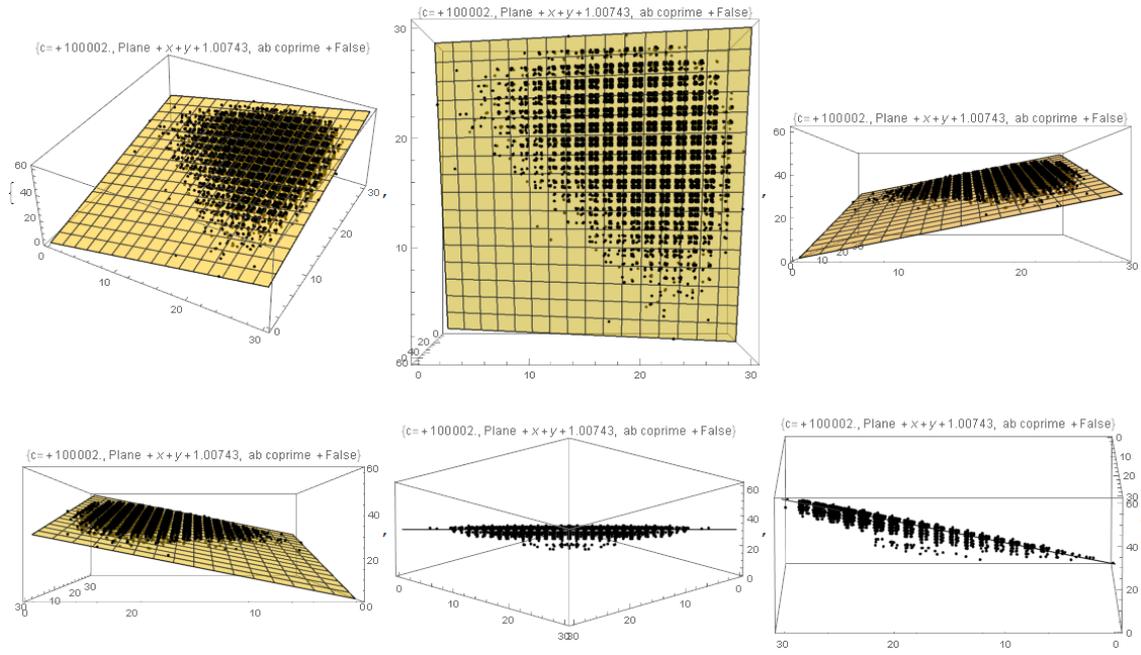


Abbildung 122: M_{abc} : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=100002$, (verschiedene Ansichten). Die 99998 Punkte liegen räumlich verteilt.

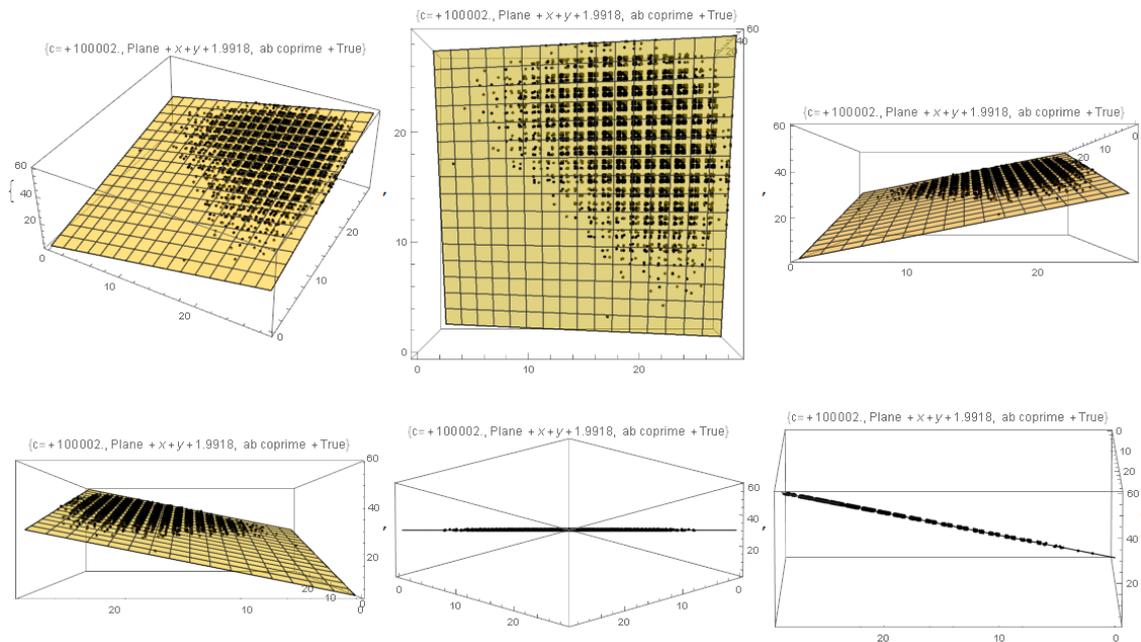


Abbildung 123: M_{abc} : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=100002$, a und b teilerfremd (verschiedene Ansichten). Die 28558 Punkte liegen etwa auf einer Ebene.

Die Menge M_{abc} und ihre Ebenengleichung

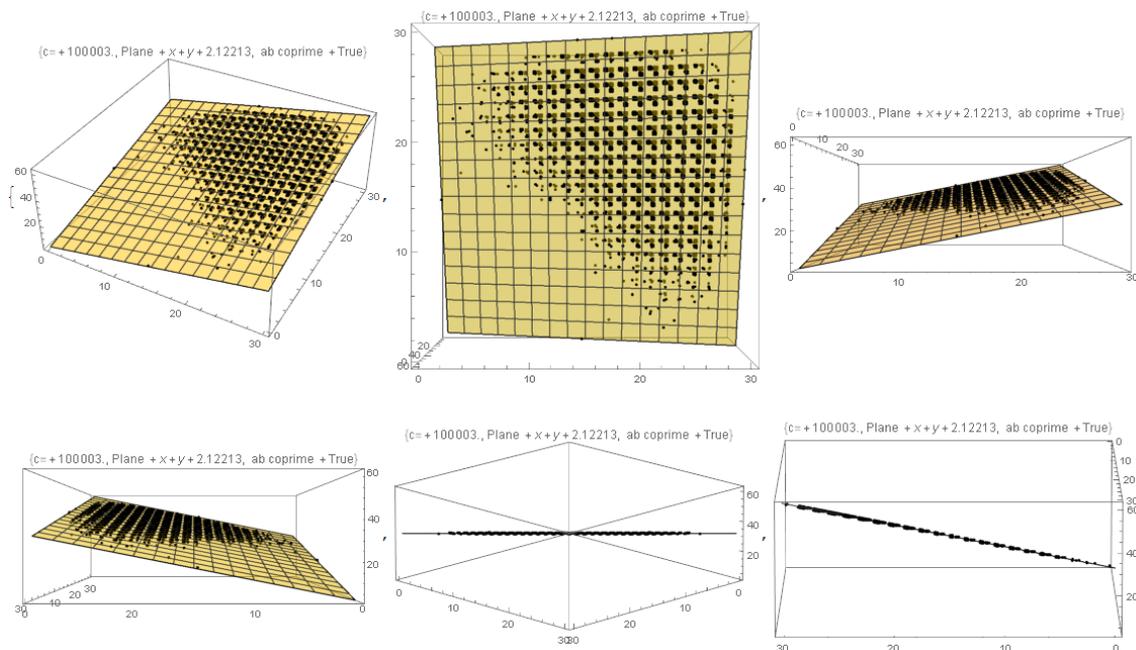


Abbildung 124: M_{abc} : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10007$ (Primzahl), a, b sind teilerfremd (verschiedene Ansichten). Die 100003 Punkte liegen etwa auf einer Ebene.

12.3 DIE MENGE M_{abc} UND IHRE EBENENGLEICHUNG

Wie wir im letzten Kapitel gesehen haben, liegen die Punkte M_{abc} näherungsweise auf einer Ebene. Diesen Sachverhalt wollen wir jetzt etwas genauer unter die Lupe nehmen. Zunächst stellen wir fest, dass das Ebenen-Prinzip für alle Varianten von GOCRONS vom Typ 4 gilt. Für die normalen GOCRONS, M2GOCRONS (die gehören zu den OCRONS, bei denen die immer führende ‚2‘ weggelassen wurde, sowie EOGRONS (die erweiterten GOCRONS)). Der Unterschied zwischen den verschiedenen Typen liegt im Wertebereich (die normalen GOCRONS sind mindestens eine Größenordnung größer als ihre Verwandten) und in ihrer ‚Rücktransformierbarkeit‘ in den Bereich der normalen Zahlen (die EGOCRONS sind für jeden beliebigen ganzzahligen Wert >2 rücktransformierbar). Wir untersuchen nun die Abhängigkeit der Parameter der entsprechenden Ebenengleichung vom Wert c sowie von verschiedenen weiteren Parametern, die bei der Berechnung der Punkte M_{abc} eine Rolle spielen könnten. Als Kriterium für eine ‚gute‘ Parameterwahl nehmen wir den ‚Standard-Fehler‘, der sich aus der Least-Square-Methode bei der Ebenenberechnung aus der Menge M_{abc} ergibt. Wir nehmen dafür die Mathematica-Funktion `NonlinearModelFit[...]` mit dem Model: $z = x + y + c_3$.

Im Vergleich der verschiedenen Gödel-Code-Symbole bei der Umwandlung von OCRONS in GOCRONS stellt sich heraus, dass für die (normalen) GOCRONS vom Typ 4 von den 24 möglichen Permutationen der Menge von Code-Symbolen

$\{ " * ", "P", "2", "^" \} \rightarrow \{0,2,3,1\}$ und $\{ " * ", "P", "2", "^" \} \rightarrow \{1,2,3,0\}$ die besten Ergebnisse liefern. Hier eine Tabelle:

Tabelle 25: $c=100003$. Fitparameter und c_3 der Ebenengleichungen für M_{abc} (Typ GOCRON4) für verschiedene Sätze von Gödelsymbolen

C	c_3	Codetable: Symbole/Werte	Max. Wert	Standardfehler	t-Statistik
100003	1.85733	1: {*, P, 2, ^}, {0, 1, 2, 3}	37.0455	0.000162446	11433.5
100003	1.48688	2: {*, P, 2, ^}, {0, 1, 3, 2}	37.4114	0.000186537	7970.96
100003	2.34062	3: {*, P, 2, ^}, {0, 2, 1, 3}	36.423	0.000200917	11649.7
100003	1.44022	4: {*, P, 2, ^}, {0, 2, 3, 1}	37.3964	0.0000736448	19556.3
100003	2.24673	5: {*, P, 2, ^}, {0, 3, 1, 2}	36.3821	0.000422527	5317.36
100003	1.73226	6: {*, P, 2, ^}, {0, 3, 2, 1}	37.0015	0.000153876	11257.5
100003	1.93765	7: {*, P, 2, ^}, {1, 0, 2, 3}	37.0444	0.000332613	5825.54
100003	1.54166	8: {*, P, 2, ^}, {1, 0, 3, 2}	37.4107	0.000303486	5079.82
100003	4.00103	9: {*, P, 2, ^}, {1, 2, 0, 3}	34.4163	0.00246733	1621.61
100003	1.44418	10: {*, P, 2, ^}, {1, 2, 3, 0}	37.3804	0.0000698268	20682.3
100003	3.79035	11: {*, P, 2, ^}, {1, 3, 0, 2}	34.4462	0.00303396	1249.31
100003	1.738	12: {*, P, 2, ^}, {1, 3, 2, 0}	36.9777	0.000172422	10079.9
100003	2.61489	13: {*, P, 2, ^}, {2, 0, 1, 3}	36.4189	0.000393053	6652.77
100003	1.54569	14: {*, P, 2, ^}, {2, 0, 3, 1}	37.3949	0.000294348	5251.23
100003	4.50595	15: {*, P, 2, ^}, {2, 1, 0, 3}	34.4009	0.00207401	2172.57
100003	1.49486	16: {*, P, 2, ^}, {2, 1, 3, 0}	37.3796	0.000170018	8792.37
100003	3.93228	17: {*, P, 2, ^}, {2, 3, 0, 1}	34.4473	0.00352994	1113.98
100003	2.26812	18: {*, P, 2, ^}, {2, 3, 1, 0}	36.2903	0.000484516	4681.22
100003	2.62617	19: {*, P, 2, ^}, {3, 0, 1, 2}	36.3757	0.00036176	7259.43
100003	1.94952	20: {*, P, 2, ^}, {3, 0, 2, 1}	36.9981	0.000303674	6419.8
100003	4.62145	21: {*, P, 2, ^}, {3, 1, 0, 2}	34.018	0.00222471	2077.33
100003	1.87489	22: {*, P, 2, ^}, {3, 1, 2, 0}	36.9753	0.000122802	15267.6
100003	4.25883	23: {*, P, 2, ^}, {3, 2, 0, 1}	34.0449	0.00320799	1327.57
100003	2.37305	24: {*, P, 2, ^}, {3, 2, 1, 0}	36.288	0.000280365	8464.13

Weitere Tabellen mit verschiedenen Methoden der Gödelisierung und verschiedene Werte von c finden sich im Anhang.

Eine Auswertung dieser Tabellen zeigt, dass die CodeTable Nr. 10 der (wenn auch knappe) ‚Sieger‘ ist (im Hinblick auf den kleinsten Standard-Fehler), falls die Gödelnummerierung der OCRONs in der normalen Reihenfolge (links nach rechts) erfolgt. Im Fall der umgekehrten Reihenfolge ist die Auswahl eines besten Gödelcode-Satzes nicht so eindeutig. Im Folgenden werden wir deshalb für alle OCRON4-Typen die Gödel-Codes $\{ '*', 'P', '2', '^' \rightarrow \{1,2,3,0\}$ und die normale Symbolreihenfolge (nicht ‚reversed‘) benutzen.

Das Programm für die Auswertung ist im Anhang aufgelistet.

Die Menge M_{abc} und ihre Ebenengleichung

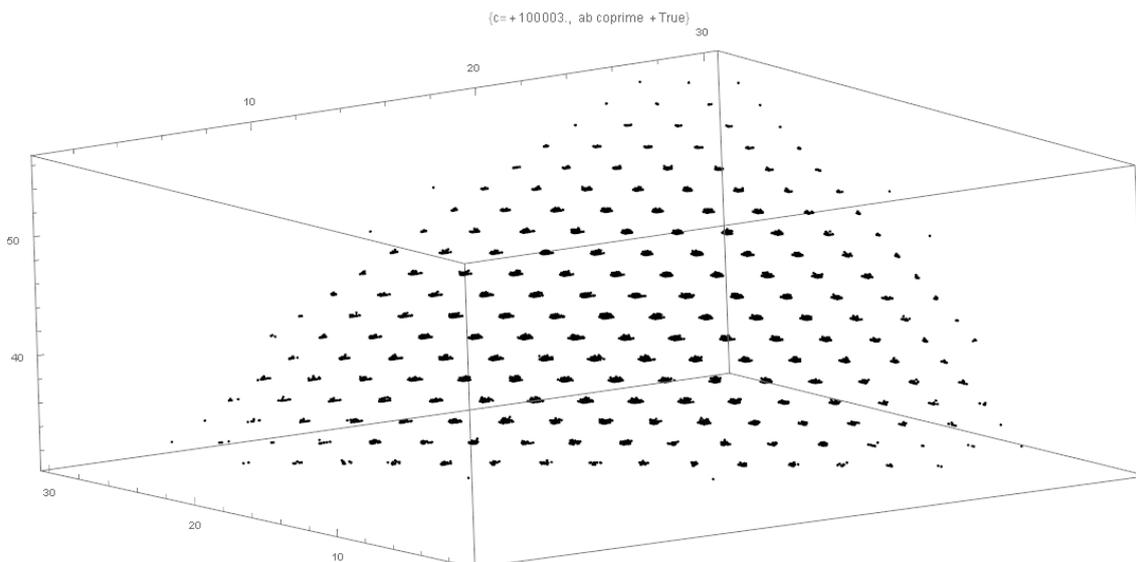


Abbildung 125: Ebene von M_{abc} -Punkten für $c=100003$ (Primzahl)

Mathematica-Programm: Bitte den Autor kontaktieren.

Die nächste Tabelle zeigt den Zusammenhang zwischen c und dem Ebenenparameter c_3 mit CT: Nr. der Gödel CodeTable, GT: GOCRON-Typ (N: normal, M2: ohne ‚2‘ am Anfang, E: Enhanced).

Tabelle 26: Verschiedene c_3 -Werte für verschiedene GOCRONS und Gödelsymbole

c	c_3 (CT:10, GT: N)	c_3 (CT:10, GT: E)	c_3 (CT:10, GT: M2)	c_3 (CT:4, GT: M2)	c_3 (CT:9, GT: M2)	c_3 (CT:12, GT: M2)
10009	1.44418	1.65067	3.0335	3.01507	4.00661	2.9951
30011	1.44423	1.65087	3.03373	3.01535	4.00661	2.99422
100003	1.44418	1.65062	3.03347	3.01512	4.00103	2.99437
1000003			3.03335	3.01497	4.00219	2.9946

Zusammenfassung:

Die Punkte der Menge M_{abc} liegen (mit einer Abweichung von 2- 3 %) auf einer Ebene mit der Gleichung $z = x + y + c_3$ (c_3 siehe obige Tabelle). Da bei der Berechnung der Punkte die logarithmischen Werte genommen werden (siehe (154)) ist die Abweichung von den ‚gefitteten‘ Werten der Ebene für die ‚un-logarithmierten‘ Werte natürlich wesentlich größer. In diesen Abweichungen von den interpolierten Werten der Ebenengleichung steckt sozusagen noch viel Struktur drin, die nach Gesetzmäßigkeiten untersucht werden müsste. Dieses einfache Modell der Ebenengleichung ist bei der Suche nach Faktoren einer gegebenen Zahl noch keine Hilfe. Dafür müssten die Punkte viel genauer auf der Ebene liegen.

Interessant ist jedoch die Tatsache, dass die Ebenen-Struktur nur auftaucht, wenn die Beziehung $\mathbf{a} + \mathbf{b} = \mathbf{c}$ und $\mathbf{a} \perp \mathbf{b}$ gilt. Lassen wir z. B. die Randbedingung $\mathbf{a} \perp \mathbf{b}$ weg, dann entsteht keine Ebene, sondern eine weit verstreute komplizierte räumliche Struktur (siehe z.B. ABBILDUNG 121).

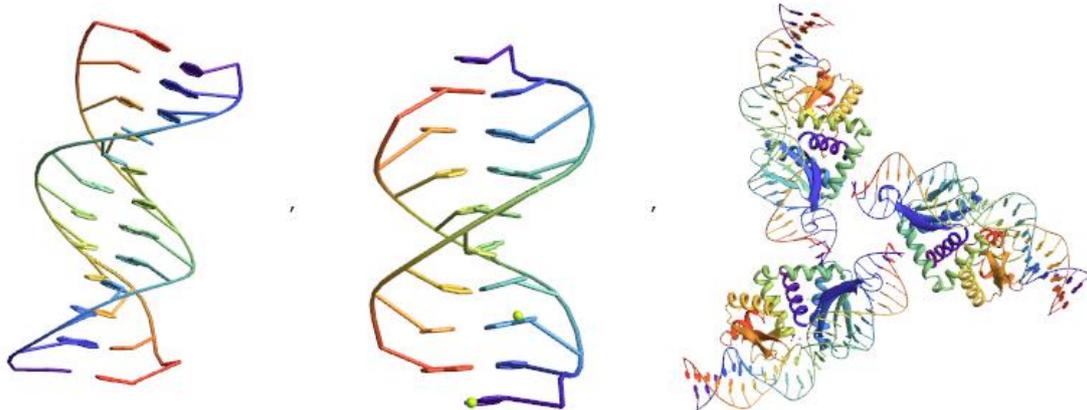
Diese Tatsache deutet auf einen Zusammenhang zur abc-Vermutung hin

.

13 PRIMZAHLEN IN DEN NATURWISSENSCHAFTEN

13.1 PRIMZAHLEN IM DNA-CODE

Viele Wissenschaftler sind der Meinung, dass Primzahlen auch in der Konstruktion und Gestaltung von DNA-Sequenzen eine Rolle spielen. Jeder kennt die berühmte Doppel-Helix der DNA. Hier ein paar willkürlich ausgewählte Beispiele:



Mathematica:

```
Import[#, "PDB"] & /@ {"http://files.rcsb.org/download/1BNA.pdb", "http://files.rcsb.org/download/208D.pdb", "http://files.rcsb.org/download/5A0W.pdb"}
```

Der folgende Text stammt im Wesentlichen aus einer Arbeit von J.F. Yan, A.K. Yan und B.C. Yan⁶³.

Wir verwenden hier die (weiter verbreiterte) englische Bezeichnung DNA (=deoxyribonucleic acid) statt der deutschen Version DNS (=Desoxyribonucleinsäure). Bevor wir die dahinterliegende Idee darstellen, ist es nützlich, sich ein paar grundlegende Eigenschaften der DNA und der RNA anzuschauen.

Die DNA ist der Träger der genetischen Erbinformation, die in Zellkernen aller Lebewesen vorhanden ist. Hierbei handelt es sich um riesige Kettenmoleküle, die aus vier verschiedenen Grundbausteinen aufgebaut sind: Den Nucleotiden. Jedes Nucleotid besteht aus einer Phosphat-Gruppe, einem Zucker (Desoxyribose) und einer von vier möglichen organischen Base (Adenin, Guanin, Cytosin und Thymin).

Die RNA (Ribonucleinsäuren) werden mit Hilfe der Information der DNA hergestellt. Für den genauen Mechanismus sei der Leser auf geeignete Literatur verwiesen⁶⁴. Die RNA ist ähnlich wie die DNA aus Nucleotiden aufgebaut, die wiederum aus vier verschiedenen organischen Basen bestehen (Adenin, Guanin, Cytosin und Uracil).

⁶³ J. theor. Biol. (11991) 151,333-341

⁶⁴ Z. B. <http://www.dna50.org/>

Die ersten drei Basen kommen in der RNA und in der DNA vor, die Base Thymin kommt jedoch nur in der DNA und Uracil nur in der RNA vor.

Die organischen Basen werden üblicherweise mit den Buchstaben A, G, C, T und U abgekürzt. Dies sind die kleinsten ‚Informationsbits‘ der DNA und der RNA. Die Unterschiede im Aufbau der RNA und DNA (Uracil statt Thymin) spielen für den Informationsgehalt, den wir betrachten wollen, keine Rolle. Wir verwenden deshalb der Einfachheit halber nur die vier Basen A, G, C und U (wobei wir in der DNA statt U streng genommen die Bezeichnung ‚T‘ verwenden müssten).

Jedes aufeinander folgende Triplet von Basen (auch Codon genannt) wird genau einer bestimmten Aminosäure zugeordnet. Die Umkehrung gilt nicht: Mehrere verschiedene Triplets können der gleichen Aminosäure zugeordnet werden. Diese Triplets sind die kleinsten Informationsbausteine der DNA und der RNA. Einige wenige Triplets (Codons) enthalten keine genetische Information. Sie fungieren jedoch bei den Replikations- und Kopiervorgängen, die während der Proteinsynthese ablaufen, als sogenannte ‚Stopp‘-Triplets. Es gibt auch Start-Triplets, die jedoch auch Information enthalten: z.B. das Triplet ‚AUG‘ (Methionin-Aminosäure).

Bei vier verschiedenen Basen gibt es für die Tripletbildung $4^3 = 64$ verschiedene Möglichkeiten. Die praktisch in der Natur vorkommenden (kanonischen, proteinogenen) Aminosäuren werden mit Hilfe von 61 Triplets kodiert. Die drei übrig bleibenden Triplets sind ‚Stop‘-Triplets. Eine sehr gute Darstellung findet man im Internet.⁶⁵ Manche Aminosäuren werden nur durch ein Triplet codiert, manche durch 2,3, 4 oder 6 verschiedene Triplets. Hinweis: Die Zahl 5 fehlt als Anzahl der aminosäurebildenden Triplets.

Was hat dies alles mit Primzahlen zu tun? Wie gelangen wir von Basen-Triplets zu Zahlenwerten? Die Antwort ist ähnlich wie im Kapitel 10.2.2.4: Durch eine ‚Gödelisierung‘. Wir ordnen jedem Basen-Triplett (unserer kleinsten Informationseinheit) einen Zahlenwert zu. Hierfür gibt es mehrere Möglichkeiten:

Klaus Lange benutzt in seiner Arbeit ‚Primzahlen im Aufbau der DNS‘⁶⁶ $G=1$, $A=3$, $C=7$, $U=9$ und bildet die Zahlen-Codes, in dem er einfach das Zehner-Dezimalsystem benutzt. Ein Beispiel: Alanin (=GCA mit dem Codewert 173). Er untersucht dann die sich ergebenden Zahlenwerte auf ihre Primfaktorenzerlegung und stellt fest, dass von den 20 kanonischen Aminosäuren 19 mindestens ein Triplet enthalten, das eine Primzahl darstellt. Auffällig ist für ihn, dass die einzige Aminosäure, die keiner Primzahl als Triplet zugeordnet werden kann Methionin ist (Met=AUG). Nach seinem System hat dieses Triplet den Wert 391. Methonin funktioniert als einziges Triplet als sogenanntes ‚Startsignal‘ (s. oben).

Diese Zahlenzuordnung erscheint dem Autor etwas willkürlich. Besser ist die Methode von Yan (s. oben), bei der einfach gesetzt wird:

$A=0$, $C=1$, $U=2$, $G=3$. Aus dieser Zuordnung wird dann eine eindeutige Menge von

⁶⁵ https://de.wikipedia.org/wiki/Genetischer_Code

⁶⁶ <http://www.primzahlen.de/primzahlen/dns.htm>

Primzahlen im DNA-Code

„Nucleotid-Zahlen“ (die jeweils einer Aminosäure zugeordnet werden können) konstruiert. Sonderfälle sind die Aminosäuren:

- 0: Für Stop-Signal (ohne Namen, UAA, UAG, UGA)
- 1: Für Tryptophan (Trp, auch Try), UGG
- 2: Für Isoleucin (Ile), AUA, AUC, AUU
- 3: Für Methionin (Met), AUG

Das Prinzip zur Konstruktion einer Nucleotid-Zahl z ($z < 64$) ist wie folgt:

Regel 1: z muss ungerade oder 2 sein. Die Werte 1 und zwei werden für AAX-Triplets reserviert.

Regel 2: Benennen wir Primzahlen der Form $4n+1$ als P1-Primzahlen und Primzahlen der Form $4n+3$ als P3-Primzahlen. Aus der Zahlentheorie wissen wir, dass P1-Zahlen sich immer eindeutig als Summe von zwei Quadraten ausdrücken lassen. Zwischen 0 und 63 gibt es insgesamt 8 P1-Primzahlen.

Alle P1-Primzahlen < 64 , die als Summe von 2 Quadraten dargestellt werden können, sind äquivalent mit den vierfach degenerierten „Synonym“-Codons die entstehen, wenn wir die ersten beiden Basen (von den drei möglichen) vorgeben.

Die weitere Vorgehensweise ist, alle 64 möglichen Codons in 4 Gruppen zu unterteilen.

Gruppe 1 sind die diophantischen Lösungen der Gleichung $z = (2i + 1)^2 + (2j)^2$

Nucleotid Zahlen	Synonyme Codons (X=A,C,U oder G)	Name d. Aminosäure (Abk)
$1^2 + 2^2 = 5$	ACX	Thr
$3^2 + 2^2 = 13$	CCX	Pro
$5^2 + 2^2 = 29$	UCX	Ser
$7^2 + 2^2 = 53$	GCX	Ala

Gruppe 2 sind die diophantischen Lösungen der Gleichung $z = (2i - 1)^2 + (2j)^2$

Nucleotid Zahlen	Synonyme Codons (X=A,C,U oder G)	Name d. Aminosäure (Abk)
$1^2 + 4^2 = 17$	CUX	Leu
$1^2 + 6^2 = 37$	CGX	Arg
$5^2 + 2^2 = 29$	GUX	Val
$7^2 + 6^2 = 61$	GGX	Gly

Gruppe 3: Für die Bestimmung der Werte in dieser Gruppe verwenden Yan et. Al, etwas heuristische (und aus der Chemie entlehnte) Argumente, die der Leser gerne nachvollziehen kann (s. Literaturverweis).

Gruppe 4 sind die diophantischen Lösungen der Gleichungen $z = 4(2i + 1) + 3$ sowie $z = 8(2i + 1) + 3$.

Schließlich ergibt sich für alle kanonischen Aminosäuren folgende Codezuordnung:

Tabelle 27: Primzahl-Codierung der kanonischen Aminosäuren nach Yan et. al

0 (Stop)	1 (Try)	2 (Ile)	3 (Met)	5 (Thr)
	7 (Lys)	11 (Asn)	13 (Pro)	17 (Leu)
	19 (Gln)	29 (Ser)	31 (Asp)	37 (Arg)
	41 (Val)	43 (Tyr)	47 (His)	53 (Ala)
	59 (Glu)	61 (Gly)	25 (Phe)	45 (Cys)

Man beachte das Yan zwei Codes verwendet, die keine Primzahlen sind: (25: Phe und 45: Cys).

Mit Hilfe dieser Codes leiten Yan et. al Codierung-Eigenschaften und -Strategien ab, die zur Erläuterung ein tieferes Verständnis der Genetik bedürfen, was jedoch hier zu weit führen würde.

13.2 SPEKTRALE EIGENSCHAFTEN VON ‚PRIMZAHL-SIGNALEN‘

Primzahlen zeigen eine gewisse Ähnlichkeit mit statistischen Daten physikalischer Experimente. Diese Ähnlichkeit kommt vermutlich von ihrer ‚Unberechenbarkeit‘ (natürlich nicht im strengen mathematischen Sinne).

Informationstheoretisch können wir ein Primzahl-Signal wie folgt definieren:

$$x_i = \pi((i + 1)M) - \pi(i \cdot M), \text{ mit einer festen Intervalllänge } M.$$

Physiker beschäftigen sich gerne mit der Auswertung von Signalen. Wenn wir unser ‚Primzahl-Signal‘ mit physikalischen Methoden untersuchen, dann betreiben wir zwar keine ‚echte‘ Physik, jedoch so etwas ähnliches wie ein physikalisches Gedankenexperiment. Stellen wir uns vor wir empfangen auf der Suche nach extraterrestrischen Lebensformen mit Hilfe von Radiosignalen ein derartiges Signal (auf die Einzelheiten der Modulation wollen wir hier nicht eingehen; wir nehmen einfach an, es wäre ein digitales Signal, aus dem die Zahlenwerte der Folge x_i extrahiert werden könnte).

Hier ein Beispiel mit $M = 2^{16}$, das folgendes ‚Signal‘ ergibt:

```
{4533, 4454, 4486, 4430, 4460, 4446, 4446, 4442, 4438, 4421, 4446, 4401, 4376, 4417,
, 4358, 4384, 4435, 4386, 4355, 4344, 4360, 4258, 4337, 4354, 4394, 4283, 4339, 4343,
, 4255, 4354, 4294, 4307, 4289, 4237, 4285, 4327, 4283, 4266, 4258, 4285, 4244, 4256,
, 4301, 4281, 4228, 4233, 4232, 4243, 4261, 4207, 4240, 4210, 4198, 4202, 4197, 4196,
, 4188, 4221, 4239, 4217, 4128, 4220, 4157, 4226, 4209, 4128, 4148, 4195, 4230}
```

Mathematica:

```
intervalLength=2^16; startValue=32; endValue=100;
pSignal[j_,m_] := PrimePi[(j+1)*m] - PrimePi[j*m]
signal=Table[pSignal[k,intervalLength], {k, startValue, endValue}]
```

Machen wir nun ein ‚Primzahl-Experiment‘, in dem wir ein Primzahl-Signal erzeugen und es mit physikalischen Methoden, wie z. B. der Spektralanalyse untersuchen. Dieses Signal hat folgendes Aussehen:

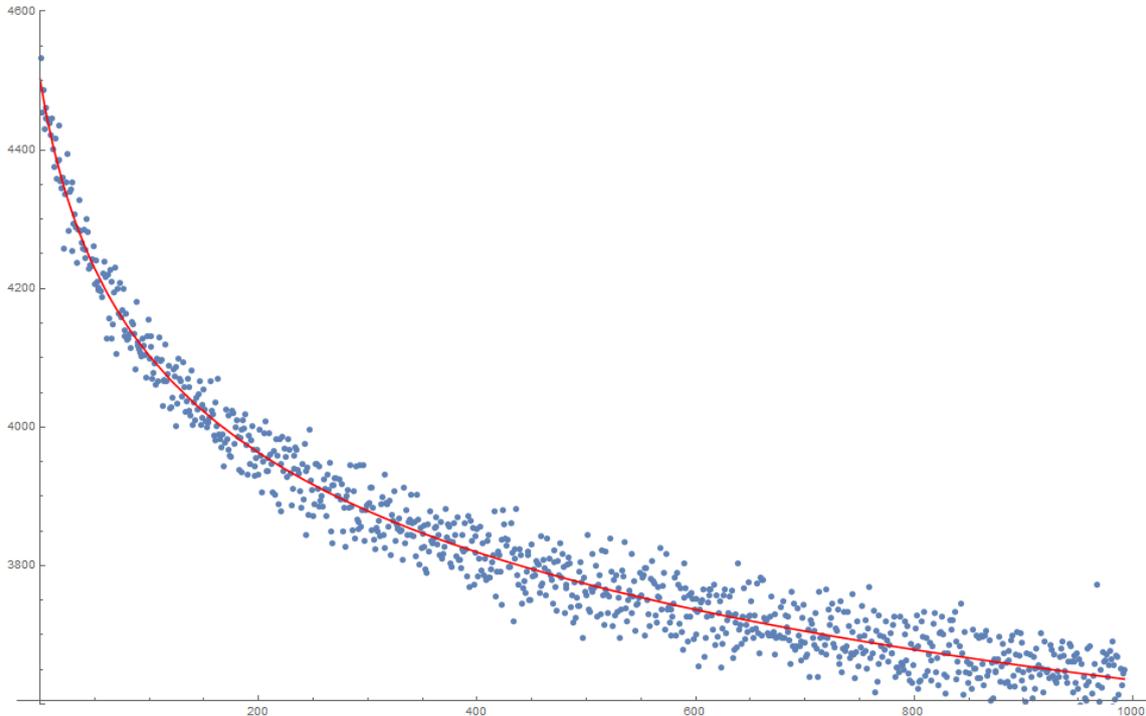


Abbildung 126: Primzahlsignal x_i mit Intervalllänge $M=2^{16}$. Rot: $\frac{M}{\ln Mi}$

Aus der Zahlentheorie wissen wir, dass das asymptotische Verhalten von x_i wie folgt ist: $x_i = \frac{M}{\ln Mi}$. Die rote Kurve zeigt das asymptotische Verhalten.

Wir wenden nun auf das Signal x_i eine diskrete Fouriertransformation (DFT) an und bekommen dann im Frequenzbereich:

$$X_k = \sum_{j=0}^{N-1} x_j e^{-\frac{2\pi ijk}{N}} \quad (156)$$

Wobei N die Länge unseres Primzahlsignals (z. B. 2^{16}) ist. Den Physiker interessiert nun die spektrale Leistungsdichte

$$S_k = |X_k|^2$$

Schauen wir uns diese spektrale Leistungsdichte im logarithmischen Maßstab an, dann erleben wir wieder eine kleine Überraschung, denn diese ist über weite Bereiche durch eine Gerade zu approximieren. Das bedeutet, dass die spektrale Leistungsdichte unseres Primzahlsignals proportional zu $1/k^\alpha$, mit einem konstanten Exponenten α ist:

$$S_k \sim \frac{1}{k^\alpha} \quad \text{mit } \alpha \approx 1.55 \quad (157)$$

In der Literatur wird der Wert $\alpha \approx 1.64$ angegeben⁶⁷. Der Autor kommt bei seinen Berechnungen auf den ungefähren Wert 1.55.

Dieses Verhalten ist den Physikern für eine Gruppe von physikalischen Systemen, nämlich solchen, die sich in einem sogenannten selbstorganisierten kritischen Zustand befinden, bestens bekannt. Viele weitere physikalische Systeme zeigen ein spektrales Verhalten nach dem $\frac{1}{k^\alpha}$ Gesetz: Spannungs-Rauschen in elektronischen Bauteilen (Flicker-Rauschen). Es gibt auch Untersuchungen über das statistische Verhalten von Tönen (innerhalb unseres 12-stufigen tonalen musikalischen Systems), die ein statistisches $1/f$ -Verhalten von traditionell komponierter (keiner Zufallsmusik) nachweisen. Somit haben wir eine (wenn auch entfernte) Verbindung zwischen Primzahlen und tonaler Musik hergestellt!

Hier ein Graph der spektralen Leistungsdichte eines Primzahl-Signales:

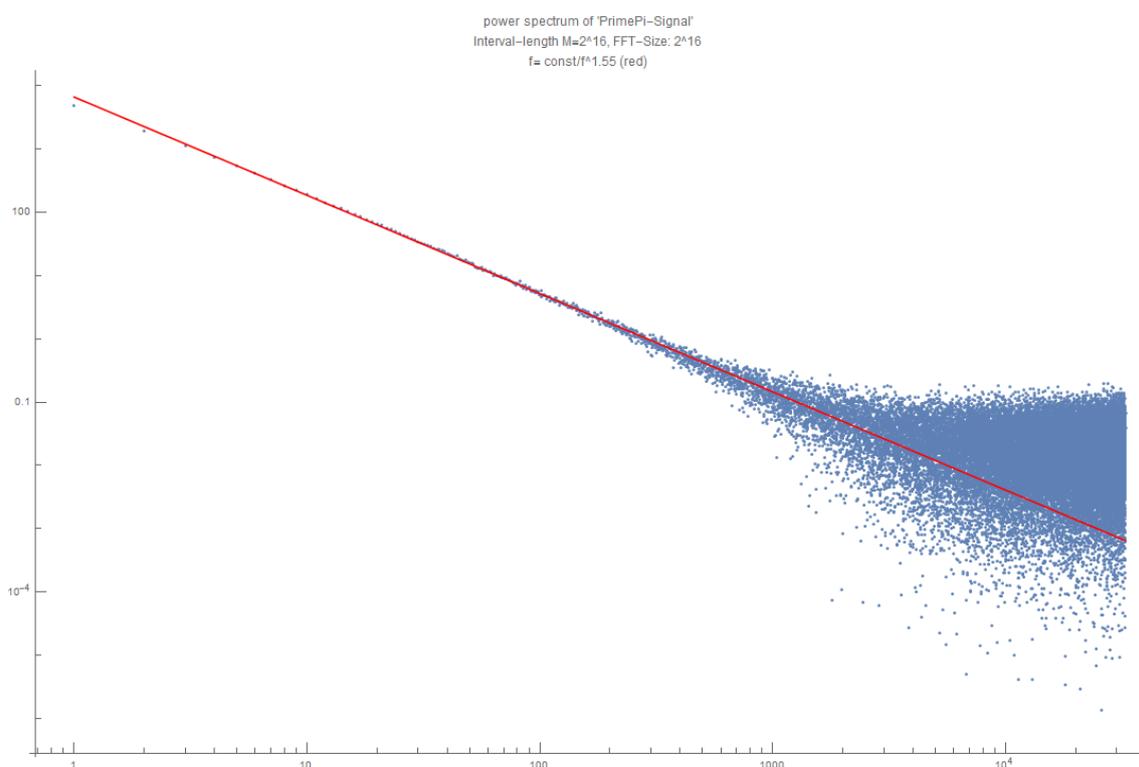


Abbildung 127: Spektrale Leistungsdichte eines Primzahl-Signales (rot: $\frac{1}{k^\alpha}$ mit $\alpha=1.55$)

Mathematica-Programm: Bitte den Autor kontaktieren.

14 PRIMZAHLEN UND ONLINE-BANKING

14.1 DIE RSA-VERSCHLÜSSELUNG

⁶⁷ Marek Wolf: PHYSICA A: Statistical Mechanics and its applications · January 1997, 493-499

Primzahlen haben längst Einzug in die Kryptographie, der ‚Wissenschaft vom Verschlüsseln‘, gehalten. Im Online Banking werden ständig hochvertrauliche Daten hin und hergeschickt. Die Menge an Transaktionen ist so gigantisch groß, dass eine symmetrische Verschlüsselung (wo beide Partner einen geheimen Schlüssel besitzen) nicht praktikabel wäre, da für jede Transaktion der geheime Schlüssel auf einem sicheren Übertragungsweg (z. B. per Briefpost) vor der eigentlichen Transaktion verschickt werden müsste. Das ist praktisch unmöglich. Es gibt jedoch ein Verfahren, das das umständliche Verschicken von Schlüsseln per Briefpost vermeidet: Die sogenannte asymmetrische RSA-Verschlüsselung.

Wir wollen hier kurz das RSA-Verfahren beschreiben. In der Abkürzung ‚RSA‘ verbergen sich die Namen der drei Informatiker, die das Verfahren als erste im Jahr 1978 praktisch implementierten: Rivest, Shamir und Adleman. Die ursprüngliche Idee dieses Verfahrens stammt jedoch von Diffie & Hellman und datiert aus dem Jahr 1976. Dieses Verschlüsselungsverfahren wird ‚asynchron‘ genannt, weil Sender und Empfänger von verschlüsselten Nachrichten unterschiedliche Schlüssel (die öffentlich bzw. geheim sind) benutzen. Der Sender benutzt einen öffentlichen ‚public key‘ zum Verschlüsseln und Senden und der Empfänger benutzt einen streng geheimen ‚private key‘ zum Empfangen und Entschlüsseln der Nachricht. Damit der Sender derart verschlüsselte Nachrichten an einen Empfänger schicken kann, muss zuerst der Empfänger einen öffentlichen (nicht geheimen) Schlüssel erzeugen und diesen dann an den Sender schicken, der ihn dann zum Versenden der Nachricht benutzt. Dieses Versenden des öffentlichen Schlüssels kann natürlich unverschlüsselt geschehen.

Da Texte verschlüsselt werden sollen, betrachten wir nicht die einzelnen Zeichen, sondern deren ASCII-Codes⁶⁸. Diese ordnen jedem Zeichen einen Wert zwischen 32 (Leerzeichen), 65 (‚A‘), 66 (‚B‘), usw. bis 90 (‚Z‘) zu. Natürlich werden auch Kleinbuchstaben durch höhere Zahlen codiert. Für das Verständnis des Verfahrens ist dies jedoch unerheblich.

Der zu verschlüsselnde Text wird zunächst mit Hilfe des ASCII Codes in eine lange Zahlenreihe dieser Codes übersetzt. Danach werden aus dieser Zahlenreihe Blöcke einer festen Länge (z.B. der Länge 64) gebildet. Jeder dieser Blöcke wird als eine (in diesem Fall 64-stellige) Zahl in einem Stellenwertsystem mit der Basis 256 interpretiert. Die Bildung von Blöcken geschieht nur aus Gründen der ‚Handlichkeit‘ um zu große ‚Zahlenmonster‘ zu vermeiden. Die Wahl der Basis ist nicht wichtig und kann auch kleiner ausfallen, falls wir einen kleineren Zeichenvorrat benutzen. Wichtig zum Verständnis ist nur, dass wir unseren Text in eine Folge von sehr, sehr großen Zahlen konvertiert haben (z. B. 50-stellige Zahlen). Es ist auch möglich, für die ganze Nachricht einen einzigen Block zu verwenden. In diesem Fall besteht unser ursprünglicher zu verschlüsselnder Nachrichtentext nur mehr aus einer einzigen (zugegebenermaßen riesigen) Zahl, die wir m (= ‚message‘) nennen wollen. Die Zahl enthält unseren gesamten Text. Bisher ist es sehr einfach aus m wieder unseren ursprünglichen Text herzustellen.

Die Verschlüsselung kommt jetzt ins Spiel. Zur Erinnerung: der Sender besitzt den öffentlichen Schlüssel, den er vom Empfänger bekommen hat. Den zum öffentlichen

⁶⁸ ASCII: ‚American Standard Code for Information Interchange‘

Schlüssel passenden geheimen privaten Schlüssel (der gleichzeitig mit dem öffentlichen Schlüssel beim Empfänger erzeugt wurde) besitzt nur der Empfänger.

Nun zu den Details. Das Erzeugen der privaten und öffentlichen Schlüssel geschieht wie folgt: Wir suchen uns zwei unterschiedliche, sehr große Primzahlen (typischerweise mit einigen Hundert Stellen Länge). Primzahlen dieser Größe sind mit einfachen mathematischen Methoden zu erzeugen, z. B. mit probabilistischen Primzahltests, dem Fermatschen Primzahltest, dem Miller-Selfridge-Rabin-Test, dem APRCL-Test oder dem Solovay-Strassen-Test (die Algorithmen sind verständlich auf Wikipedia dokumentiert). Primzahlen, die mit einem guten probabilistischen Verfahren erzeugt werden, werden allgemein als PRP-Zahlen bezeichnet. Sie sind nach menschlichem Ermessen ‚echte‘ Primzahlen, da die Fehlerwahrscheinlichkeit bei solch großen Zahlen astronomisch gering ist (typischerweise z. B. 10^{-100}). Es gibt zwar auch exakte Methoden (mit polynomieller Laufzeit), die jedoch für die Erzeugung sehr großer Primzahlen wegen ihrer langen Laufzeit nicht geeignet sind.

Hinweis: Der Rekord für die größte derzeit bekannte PRP-Zahl ist eine sogenannte Wagstaff-Primzahl (4.13) und lautet (Stand Mai 2016):

$$(2^{13372531} + 1)/3 \quad \text{Anzahl der Dezimalstellen: 4025533}$$

Zum Vergleich die größte ‚allgemeine‘ Primzahl, die mit einem Verfahren für beliebige Primzahlen (nicht Primzahlen einer speziellen Form) gefunden wurde (Stand 2011):

$$((((((2521008887^3 + 80)^3 + 12)^3 + 450)^3 + 894)^3 + 3636)^3 + 70756)^3 + 97220$$

Diese Zahl ist die 11-te Mills-Primzahl und besitzt 20562 Dezimalstellen⁶⁹.

Man erkennt deutlich, dass für Primzahlen einer speziellen Form Primzahltests für wesentlich größere Primzahlen verfügbar sind. Der Lucas-Lehmer Test für Mersenne Primzahlen liefert immer noch die größten Primzahlen (über 10 Millionen Stellen). Die größte derzeit allgemeine Primzahl mit 20562 Dezimalstellen wirkt sich dagegen recht bescheiden aus, da sie ca. 1000 mal weniger Dezimalstellen besitzt. Zurück zum RSA-Verfahren:

Die Verfahren zum Erzeugen großer, für das RSA-Verfahren geeigneter Primzahlen soll hier nicht beschrieben werden. Wir verlassen uns wieder einmal auf die Mathematica-Software, die eine Reihe von Funktionen zur Verfügung stellt, die in der Kryptographie benutzt werden:

```
PowerMod[], PowerModList[], PolynomialMod[], RandomPrime[],
Prime[], PrimeQ[], CoprimeQ[], FactorInteger[],
GenerateAsymmetricKeyPair[], Encrypt[], Decrypt[],
PrivateKey[], PublicKey[]
```

Mathematica erzeugt im Bruchteil einer Sekunde 200-stellige Primzahlen (und verwendet dabei PRP-Algorithmen):

⁶⁹ Paulo Ribenboim: Die Welt der Primzahlen (Springer), S.118

Als Nächstes brauchen wir den Entschlüsselungsexponent l . Der Entschlüsselungsexponent l ist das multiplikative inverse Element bezüglich $\varphi(n)$.

Somit ist $k \cdot l \equiv 1 \pmod{\varphi(n)}$. Zum Berechnen von l gibt es viele Verfahren, z. B. einen etwas modifizierten ‚Euklidischen‘ Algorithmus, oder mit Mathematica:

```
l=PowerMod[k, -1, n]
```

Die Primzahlen p und q sowie der Entschlüsselungsexponent l bilden den privaten, geheimen Schlüssel.

Eigentlich wird zum Entschlüsseln nur der (geheime) Entschlüsselungsexponent l und das (öffentliche) Modul n gebraucht, so dass man auch von den Schlüsselpaaren (k, n) (öffentlich) und (l, n) (privat) spricht. Ist der Verschlüsselungsalgorithmus einmal eingerichtet, werden die Werte $\varphi(n)$ sowie die Primzahlen p und q nicht mehr zur Entschlüsselung benötigt und können (zur Sicherheit) wieder gelöscht werden.

Nun zum eigentlichen Ver- und Entschlüsseln von Nachrichten:

Unsere Nachricht wird, wie oben beschrieben (ASCII-Codes) in eine Zahl m konvertiert.

Diese Zahl m sollte kleiner als unser Modul n sein: $m < n$.

Wenn diese Bedingung nicht gilt, dann muss die Nachricht in mehrere Blöcke m_i aufgeteilt werden, so dass wieder $m_i < n$ gilt.

Das Verschlüsseln geschieht einfach durch Berechnung von $r = m^k \pmod{n}$ mit Hilfe des Verschlüsselungsexponenten k . Die meisten Implementierungen benutzen für k den Wert 65537 (die 5-te Fermat-Zahl).

Zu kleine Werte für k verringern die Sicherheit des Verfahrens und machen die Verschlüsselung angreifbar. Dieser verschlüsselte Wert r wird an den Empfänger geschickt.

Das Entschlüsseln geschieht einfach durch Berechnung von $m = r^l \pmod{n}$ mit Hilfe des (inversen) Entschlüsselungsexponenten l .

Im Originalartikel von Rivest, Shamir und Adleman⁷⁰ wird die Zusatzbedingung angegeben, dass m und n zueinander teilerfremd sein müssen (dann gilt nämlich, dass $m^{\varphi(n)} \equiv 1 \pmod{n}$). Die RSA-Methode scheint jedoch auch zu funktionieren, wenn $\text{ggT}(m, n) \neq 1$, (d.h. wenn unsere Nachricht m und der RSA-Modul n einen Teiler gemeinsam haben). Die Annahme, dass m und n zueinander teilerfremd sind vereinfacht lediglich den Beweis der Gültigkeit des RSA-Verfahrens.

Der Beweis, dass dieser Algorithmus funktioniert, ist einfach:

$$r^l = (m^k)^l = m^{kl}, \quad k \cdot l \equiv 1 \pmod{\varphi(n)}$$

⁷⁰ „A method for Obtaining Digital Signatures and Public-Key Cryptosystems“, S. 7 (1978)

Die RSA-Verschlüsselung

daraus folgt, dass es eine ganze Zahl s gibt, so dass

$$k \cdot l = s \cdot \varphi(n) + 1$$

Ein paar Umformungen genügen, um zu zeigen, dass Verschlüsselung von m und anschließende Entschlüsselung wieder m ergibt:

$$\begin{aligned} r^l(\bmod n) &= m^{kl}(\bmod n) \\ &= m^{s \cdot \varphi(n) + 1}(\bmod n) = m(m^{\varphi(n)})^s(\bmod n) \\ &= m(\bmod n), \quad \text{da } m^{\varphi(n)} \equiv 1(\bmod n) \\ &= m \end{aligned}$$

Praktisch angewendete Verfahren verwenden zusätzlich unterschiedliche Padding-Methoden, die in den einschlägigen Spezifikationen beschrieben sind. Padding bedeutet, dass dem zu verschlüsselnden Text zusätzliche Informationen (evtl. auch Zufallselemente oder Angaben über die Länge des Textes) angehängt werden, um die Sicherheit des Verfahrens zu erhöhen. Gängige Padding-Methoden sind z.B. ‚PKCS#1‘, ‚OAEP‘ (Optimal Asymmetric Encryption Padding) oder ‚SSLV23‘. PKCS1 und SSLV23 fügen dem zu verschlüsselnden Datenblock 11 Zusatz-Bytes hinzu, bei dem ‚OAEP‘-Verfahren sind es sogar 41 Bytes. Auf weitere Details soll hier nicht eingegangen werden. Zum Nachlesen seien dem Leser Quellen aus dem Internet empfohlen.⁷¹ Die Software Mathematica unterstützt zum Verschlüsseln und Entschlüsseln das PKCS1-Padding-Verfahren. Man beachte, dass für die modifizierte Nachricht m' immer noch gelten muss: $m' < n$.

⁷¹ <https://de.wikipedia.org/wiki/RSA-Kryptosystem>,
<http://people.csail.mit.edu/rivest/Rsapaper.pdf>,
http://www.di-mgt.com.au/rsa_theory.html

14.2 DIE SICHERHEIT DER RSA-METHODE

In den praktischen Implementierungen der RSA-Verschlüsselungen sind noch einige ‚Zusatz-Features‘ eingebaut, die die Übertragung noch sicherer machen. Dennoch beruht die Sicherheit des Verfahrens darauf, dass der öffentliche Schlüssel (RSA-Modul) mit der Zahl n , die das Produkt zweier großer Primzahlen darstellt mit den momentan bekannten Faktorisierungsmethoden nicht in die beiden Primfaktoren zerlegt werden und somit auch $\varphi(n)$ und der Entschlüsselungsexponent l berechnet werden können.

Hier verbirgt sich wieder einmal Sprengstoff: Niemand (auch die Spezialisten auf diesem Gebiet) kann sagen, ob nicht irgendwann ein schneller Faktorisierungs-Algorithmus gefunden wird. Ein Faktorisierungsalgorithmus, der in polynomieller Zeit ablief, könnte die Sicherheit zum Einsturz bringen. Es ist z. Zt. (Stand Mai 2016) noch unbewiesen, ob es solche schnelle Algorithmen gibt. Dies ist bemerkenswert, denn in vielen anderen Fällen ist es Mathematikern möglich, zu beweisen, dass es für ganze Klassen von Problemen keine schnellen Algorithmen geben kann. Dieser Beweis steht für das Faktorisierungsproblem noch aus, d.h. es ist durchaus denkbar, dass solche Algorithmen existieren (10.3). Unser gesamtes Bankwesen würde buchstäblich zusammenbrechen, wenn Hacker in den Besitz solcher Algorithmen gelangen würden!

Hinweis: Eine schnelle Methode zur Berechnung von $\varphi(n)$ bzw. $\sigma(n)$ hätte die gleichen Folgen (20.9.3.2). Die bis heute (Stand Mai 2016) bekannten Berechnungsmethoden sind jedoch von der gleichen Komplexität wie das Faktorisierungsproblem.

Eine andere Gefahr droht von der immer schneller werdenden Computer-Hardware. Mit der momentan verwendeten Computer-Hardware ist es zwar auszuschließen, dass die Schlüssel mit den momentan verwendeten Schlüssellängen geknackt werden können (die Rechenzeiten hierfür liegen immer noch astronomisch hoch, bis zum Alter unseres Universums...). Sollte es jedoch eines Tages die vielgepriesenen Quanten-Computer tatsächlich geben (an denen im Moment eifrig geforscht wird), dann wäre auch die RSA-Verschlüsselung in Gefahr. Dem Autor ist nicht bekannt, ob es Untersuchungen darüber gibt, ob durch Vergrößerung der Verschlüsselungstiefe beim RSA-Verfahren die Gefahren einer Entschlüsselung durch Quantencomputer gebannt werden könnte...

14.3 RECHENBEISPIELE ZUR RSA-VER- UND ENTSCHLÜSSELUNG

Hier ein paar einfache Rechenbeispiele. An den Farben erkennt man die verschiedenen Bereiche (öffentlich in **blau**, geheim beim Empfänger in **rot**, geheim bei Empfänger und Sender in **grün**):

Beispiel 1:

Der Sender möchte eine geheime Nachricht an den Empfänger senden, die nur aus der Zahl 1115 besteht:

mSource = 1115

Der Empfänger wählt zwei verschiedene geeignete Primzahlen:

p = 47 und **q = 59**

Daraus ergibt sich folgender Modul, der dem Sender (unverschlüsselt) mitgeteilt wird:

pubModulus = pq = 2773

Die Anzahl zu pubModulus teilerfremder Zahlen ist:

privModulus = $\varphi(\text{pubModulus}) = (p - 1) * (q - 1) = 2668$

Der Empfänger bestimmt einen geeigneten Verschlüsselungsexponenten und schickt ihn (unverschlüsselt) an den Sender:

pubExponent = 17

Der Empfänger berechnet auch den multiplikativ inverse Entschlüsselungsexponent mit Hilfe von φ :

privExponent = PowerMod[pubExponent, -1, privModulus] = 157

Der Sender verschlüsselt mSource zu mCrypt und schickt mCrypt an den Empfänger:

mCrypt = mSource^{pubExponent} (mod pubModulus) = 1379

Der Empfänger entschlüsselt mCrypt wieder zu mSource:

mSource = mCrypt^{privExponent} (mod pubModulus) = 1115

Man kann den Entschlüsselungsexponent ‚privExponent‘ leicht knacken, indem man die Primfaktorenzerlegung des Moduls mit den Faktoren p und q berechnet und daraus über privModulus wieder über privExponent verfügt.

Mathematica:

```
(*very simple example for the RSA encryption*)
(##### implement coding mechanism #####)
(*choose two different prime numbers:*)
p=47; q=59;
If[!PrimeQ[p]||!PrimeQ[q],Print["Error: p or q not Prime!"];Exit[;]
(*Compute public module and phi(*)
pubModulus=p*q; privModulus=(p-1)*(q-1); (*=EulerPhi[pubModulus]*)
(*pubExponent Can be chosen freely, must be between 3 and privModulus
and coprime to privModulus*)
pubExponent=17;
If[pubExponent >= privModulus||pubExponent<3,Print["Error: pubExponent
> privModulus!"];Exit[;]
If[!CoprimeQ[pubExponent,privModulus],Print["Error: pubExponent not
coprime to privModulus!"];Exit[;]
(*compute private exponent: inverse of public exponent*)
privExponent=PowerMod[pubExponent,-1,privModulus];
(##### Encode and Deccode messages:#####)
(*this is our message to be encoded:*)
mSource=1115
```

```

If[mSource >=pubModulus,Print["Error: message bigger than module (use
bigger primes)!"];Exit[;]
(*encode: *)
mCrypt=PowerMod[mSource,pubExponent,pubModulus]
(*decode:*)
mSource1=PowerMod[mCrypt,privExponent,pubModulus]
If[mSource!=mSource1,Print["Error: RSA Coding/Encoding failed"]];
(*##### Hacking the module:#####*)
Print["Try to hack RSA module..."];
pqHacked=FactorInteger[pubModulus];
pH=First[pqHacked[[1]];qH=First[pqHacked[[2]]];
privExponentH=PowerMod[pubExponent,-1,(pH-1)*(qH-1)]
If[privExponentH==privExponent,Print["Hack of RSA module
succeeded!"]];

```

Beispiel 2 (wie Beispiel 1):

Der Sender möchte eine geheime Nachricht an den Empfänger senden, die nur aus der Zahl 1115 besteht. In diesem Beispiel verwenden wir die in Mathematica implementierten Funktionen `PublicKey[]`, `PrivateKey[]`, `Encrypt[]`, sowie `Decrypt[]`. Es wurde kein Padding-Verfahren benutzt („None“).

Das verwendete Programm kann im Anhang nachgelesen werden (20.11.11).

Die von Mathematica erzeugten privaten und öffentlichen Schlüssel ergeben sich zu:

PrivateKey [ cipher: RSA
private exponent length: 8 bits
public modulus length: 12 bits
padding: None
public exponent: 17
private exponent: 157
public modulus: 2773]

PublicKey [ cipher: RSA
public modulus length: 12 bits
padding: None
public exponent: 17
public modulus: 2773]

Die originale Zahl lautet: **1115**

Die zur Erzeugung der Schlüssel verwendeten Primzahlen und Modul lauten:

$p = 47$; $q = 59$; public Modulus=2773

Die Verschlüsselte Zahl lautet: **1379**

Beispiel 3:

Der Sender möchte eine geheime Nachricht an den Empfänger senden, die nur aus dem Text „OK“ besteht. In diesem Beispiel verwenden wir die in Mathematica implementierten Funktionen `GenerateAsymmetricKeyPair[]`, `Encrypt[]`, sowie `Decrypt[]` in Verbindung mit dem „PKCS1“-Padding Verfahren bei einer Schlüssel-länge von 97 Bit.

Das verwendete Programm kann im Anhang nachgelesen werden (20.11.11).

Die von Mathematica erzeugten privaten und öffentlichen Schlüssel ergeben sich zu:

```

<|"PrivateKey" -> PrivateKey [
  cipher: RSA
  public modulus length: 97 bits
  private exponent length: 93 bits
  padding: PKCS1
  public exponent: 65537
  public modulus: 12202 433 704 <<7>> 277 596 949 541
  private exponent: 76245 427 803 <<5>> 502629 985 493
], "

PublicKey" -> PublicKey [
  cipher: RSA
  public modulus length: 97 bits
  padding: PKCS1
  public exponent: 65537
  public modulus: 12202 433 704 <<7>> 277 596 949 541
] ]>
    
```

Hier im Einzelnen die genauen Werte (Ausgabe des Programmes in Englisch):

Original string to be encoded: „OK“
 Original string as number: **20299**
 Original text as Byte array including 10 Bytes padded by PKCS1 algorithm:
 {2,32,69,224,233,133,242,219,235,0,**79,75**} „\02 Eàé...òÛä\00**OK**“
 Public Modulus: **122024337043892852277596949541**
 Private Exponent: **7624542780333828502629985493**
 Private Modulus (Phi[publicModulus]): **122024337043892092448561992492**
 Encrypt. object (data) {**0,42,120,153,109,62,0,217,150,54,211,165,4**}
 Encrypt. Number:**13144166048085041547004060932**
 Decryption-result (using encrypted Byte data as parameter restores original Bytes:
{79,75})
 Decryption-result (using encrypted Object as parameter restores original String):
 „OK“ (20299)

Beispiel 4:

Ein neugieriger Hacker ist im Besitz einer verschlüsselten Nachricht, sowie des dazugehörigen öffentlichen Schlüssels und möchte gerne die Nachricht entschlüsseln, obwohl er den privaten Schlüssel (mit dem privaten Exponenten) nicht besitzt. In diesem Beispiel verwenden wir die in Mathematica implementierten Funktionen PrivateKey[], sowie Decrypt[] in Verbindung mit dem „PKCS1“-Padding Verfahren bei einer Schlüssellänge von 192 Bit.

Das verwendete Programm kann im Anhang nachgelesen werden (20.11.11). Nehmen wir an, irgendjemand hat folgenden öffentlichen RSA-Schlüssel generiert (z. B. indem er die Mathematica Funktion GenerateAsymmetricKeyPair[] benutzte. Nehmen wir weiter an, er hat den Modul, den öffentlichen Verschlüsselungsexponent, sowie eine verschlüsselte Nachricht veröffentlicht:
pubExponent = 65537;
pubModulus = 5369695965139088101081485235567478142438728289315726900871;
mCrypt = 1917971481256834478883961041086543933343882914074934636133;

Unser Hacker braucht nur eine Handvoll Mathematica Programm-Code um den Schlüssel zu hacken:

Er faktorisiert den Modul in die Primzahlen pH und qH:

```
FactorInteger[5369695965139088101081485235567478142438728289315726900871]
{{68357071940820194611682396513, 1}, {78553627484042565312533006567, 1}}
```

Er berechnet Phi[] (auch der ‚private Modul‘ genannt):

privModulus = (pH - 1) * (qH - 1):

```
5369695965139088101081485235420567443013865529391511497792
```

Er berechnet den privaten Exponenten:

privExponent = PowerMod[pubExponent, -1, privModulus];

```
484499185966049249555967871982611572207133532958607342401
```

Und generiert damit schließlich einen neuen privaten Schlüssel, mit dem er die Nachricht entschlüsseln kann:

```
privKey=PrivateKey[<|"Cipher"->"RSA", "Padding"-
>"PKCS1", "PublicExponent"->pubExponent, "PrivateExponent"-
>privExponent, "PublicModulus"->pubModulus|>]
```

Schließlich kann er die Nachricht entschlüsseln:

```
bCryptArray=ByteArray[IntegerDigits[mCrypt, 256]];
decryptedByteArray=Normal[Decrypt[privKey, bCryptArray]];
decryptedString=FromCharacterCode[decryptedByteArray]
```

und voilà die entschlüsselte Nachricht:

„Elvis lives!“

Das verwendete Programm kann im Anhang nachgelesen werden (20.11.11).

Hinweis: Die Rechenzeit beträgt auf einem 2.6 GHZ Quadcore Rechner ca. 30 Sekunden.

Beispiel 5:

Ein weiteres Beispiel, wie ein privater Schlüssel mit einer Schlüssellänge von 2048 Bit gehackt werden kann und eine derart verschlüsselte Nachricht (ca. 256 Bytes) ohne Kenntnis des privaten Schlüssels gelesen werden kann, findet sich ebenfalls im Anhang (20.11.11).

15 PRIMZAHLEN IN DER MUSIK

15.1 EULERS KONSONANZTHEORIE UND DER GRADUS SUAVITATIS

Jedem Musiker ist bekannt, dass musikalische Intervalle und Akkorde konsonant oder dissonant klingen können, wobei die Grenze zwischen ‚konsonant‘ und ‚dissonant‘ fließend ist und wohl auch etwas vom Zeitgeschmack der jeweiligen historischen Epoche der Musik abhängt. Dennoch war der Mathematiker Leonhard Euler (1707-1783) davon überzeugt, eine mathematische Definition des Wohlklanges (genauer: des ‚Grades‘ von Wohlklang) geben zu können. Der aufmerksame Leser wird bemerken, dass uns Euler in dieser Arbeit schon mehrfach begegnet ist...

Er fand eine Formel, die den Grad des Wohlklanges als eine natürliche Zahl angibt und nannte diese sich aus der Formel ergebende Zahl den ‚Gradus Suavitatis‘.⁷² Bei der Berechnung des Gradus Suavitatis spielen (wie sollte es anders sein) Primzahlen eine besondere Rolle. Euler verwendet den Begriff der Konsonanz für beliebige (nicht nur wohlklingende) Zusammenklänge. Unter Zusammenklang verstehen wir gleichzeitig erklingende Töne, wobei die Töne ‚rein‘ gestimmt sein sollen und somit untereinander rationale (mit natürlichen Zahlen beschreibbare) Verhältnisse haben. Obwohl der Gradus Suavitatis für beliebige Zahlenverhältnisse berechnet werden kann, kamen in der Musik für ihn anfangs nur Intervalle in Frage, deren Verhältnisse mit den Primzahlen 2, 3 und 5 beschrieben werden können. In seinen späteren Schriften plädiert er jedoch für die Einführung der Primzahl 7 in die ‚musikalische Arithmetik‘.

⁷² Euler, Leonhard:

Tentamen Novae Theoriae Musicae Ex Certissimis Harmoniae Principiis Dilucide Expositae
Petropolis 1739

CAPVT QVARTVM DE CONSONANTIIS!

§. I.

PLures soni simplices simul sonantes constituunt sonum compositum, quem hic consonantiam appellabimus. Ab aliis quidem consonantiae vox strictiore sensu accipitur, ut tantum denotet sonum compositum auditui gratum multumque suauitatis in se habentem: hancque consonantiam distinguunt a dissonantia, quae ipsis est sonus compositus parum vel nihil suauitatis complectens. At quia partim difficile est consonantiarum et dissonantiarum limites definire, partim vero haec distinctio cum nostro tractandi

Abbildung 128: Anfang des 4. Kapitels aus Leonhard Eulers Arbeit „Tentamen Novae Theoriae musicae“

Anders als bei der pythagoreischen Stimmung, bei der man alle vorkommenden Skalentöne mit Hilfe von Quinten (die bei Bedarf durch Oktavierung in den eine Oktave umfassenden Oktavraum projiziert werden) konstruiert, verwendet man bei der reinen Stimmung für die kleinen und großen Terzen die ‚reinen‘ Zahlenverhältnisse $\frac{5}{4}$ bzw. $\frac{6}{5}$. Möchte man allen 12 Halbtönen der Oktave einen ‚reinen‘ Zahlenverhältnis zuordnen, so haben wir für die Zwischentöne eine gewisse Mehrdeutigkeit, denn eigentlich sind nur die Oktave, Quarte, Quinte, große/kleine Terz sowie die große/kleine Sexte genau festgelegt. Nehmen wir als Grundton C, dann lauten die 7 ‚festen‘ Töne:

C, Eb, E, F, G, Ab, A (mit den Verhältnissen $1, \frac{6}{5}, \frac{5}{4}, \frac{4}{3}, \frac{3}{2}, \frac{8}{5}, \frac{5}{3}$).

Die restlichen Töne können unterschiedlich gewählt werden, je nachdem von welchem Ton aus sie konstruiert werden. Hier ein Beispiel: Der Ton Bb kann vom F aus konstruiert werden: Die zwei hintereinander gesetzten Quartan vom C aus ergeben dann den Wert $\frac{4}{3} \cdot \frac{4}{3} = \frac{16}{9}$. Bb kann jedoch auch vom G aus konstruiert werden. Quint und kl. Terz vom C aus hintereinander gesetzt ergeben dann $\frac{3}{2} \cdot \frac{6}{5} = \frac{9}{5}$.

Die am meisten verwendete reine Skala, die der temperierten Tonskala am nächsten kommt, lautet:

Chromat. Tonleiter	C	Des	D	Es	E	F	Fis	G	As	A	B	H	c
Frequenzverhältnis	1/1	16/15	9/8	6/5	5/4	4/3	45/32	3/2	8/5	5/3	9/5 (16/9)	15/8	2/1

Aus dieser Mehrdeutigkeit erkennen wir, dass die Anwendung des Gradus Suavitatis auf unsere heutzutage übliche temperierte Stimmung (bei der die Frequenzverhältnisse der chromatischen Tonleiter einfach durch eine geometrische Folge mit dem Faktor $q = \sqrt[12]{2}$ entstehen) problematisch ist.

Definieren wir ein Intervall der reinen Stimmung als Frequenzverhältnis $\frac{p}{q}$, $p, q \in \mathbb{N}$ (s. Tabelle) und bezeichnen den sogenannten Euler-Exponent $E = \text{kgV}(p, q)$, bzw. in Englisch: $E = \text{lcm}(p, q)$, dann berechnet sich der **Gradus Suavitatis G** wie folgt:

$$G(E) = 1 + \sum_{i=1}^{\omega(E)} e_i (p_{n_i} - 1), \quad \text{wobei } E = \text{kgV}(p, q) = \prod_{i=1}^{\omega(E)} p_{n_i}^{e_i} \quad (158)$$

Dabei ist $\text{kgV}(p, q)$ bzw. $\text{lcm}(p, q)$ das kleinste gemeinsame Vielfache von p und q (least common multiple), $p_{n_i}^{e_i}$ sind die vorkommenden Primfaktoren mit ihrer Multiplizität, n_i die Indices der vorkommenden Primzahlen, $\omega(E)$ bezeichnet die Anzahl der unterschiedlichen Primzahlen. Manchmal wird der Gradus Suavitatis in der Literatur auch mit dem Symbol $\Gamma(p, q)$ bezeichnet. Wir verwenden hier lieber das Symbol G um Verwechslungen mit der Gamma-Funktion $\Gamma(x)$ zu vermeiden. Der Gradus Suavitatis kann auch problemlos auf Zusammenklänge von mehr als zwei Tönen angewendet werden. In diesem Fall berechnet sich E einfach zu $E = \text{kgV}(q_1, q_2, q_3, \dots)$ wobei q_1, q_2, q_3, \dots die zum tiefsten Ton vorkommenden (auf ganze Zahlen erweiterte) Frequenzverhältnisse darstellen.

Der Gradus Suavitatis kann somit auch für Zusammenklänge von beliebig vielen Tönen berechnet werden. Bei einem einzigen Argument n gilt: $E = n$. In diesem Fall ist der Gradus Suavitatis lediglich eine zahlentheoretische Funktion, die nur von der positiven ganzzahligen Variable n abhängt. Außerdem definieren wir: $G(1) = 1$. Hinweis: Wir nehmen außerdem an, dass die Verhältniszahlen (q_1, q_2, q_3, \dots) minimal, d.h. ‚gekürzt‘ sind, da sonst der Gradus Suavitatis höhere Werte ergeben würde, somit ist die Schreibweise $G\left(\frac{3}{2}\right)$ etwas ‚ungenau‘. Da wir in diesem Fall den Gradus Suavitatis für einen Zusammenklang aus 2 Tönen (Intervall) mit den Frequenzverhältnissen 1 (Grundton) und $\frac{3}{2}$ (Quinte) berechnen möchten, sollte man mathematisch genauer schreiben:

$G\left(1, \frac{3}{2}\right) = G(2, 3)$. Um Verwechslungen zu vermeiden sollte man am besten für den zu untersuchenden Zusammenklang zuerst den Euler-Exponenten E berechnen und dann den Gradus Suavitatis für das ganzzahlige Argument E .

Hier ein paar Beispiele:

$$G(2) = 1 + 1 \cdot (2 - 1) = 2, \quad G(3) = 1 + 1 \cdot (3 - 1) = 3, \quad G(4) = 1 + 2 \cdot (2 - 1) = 3$$

$$G\left(\frac{3}{2}\right) = G(3, 2) = G(6) = 1 + 1 \cdot (2 - 1) + 1 \cdot (3 - 1) = 4 \text{ (Quinte)}$$

$$G\left(\frac{2}{3}\right) = G(2, 3) = G(6) = 4 \text{ (Quarte)}$$

$$G\left(\frac{12}{5}\right) = G(12,5) = G(60) = 1 + 2 \cdot 1 + 1 \cdot 2 + 1 \cdot 4 = 9 \text{ (kl. Dezime)}$$

$$\text{Dur-Dreiklang: } q_1 = 1, q_2 = \frac{5}{4}, q_3 = \frac{3}{2}, \Rightarrow E = \text{kgV}(4,5,6) = 60, G(60) = 9$$

$$1. \text{ Umkehr.: } q_1 = 1, q_2 = \frac{6}{5}, q_3 = \frac{8}{5}, \Rightarrow E = \text{kgV}(5,6,8) = 120, G(120) = 10$$

$$2. \text{ Umkehr.: } q_1 = 1, q_2 = \frac{4}{3}, q_3 = \frac{5}{3}, \Rightarrow E = \text{kgV}(3,4,5) = 60, G(60) = 9$$

$$\text{Moll-Dreiklang: } q_1 = 1, q_2 = \frac{6}{5}, q_3 = \frac{3}{2}, \Rightarrow E = \text{kgV}(10,12,15) = 60, G(60) = 9$$

$$1. \text{ Umkehr.: } q_1 = 1, q_2 = \frac{5}{4}, q_3 = \frac{5}{3}, \Rightarrow E = \text{kgV}(12,15,20) = 60, G(60) = 9$$

$$2. \text{ Umkehr.: } q_1 = 1, q_2 = \frac{4}{3}, q_3 = \frac{8}{5}, \Rightarrow E = \text{kgV}(15,20,24) = 120, G(120) = 10$$

Der Gradus Suavitatis ist umso größer, je größer die Dissonanz der betrachteten Intervalle ist. Kleinere Gradus-Suavitatis-Werte bedeuten eine höhere Konsonanz (mehr ‚Wohlklang‘).

Eine eindeutige Zuordnung aus dem Gradus Suavitatis in die Kategorien ‚Konsonant‘ und ‚Dissonant‘ ist jedoch problematisch, im Originaltext heißt es:

§. 14. Tam monui, me hic sub consonantiae nomine tam consonantias, quam dissonantias vulgo sic dictas complecti. Ex tabula autem apposita et methodo nostra limites quodammodo definiri posse videntur. Dissonantiae enim ad altiores pertinent gradus, et pro consonantiis habentur, quae ad inferiores gradus pertinent. Ita tonus, qui constat sonis rationem 8 : 9 habentibus, et ad octavum gradum est relatus, dissonantiis annumeratur, ditonus vero seu tertia maior ratione 4 : 5 contentus, qui ad septimum gradum pertinet, consonantiis. Neque tamen ex his octavus gradus initium potest constitui dissonantiarum; nam in eodem continentur rationes 5 : 6, et 5 : 8, quae dissonantiis non accensentur.

Abbildung 129: 4. Kapitel, §14 aus Leonhard Eulers Arbeit „Tentamen Novae Theoriae musicae“

Übersetzung lt. Mitzler:

„...Die Dissonanzen gehören zu höheren Graden, und für Konsonanzen werden diejenigen gehalten, die zu tieferen Graden gehören. So wird der Ganzton, der aus Tönen im Verhältnis 8 : 9 besteht und zum achten Grad gehört, zu den Dissonanzen gezählt, der Ditonus aber (die große Terz), der im Verhältnis 4 : 5 enthalten ist, welcher zum siebten Grad gehört, wird zu den Konsonanzen gezählt. Und trotzdem kann aus diesem achten Grad nicht der Anfang der Dissonanzen festgelegt werden; denn in demselben sind die Verhältnisse 5 : 6 und 5 : 8 enthalten, welche nicht zu den Dissonanzen gerechnet werden.“

Hier ein paar Mathematica-Programme zur Berechnung des Gradus-Suavitatis:

```
(*Computing Eulers Gradus Suavitatis *)
(*works for any number of arguments n >1, arguments can be rational:*)
eulerExp[q_]:=Module[{exp,qList},
qList=List[q];
If[Length[qList]==1&&IntegerQ[qList[[1]]],
exp=qList[[1]], exp=Apply[LCM,qList]/Apply[GCD,qList]];
Return[exp];
];
(*works for 1 argument. argument can be rational:*)
eulerExp[r_]:=eulerExp[Numerator[r],Denominator[r]];
(*works for 1 integer argument, numbertheoretic version:*)
gs[n_]:=Module[{s=FactorInteger[n]},1+Sum[s[[k,2]]*(s[[k,1]]-1),{k,Length[s]}]];
(*works for 1 or 2 arguments arguments can be rational:*)
gs2[p_,q_]:=gs[LCM[p,q]/GCD[p,q]];
gs2[x_]:=gs2[Numerator[x],Denominator[x]];
(*works for any number of arguments n > 1, arguments can be rational:*)
gsn[q_]:=Module[{exp,retValue},
exp=eulerExp[q];
s=FactorInteger[exp];
retValue=1+Sum[s[[k,2]]*(s[[k,1]]-1),{k,Length[s]}];
Return[retValue];
];
```

15.1.1 MATHEMATISCHE EIGENSCHAFTEN DES GRADUS SUAVITATIS

- 1) $G(1) = 1$
- 2) $G(p) = p$, wenn $p \in \mathbb{P}$
- 3) $G(pq) = G(p) + G(q) - 1$ (quasi-logarithmisch')
- 4) $G(2^n) = n + 1$
- 5) $G(p^n) = (p - 1)n + 1$
- 6) $G\left(\frac{p}{q}\right) = G\left(\frac{q}{p}\right)$ oder $G(p, q) = G(q, p)$
- 7) $G(q_1, q_2, q_3, \dots, q_n) = G(q_{i_1}, q_{i_2}, q_{i_3}, \dots, q_{i_n})$, wobei die $i_1, i_2, i_3, \dots, i_n$ alle möglichen Permutationen von $1, 2, 3, \dots, n$ durchlaufen
- 8) $G\left(\frac{1}{q_1}, \frac{1}{q_2}, \frac{1}{q_3}, \dots, \frac{1}{q_n}\right) = G(q_1, q_2, q_3, \dots, q_n)$

Für Dur- und Molldreiklänge gilt (seien X die 3 Frequenzverhältnisse eines Dur-Akkordes, Xm die drei Verhältnisse eines Moll-Akkordes; die Indices 1 und 2 bezeichnen jeweils die erste bzw. 2.te Umkehrung):

- 9) $G(X) = G(X_2) = G(Xm) = G(Xm_1)$
- 10) $G(X_1) = G(Xm_2)$

Nach Euler kann eine Folge von Frequenzverhältnissen (Akkord) so weiter geführt werden, dass der Gradus-Wert gleich bleibt. Euler nennt dies die vollständige Konsonanz.

Für den Dur-Dreiklang ergibt die vollständige Konsonanz folgende Frequenzverhältnisse:

$$1: 2: 3: \mathbf{4: 5: 6: 10: 12: 15: 20: 30: 60 \dots}$$

Man beachte, dass in dieser vollständigen Konsonanz Dur-Akkorde ($\mathbf{4: 5: 6}$, entspricht z. B. C-E-G) und Moll-Akkorde ($\mathbf{10: 12: 15}$ entspricht z. B. E-G-H) vorkommen. Hinweis: bis zum sechsten Glied ist diese Folge identisch mit der Obertonreihe.

Diese entspricht durchaus den modernen sich in der Jazz-Harmonik widerspiegelnden Gegebenheiten, nämlich, dass im Dur-Akkord die große Septime enthalten ist.

Hier ein Plot des Gradus Suavitatis, der große Ähnlichkeit mit ABBILDUNG 43 (Integer-Logarithmus) zeigt:

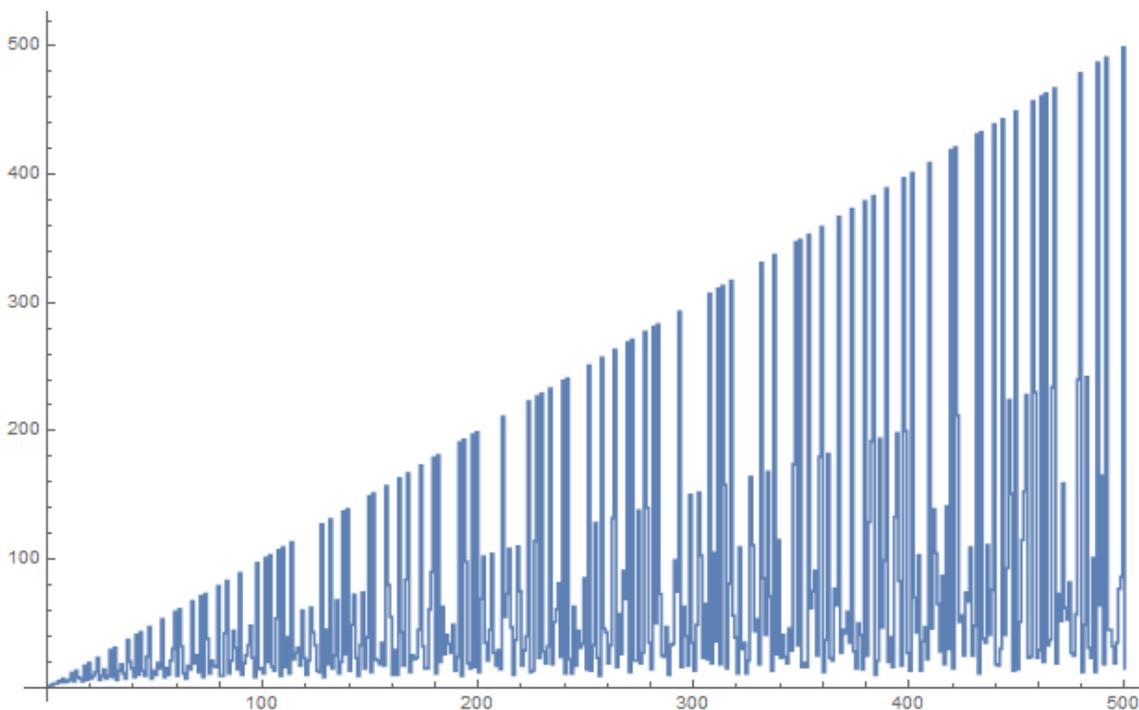


Abbildung 130: Leonhard Eulers Gradus Suavitatis im Bereich 1 bis 500

15.1.2 DAS ‚ZURECHTHÖREN‘ VON KOMPLEXEN ODER IRRATIONALEN INTERVALLEN

Die Methode des ‚Gradus Suavitatis‘ ist natürlich nur eine grobe Näherung und entspricht dem tatsächlichen Hörempfinden (konsonant oder dissonant) nur bedingt, und zwar nur für Proportionen, in denen kleine Zahlenwerte (und dazu nur solche, in deren Zerlegung die Primzahlen 2,3 und 5) vorkommen.

Dies ergibt sich schon allein aus der Tatsache, dass unhörbare kleine Verstimmungen in Intervallen bei der Berechnung des GS (die Berechnung verlangt natürlich eine Annäherung durch rationale Zahlenwerte) zu riesig großen Gradus-Suavitatis-Werten führen würden, obwohl sie immer noch als angenehm empfunden würden. Als Beispiel nehmen wir einen A-Dur Akkord mit 440 Hz bei A:

Die Frequenzen des A-Dur Akkordes (A-C#-E) stehen im Verhältnis 440:550:660. Der Gradus Suavitatis für diesen Dur-Akkord beträgt (wie oben schon beschrieben):

$G(440,550,660) = G(4,5,6) = 9$. Betrachten wir nun den um 1 Hz unhörbar verstimmt Ton E mit 661 Hz statt 660 Hz. Für das menschliche Ohr ist keine Änderung bemerkbar, der Gradus Suavitatis des minimal verstimmt Akkordes ist jedoch $G(440,550,661) = 682!$

Euler ist deshalb der Meinung, dass die menschliche Seele solche kleinen Verstimmungen ‚zurechthören‘ kann, d.h. eine temperierte Quinte mit einem irrationalen Frequenzverhältnis von $(\sqrt[12]{2})^7$ wird von unserer Seele einfach auf den ungefähr gleichen Wert $\frac{3}{2}$ ‚zurechtgehört‘. Welch ein Glück, dass $(\sqrt[12]{2})^7 = 1.49831$ so nahe am Wert der reinen Quinte ($\frac{3}{2} = 1.5$) liegt! Wäre dem nicht so, könnten wir in der temperierten 12-tönigen Skala keine (zumindest abendländische) Musik betreiben! Was wiederum die Frage aufwirft, ob wir es wirklich einem Zufall zu verdanken haben, dass die temperierte Quinte so nah an der reinen Quinte liegt. Manchen Menschen ist der Gedanke etwas unbehaglich, dass wir es einem Zufall zu verdanken haben, dass es so unglaublich schöne Musik wie die Werke von J. S. Bach gibt.

15.2 PRIMZAHLEN ALS RHYTHMISCHE MUSTER

Fasst man die Differenzen zwischen den Primzahlen als zeitliche Abstände auf, so bekommen wir ein rhythmisches Muster.

Wir benutzen das Sieb des Eratosthenes, ‚sieben‘ die ersten 50 Primzahlen (2 bis 229) und interpretieren die X-Achse als Zeit-Achse. Die Y-Achse interpretieren wir als Tonhöhe. Um in einen ‚hörbaren‘ Bereich zu kommen, multiplizieren wir die jeweilige Primzahl mit dem Frequenzfaktor 110 Hz, so dass die tiefsten Töne (2-er Rhythmus) bei 220 Hz liegen. Dies entspricht dem A3 (internationale Bezeichnung) oder a (kleines A, Bezeichnung in deutschsprachigen Ländern). Die höchsten Töne in dieser Darstellung liegen dann bei ca. 20000 Hz. Somit ergibt sich folgendes Diagramm, wobei jeder ‚Primzahl-Rhythmus‘ durch eine andere Farbe markiert wurde:

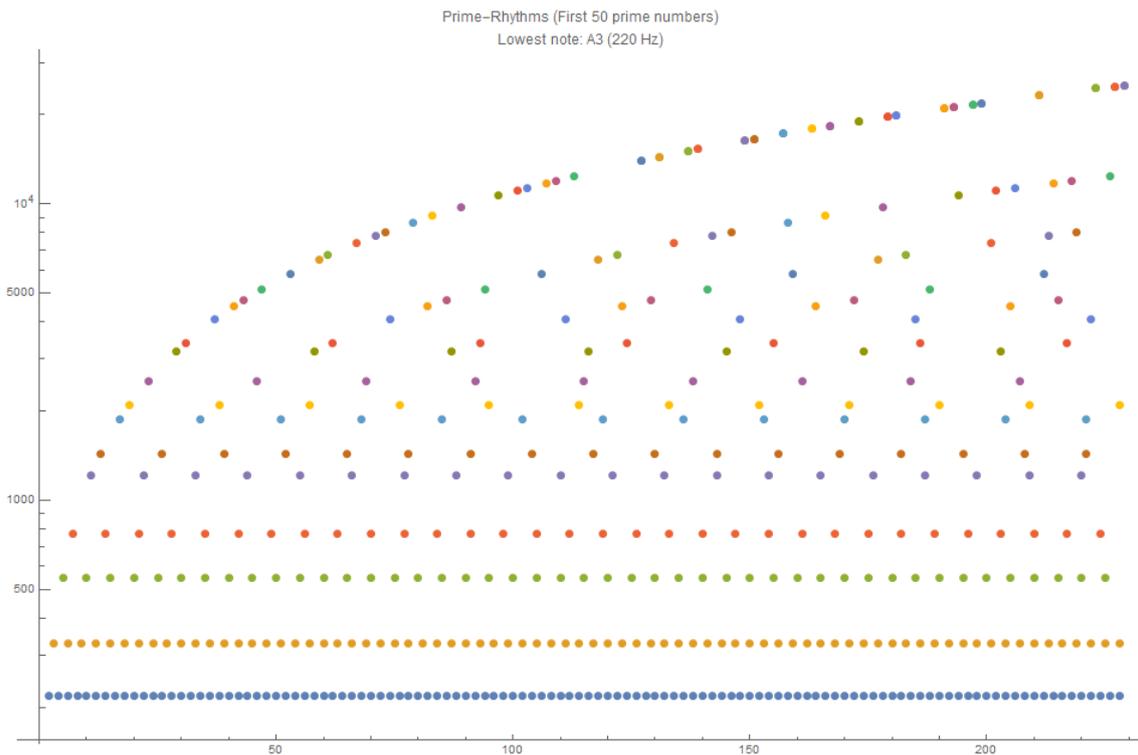


Abbildung 131: Primzahl-Rhythmen, mit den ersten 50 Primzahlen und A3 als tiefstem Ton

Mathematica:

```
(*Generate a list with 50 different Sequences of
(frequency,Primenumber)-Pairs, using A2=110Hz as the base-frequency*)
tab=Table[{j,Table[{i,110*Prime[j]},{i,Prime[j],Prime[50],Prime[j]}]},
{j,1,50}];
ListLogPlot[Table[tab[[k]][[2]],{k,1,50}],PlotLabel->"Prime-Rhythms
(First 50 prime numbers)\nLowest note: A3 (220 Hz)", ImageSize->Large]
```

Mit Mathematica ist es sehr einfach, diese Tabellen zu Musik werden zu lassen. Wir verwenden die Tabelle von ABBILDUNG 131 und erzeugen einen „Primzahl-Song“ von 46 Sek. Dauer:

Mathematica:

```
(*Benötigt die Prime-Sound-Library, zu finden im Anhang*)
tab=Table[{j,Table[{i,110*Prime[j]},{i,Prime[j],Prime[50],Prime[j]}]},
{j,1,50}];
sortedTab=Sort[Flatten[Drop[tab,None,1],2]];
noteList=createNoteListFromSortedTable[sortedTab];
song1=Sound[{"Percussion",Table[SoundNote[noteList[[k]][[2]]-
10,0.2],{k,1,Length[noteList]}]},{0,46}];
song2=Sound[{"Marimba",Table[SoundNote[noteList[[k]][[2]],0.2],{k,1,Le
ngth[noteList]}]},{0,46}];
primenumberSong=Sound[{song1,song2}];
Export["C:/Primzahlen/Sounds/primenumberSong46Sec.mid",primenumberSong
];
```

Primzahlen als rhythmische Muster

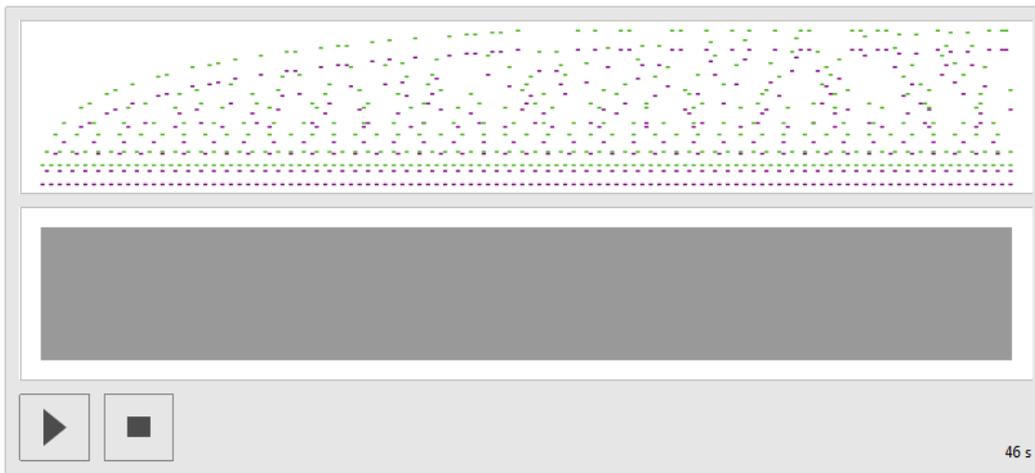


Abbildung 132: Mathematica-Soundobjekt (Primzahl-Song)

Die Idee, Primzahlen als rhythmische Muster zu interpretieren stammt von Peter Neubäcker, Chef der Firma Celemony und Erfinder der Musiksoftware ‚Melodyne‘⁷³.

Mit Melodyne ist es ebenfalls sehr einfach möglich, ‚Songs‘ mit Primzahlen zu erstellen. Melodyne hat gegenüber Mathematica den Vorteil, dass das Ergebnis als echter Sound (und nicht nur im MIDI-Format) exportiert werden kann. Außerdem kann die Tonhöhe der erzeugten ‚Note-Events‘ viel genauer festgelegt werden.

Hier ein Screen-Shot von Melodyne mit einem Primzahl-Arrangement:

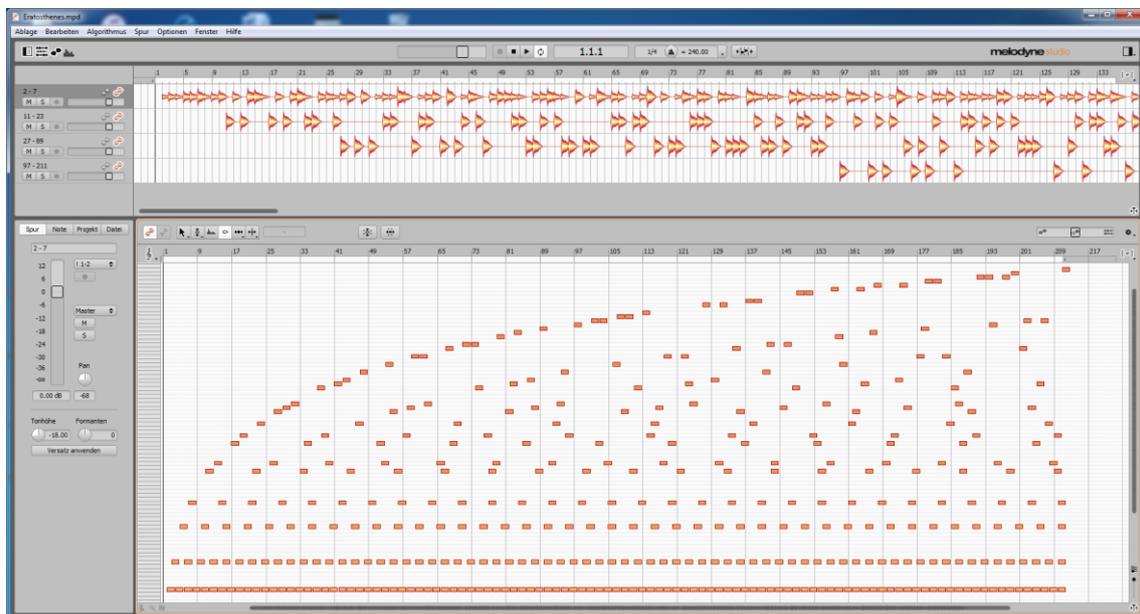


Abbildung 133: Melodyne erzeugt Primzahlrhythmen mit dem Sieb des Eratosthenes

⁷³ <http://www.celemony.com>

16 PRIMZAHLEN IN DER POESIE

16.1 HAIKUS UND TANKAS



荒海や
佐渡によこたふ
天河

**ura umi ya
sado ni yokotau
ama no gawa**

Tosende See.
nur die Milchstraße reicht
zur Insel Sado hinüber.

Turbulent the sea—
across to Sado stretches
the Milky Way



古池や
蛙飛び込む
水の音

**furu ike ya
kawazu tobikomu
mizu no oto**

Der alte Weiher:
Ein Frosch springt hinein.
Oh! Das Geräusch des Wassers

Ah! The ancient pond
As a frog takes the plunge
Sound of the water



Matsuo Bashô (松尾芭蕉), 1644-1695

道のべに清水流るる柳影
しばしとてこそ立ちどまりつれ

**Michi no be ni
Shimizu nagaruru
Yanagikage
Shibashi tote koso
Tachidomaritsure**

Wo am Wegerand ein Bach
Fließt mit glasklarem Wasser
Und eine Weide steht,
Da würde ich gerne noch bleiben:
„Ach, nur ein Weilchen“

Along the road
A pure stream flows
In the shade of a willow
Wanting to rest
I paused – and have not left

Saigyō (1118-1190)

Was haben diese drei japanischen Gedichte aus dem 12. bzw. 17. Jahrhundert mit Primzahlen zu tun (die ersten beiden sind Haikus, das dritte ein Tanka)?

Es ist die Form, die sich in der Anzahl von Silben (sogenannten Moren) widerspiegelt. Diese Form existiert leider nur im Japanischen, da bei der Übersetzung in andere Sprachen eine andere Anzahl von Silben entsteht.

Ein Haiku besteht aus drei Zeilen (Wortgruppen) mit jeweils (5-7-5) Lauteinheiten, insgesamt somit 17 Silben.

Ein Tanka besteht aus fünf Zeilen (Wortgruppen) mit jeweils (5-7-5-7-7) Lauteinheiten, insgesamt somit 31 Silben.

Alle vorkommenden Anzahlen von Silben sind Primzahlen! Haikus und Tankas sind reimlos und handeln meist (aber nicht immer) von der Natur. Haikus und Tankas sollen Gefühle und erlebte Momente vermitteln. Im Gegensatz zur in der abendländischen Dichtung vorherrschenden äußeren Struktur des Reimes sind es die ‚primen‘ Silbenanzahlen, die jedem Gedicht eine besondere Struktur innewohnen lassen.

Daniel Tammet hat sich in seinem Buch „Die Poesie der Primzahlen“ sehr ausführlich mit diesem Thema beschäftigt. (Tammet, 2014)⁷⁴. Tammet schreibt:

„Primzahlen tragen zur elementaren Einfachheit der Haiku-Form bei. Jedes Wort und jedes Bild verlangt unsere ungeteilte Aufmerksamkeit. Das Ergebnis ist eine Impression wie eine plötzliche, tiefe Erkenntnis, als ob der Gegenstand des Gedichts zum allerersten Mal in Worte gefasst worden sei.... Wenn ich über die Komplizenschaft zwischen Poemen und Primzahlen nachdenke, erscheint mir als vielleicht einzige Überraschung daran, dass wir sie überhaupt überraschend finden. Auf eine bestimmte Weise betrachtet ist sie eine völlig selbstverständliche Beziehung. Dichtung und Primzahlen haben eines gemeinsam: Beide sind ebenso unberechenbar, schwer definierbar und vieldeutig wie ein Leben.... Gedichte wie Primzahlen sind nur schwer zu erkennen. Ein kurzer Blick genügt

⁷⁴ „Die Poesie der Primzahlen“, S. 227 (Hanser-Verlag München, 2014)

gewöhnlich nicht, um uns zu sagen, ob eine bestimmte Zahl Teiler oder ein bestimmter Text eine tiefere Bedeutung hat...“

Daniel Tammet ist einer von etwa 100 ‚inselbegabten‘, die weltweit leben. Er lernt neue Sprachen binnen einer Woche und rechnet fast so schnell wie ein Computer. Außerdem hält er den Europarekord im auswendigem Aufsagen von Ziffern der Zahl π (22514 Dezimalstellen, Stand: Juni 2016).

16.2 SESTINE

Eine andere, längst in Vergessenheit geratene Gedichtform, in der uns auch Primzahlen begegnen ist die Sestine. Die Sestine beschreibt eine Gedichtform aus sechs Strophen zu je sechs Zeilen mit einer Schluss-Strophe von drei Zeilen. Der Name stammt von dem Wort ‚sesto‘ (sechs). Als Erfinder der Sestine gilt der französische Troubadour Arnaut Daniel, der von 1150 bis 1200 in der Provence lebte.

Ähnlich wie das japanische Haiku wird die Sestine nicht durch Formen wie Reim oder Symbolik zusammen gehalten. Anders als beim Haiku spielt jedoch die Anzahl der Silben oder Wörter pro Zeile keine große Rolle. Im Deutschen wird jedoch das jambische Versmaß bevorzugt.

Die Struktur, die eine Sestine zusammenhält ist folgende: Jede Sestine hat einen ‚Kern‘ von sechs Wörtern. Jeweils das letzte Wort einer Zeile muss eines dieser 6 Kern-Wörter sein, und zwar alternierend, bis alle sechs Kernwörter aufgebraucht sind, was offensichtlich nach sechs Zeilen der Fall ist. Für die nächste Gruppe von 6 Zeilen gilt das gleiche, jedoch mit einer anderen Reihenfolge der Kernwörter, die jeweils am Zeilenende stehen.

Man sieht, dass die ‚Kraft‘ und die fast musikalisch zu nennende Anziehungskraft in der Wiederholung liegt. Im Laufe des 36-zeiligen Gedichtes kommt jedes Kernwort genau 6-mal vor. Die Reihenfolge, wie innerhalb einer Sechsergruppe die Kernwörter am Ende permutiert werden ist kompliziert und erinnert mehr an das Werfen eines Würfels. Nehmen wir an, unsere Kernwörter sind von 1 bis 6 durchnummeriert, dann kommen die (am Ende stehenden) Kernwörter im gesamten Gedicht in folgender Reihenfolge vor:

Strophe 1: 1,2,3,4,5,6
 Strophe 2: 6,1,5,2,4,3
 Strophe 3: 3,6,4,1,2,5
 Strophe 4: 5,3,2,6,1,4
 Strophe 5: 4,5,1,3,6,2
 Strophe 6: 2,4,6,5,3,1

Schlussstrophe: (1,2),(3,4),(5,6)

Hinweis: In der dreizeiligen Schluss-Strophe kommen pro Zeile jeweils zwei Kernwörter (eines am Ende und eine innerhalb der Zeile) vor. Die Reihenfolge der Kernwörter in der Schluss-Strophe kann jedoch auch anders lauten, es muss in den Schlusszeilen jedes Kernwort genau einmal vorkommen.

Hier ein Beispiel einer Sestine, das der Autor im Internet gefunden hat:⁷⁵

Martin Opitz, in "Schäfferey von der Nimfen Hercinie"

*Wo ist mein Auffenthalt, mein Trost und schönes Liecht?
Der trübe Winter kömpt, die Nacht verkürzt den Tag;
Ich irre gantz betrübt umb diesen öden Waldt.
Doch were gleich jetzt Lentz und Tag ohn alle Nacht
Und hett' ich für den Wald die Lust der gantzen Welt,
Was ist Welt, Tag und Lentz, wo nicht ist meine Zier?*

*Ein schönes frisches Quell giebt Blumen ihre Zier,
Dem starcken Adler ist nichts liebers als das Liecht,
Die süsse Nachtigal singt frölich auff den Tag,
Die Lerche suchet Korn, die Ringeltaube Waldt,
Der Reiger einen Teich, die Eule trübe Nacht;
Mein Lieb, ich suche dich für allem auff der Welt.*

*So lange bist du mir das liebste von der Welt.
So lange Pales hegt der grünen Weide Zier,
So lange Lucifer entdeckt das klare Liecht.
So lange Titans Glantz bescheint den hellen Tag,
So lange Bacchus liebt den Wein und Pan den Waldt,
So lange Cynthia uns leuchtet bey der Nacht,*

*Die schnelle Hindin sucht den Hirschen in der Nacht,
Was schwimmt und geht und kreucht, liebt durch die gantze Welt,
Die grimme Wölffin schätzt den Wolff für ihre Zier,
Die Sternen leihen uns zum Lieben selbst ihr Liecht;
Ich aber gehe nun allhier schon manchen Tag,
O Schwester, ohne dich durch Berge, Wildt und Wald.*

*Was ist, wo du nicht bist? So viel der kühle Waldt
Ein Sandfeldt übertrifft, der Morgen für der Nacht
Uns angenemer ist, der Mahler dieser Welt,
Der Lentz, für Winterlufft, so viel ist deine Zier,
Die Schönheit, diese Lust mir lieber, o mein Liecht,
Als das, so weit und breit bestralt wird durch den Tag.*

*Der Trost erquickt mich doch, es komme fast der Tag,
Da ich nicht werde mehr bewohnen Berg und Wald,
Da deine Gegenwart und die gewünschte Nacht
Der Treu noch lohnen soll; in dessen wird die Welt
Vergessen ihrer selbst, eh' als ich deiner Zier,
Mein höchster Auffenthalt, mein Trost und schönes Liecht.*

*Laß wachsen, edler Wald, mit dir mein treues Liecht,
Die liebste von der Welt; es schade deiner Zier,
O Baum, kein heisser Tag und keine kalte Nacht.*

Die ‚durchpermutierten‘ Kernwörter an den Zeilenenden ähneln den Permutationen (Perioden) der Ziffern einer zyklischen Zahl. Zyklische Zahlen entstehen bei der Division

⁷⁵ <http://www.leselupe.de/lw/titel-Gedichtstrophen--Die-Sestine-100219.htm>

durch Primzahlen. So entstehen z. B. die sechs zyklischen Zahlen 1,4,2,8,5,7 wenn man die Zahl 1 durch die Primzahl 7 dividiert:

$$1/7 = 0,142857\ 142857\ 142857\ \dots$$

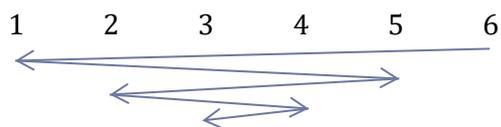
Die Permutationen dieser 6 Zahlen 1,4,2,8,5,7 entstehen nun, in dem wir die Zahl 142857 jeweils mit allen ganzen Zahlen $1 \leq n < 7$ multiplizieren:

$$\begin{aligned} 142857 \cdot 1 &= 142857 \\ 142857 \cdot 2 &= 285714 \\ 142857 \cdot 3 &= 428571 \\ 142857 \cdot 4 &= 571428 \\ 142857 \cdot 5 &= 714285 \\ 142857 \cdot 6 &= 857142 \end{aligned}$$

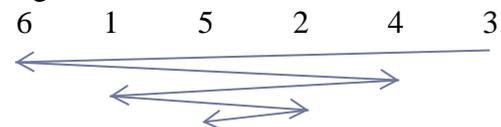
Das Ganze hat eine verblüffende Ähnlichkeit mit den Permutationen der Kernwörter der Sestine!

Noch einmal zurück zur Reihenfolge der jeweils am Ende stehenden Kernwörter:

Das Bildungsgesetz der zunächst kompliziert erscheinenden Permutationen ist recht einfach und sei hier durch folgendes Zick-Zack-Schema verdeutlicht:



ergibt:



ergibt:



usw....

Es stellt sich nun die Frage, wieso gerade die Anzahl von sechs Strophen eine derartige Bedeutung in der Dichtung erlangte, und nicht etwa Gedichte aus vier Strophen (Tetrinen) oder sieben Strophen (Septine). Die durch ihre Form begründete Schönheit der Sestine liegt darin, dass sich nach 6-maligem iterativen Anwenden des Zick-Zack-Schemas auf die ‚Ursprungsreihenfolge‘ 1,2,3,4,5,6 wieder die gleiche Reihenfolge 1,2,3,4,5,6 ergibt und dass jeweilige Kernwort bei jeder Strophe bei einer anderen Zeilennummer vorkommt. Wir verallgemeinern nun und fordern, dass für eine ‚schöne‘ ‚n-tine‘ das gleiche Prinzip gelten soll:

Unsere Ausgangsreihenfolge der Kernwörter sei : 1,2,3, ..., n.

Bei (n-1)-maliger Anwendung des Zick-Zack-Schemas soll bei jedem Iterationsschritt der Index des Kernwortes unterschiedlich sein (d.h. ein Kernwort darf nicht in zwei unterschiedlichen Strophen in der gleichen (z. B. der 3.ten Zeile) vorkommen).

Sestine

Doch genau das passiert für manche Werte von n . Z. B. würde eine ‚Septine‘ zu unschönen Worthäufungen des jeweils gleichen Kernwortes in der jeweils 5. Zeile führen (hier das Schema der Kernwort-Indices):

Strophe 1: 1,2,3,4,**5**,6,7
Strophe 2: 7,1,6,2,**5**,3,4
Strophe 3: 4,7,3,1,**5**,6,2
Strophe 4: 2,4,6,7,**5**,3,1
Strophe 5: 1,2,3,4,**5**,6,7

Außerdem wiederholt sich das Schema schon nach 4 Permutationen und nicht wie zu erwarten erst nach 7 Permutationen.

Mit den Forderungen nach dem Prinzip für schöne ‚ n -tinen‘ ergibt eine genauere Untersuchung für Werte von 3 bis n folgende Ergebnisse:

‚ n -tinen‘ sind ‚schön‘, wenn $n \cdot 2 + 1$ eine Primzahl ergibt. Deshalb sind ‚Tritinen‘, ‚Quintinen‘, Sestinen oder ‚11-tinen‘ ‚schön‘, nicht jedoch ‚Quartinen‘ oder ‚Dezinen‘... Diese Bedingung ist für 31 Zahlen $n < 100$ erfüllt.

Hinweis: Eine einfachere Variante der Sestine ist die aus dem Arabischen stammende Gedichtform des Ghasel, bei dem es nur ein Kernwort (letztes Wort einer Zeile) gibt, das für jede zweite Zeile wiederholt wird.⁷⁶

⁷⁶ <https://de.wikipedia.org/wiki/Ghasel>

Archaischer Torso Apollos

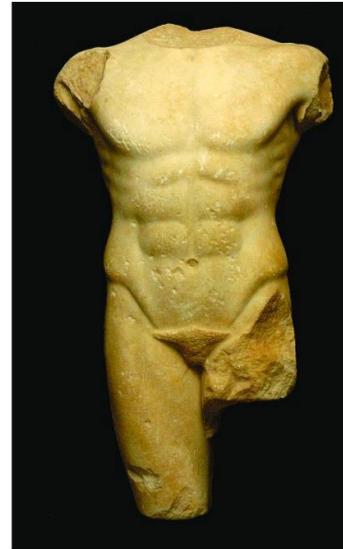
*Wir kannten nicht sein unerhörtes Haupt,
darin die Augenäpfel reiften. Aber
sein Torso glüht noch wie ein Kandelaber,
in dem sein Schauen, nur zurückgeschraubt,*

*sich hält und glänzt. Sonst könnte nicht der Bug
der Brust dich blenden, und im leisen Drehen
der Lenden könnte nicht ein Lächeln gehen
zu jener Mitte, die die Zeugung trug.*

*Sonst stünde dieser Stein entstellt und kurz
unter der Schultern durchsichtigem Sturz
und flimmerte nicht so wie Raubtierfelle;*

*und bräche nicht aus allen seinen Rändern
aus wie ein Stern: denn da ist keine Stelle,
die dich nicht sieht. Du musst dein Leben ändern.*

(Rainer Maria Rilke)



Dieses Gedicht ist eines der schönsten, aber auch rätselhaftesten Gedichte von Rilke. Der Leser möge herausfinden, was es mit Primzahlen zu tun hat...

Schon aus der Form des Gedichtes (Sonett), sehen wir, dass dieses Gedicht nicht nur Text-Botschaften enthält, sondern noch viel mehr aussagen soll. Es berührt uns auf eine Art und Weise, die man eigentlich nur mit dem Wort ‚geheimnisvoll‘ beschreiben kann.

Die Interpretationen dieses Gedichtes gehen weit auseinander. Manche interpretieren in dieses Gedicht die zentralen, philosophischen Fragen des menschlichen Lebens hinein: „Wer bin ich?“, „Was soll ich tun“ und finden in Rilkes Gedicht auch Antworten (zumindest Andeutungen von Antworten...).

Eine sehr schöne Interpretation findet sich in dem Buch „Vom musikalischen Denken“⁷⁷.

Darin beschreibt Zucker кандl, wie die Situation von Betrachter und betrachtetem Objekt sich plötzlich umkehren: Das betrachtete Kunstwerk wird zum Betrachter und schaut ‚mich‘ an, der somit zum ‚Angeschauten‘ wird. Dieser wortlos auf den Betrachter gerichtete Blick endet mit der Forderung: „Du musst Dein Leben ändern“. Für die weitere Interpretation zitiert Zucker кандl die Geschichte vom Schmetterlingstraum von Dschuang-Dschou⁷⁸, die so schön ist, dass sie hier auch mit zitiert sei:

⁷⁷ Victor Zucker кандl: ‚Vom musikalischen Denken‘ (S.151), Rhein-Verlag Zürich, 1964

⁷⁸ Dschuang Dsi: Das wahre Buch vom südlichen Blütenland: Eugen Diederichs Verlag München (1988)

Schmetterlingstraum

Einst träumte Dschuang Dschou, dass er ein Schmetterling sei, ein flatternder Schmetterling, der sich wohl und glücklich fühlte und nichts wusste von Dschuang Dschou. Plötzlich wachte er auf: da war er wieder wirklich und wahrhaftig Dschuang Dschou. Nun weiß ich nicht, ob Dschuang Dschou geträumt hat, dass er ein Schmetterling sei, oder ob der Schmetterling geträumt hat, dass er Dschuang Dschou sei, obwohl doch zwischen Dschuang Dschou und dem Schmetterling sicher ein Unterschied ist. So ist es mit der Wandlung der Dinge.

Dieses tief bedeutungsvolle Gleichnis weist (wie Rilkes Gedicht) darauf hin, dass beide Ebenen gleich ‚wirklich‘ sind und im Gleichgewicht stehen. Jedes Kunstwerk besitzt ein Eigenleben, es ist tatsächlich lebendig. Daher kann es auch ‚mich‘ betrachten. Für den Künstler und den kreativen Prozess des Erschaffens bedeutet dies, dass ein Kunstwerk (sobald eine gewisse Schwelle während des Erschaffungsprozesses überschritten wird) quasi anfängt „lebendig“ zu werden und somit auch einen eigenen Willen besitzt. Das Kunstwerk ‚kommuniziert‘ mit dem Künstler und möchte an seiner ‚Erschaffung‘ mitgestalten, es möchte sich ‚verwirklichen‘...

Die Form, d. h. ihre zahlenmäßige Struktur, in der Primzahlen eine Rolle spielen, sowie das Vermaß und natürlich auch der eigentliche Text bilden ein komplexes Geflecht, in dem der Begriff ‚Ästhetik‘ erst einen Sinn ergibt.

Dass dieser innewohnende Sinn nicht immer rational und logisch in Worte gefasst werden kann, jedoch von unserem ästhetischen Empfinden sehr wohl ‚verstanden‘ werden kann, erinnert uns sehr an die im letzten Kapitel besprochenen Haikus und andere japanische Gedichtformen. Auch hier scheinen die zwei (scheinbar) unabhängigen Bedeutungsebenen ‚Form‘ und ‚Inhalt‘ untrennbar miteinander verbunden zu sein, oder mit den Worten von Ludwig Wittgenstein gesprochen:

***Ethik und Ästhetik sind Eins*⁷⁹.**

Man müsste Wittgensteins Ausführungen weiter vertiefen, um genau zu verstehen, was er damit meint. Ähnlichkeiten zu vielen Koans aus dem Zen-Buddhismus sind offensichtlich. Hier ein Beispiel:

*„Das Auge, mit dem ich Gott erblicke,
ist genau das gleiche Auge, mit dem Gott mich erblickt.“
„Zeige mir dieses Auge!“*

Der mathematisch interessierte Leser findet in diesen Gedichten sofort das Element der Selbstbezüglichkeit und der Rekursion. Die Rekursion scheint somit nicht nur in der Mathematik ein sehr mächtiges Instrument zu sein, sie scheint auch in der Philosophie und der Dichtung ein Mittel zu sein, um Dinge ‚auszudrücken‘, die durch Worte eben nicht ausdrückbar sind: Erkenntnisse einer höheren Ebene der Wahrheit?

⁷⁹ Aus dem ‚Tractatus logico-philosophicus‘ von Ludwig Wittgenstein

17 PRIMZAHLEN UND AUßERIRDISCHE LEBENSFORMEN

Wenn in nicht allzu ferner Zukunft der Fall eintreten wird, dass wir elektromagnetische Signale von außerirdischen Zivilisationen empfangen werden, wird eine Diskussion in Gang kommen, wie eine Kommunikation mit außerirdischen intelligenten Lebewesen aussehen könnte. Diese Suche ist seit Jahren in Gang und wird hauptsächlich vom Projekt ‚SETI‘ („Search for Extraterrestrial Intelligence“) in Mountain View bei San Francisco betrieben.⁸⁰

Zunächst einmal muss man sich darüber im Klaren sein, dass der nächste mit intelligenten Lebewesen bewohnte extraterrestrische Planet nicht in unserer unmittelbaren Nachbarschaft zu finden sein wird, sondern aller Wahrscheinlichkeit nach etliche hundert, wenn nicht Tausende von Lichtjahren von der Erde entfernt sein wird. Kommunikation könnte sich also nur in Zeiträumen von vielen Jahrhunderten abspielen. In welcher ‚interstellaren Sprache‘ sollen wir Botschaften senden, bzw. erwarten wir interstellare Botschaften?

Grundsätzlich gilt, dass beide Kommunikationspartner etwas finden müssen was beiden gemeinsam ist. Das ist zum einen der (nach unserem derzeitigen Kenntnisstand einzige praktikable) Übertragungsweg der elektromagnetischen Wellen. Man kann davon ausgehen, dass außerirdische Lebensformen, sobald sie ebenfalls auf der Suche nach ‚kosmischen‘ Nachbarn sind, diese Technik beherrschen. Die zweite, physikalisch existierende Wellenform, die sich ebenfalls über große Distanzen ausbreitet, sind die vor kurzem nachgewiesenen Gravitationswellen. Theoretisch wäre es möglich, dass außerirdische Lebewesen sich auch dieser Möglichkeit bedienen. Wir verfügen jedoch z. Zt. noch über keine Technik, die es erlauben würde, Signale in modulierten Gravitationswellen aufzuspüren.

Die Sprache, die vermutlich alle intelligente Lebensformen im Universum beherrschen, ist die Sprache der Mathematik, die jeden Bereich unserer Existenz ‚durchdringt‘. Unsere ganze Physik ist in der Sprache der Mathematik geschrieben. Somit können wir uns sicher sein, dass die Sprache der Mathematik im ganzen Universum ‚verstanden‘ wird. Hinweis: Es gibt auch Theorien, die von einer Vielzahl von Universen ausgehen, die unter Umständen auch eine völlig andere Physik und vielleicht sogar eine andere Mathematik ‚besitzen‘ (Tegmark, 2015). Wenn solche Universen existieren, wären sie unserem Universum sowieso nicht ‚physikalisch‘ zugänglich. Wir beschränken uns hier für diese Überlegungen auf unser, für uns beobachtbares Universum.

Primzahlen wären ideal für solche interstellaren Botschaften, da sie nicht auf natürliche Art und Weise ohne das Einwirken einer Intelligenz entstehen können. Unser Weltall enthält sehr wohl modulierte elektromagnetische Signale, die meistens von rotierenden Neutronensternen oder anderen physikalischen Prozessen stammen (überall dort, wo Ladungen stark beschleunigt werden). Allen diesen bisher beobachteten Signalen ist jedoch gemeinsam, dass sie mehr oder weniger periodisch sind, und deshalb keine Information enthalten.

Da die formale Sprache, in der die Mathematik (an den verschiedenen Orten im Weltall...) betrieben wird, sich auch unterscheiden wird, ist es vernünftig, eine Botschaft zunächst

⁸⁰ <http://www.seti.org>

Zum Nachdenken

einmal möglichst einfach zu verfassen. Die einfachste Möglichkeit, die die Mathematik bietet, sind die natürlichen Zahlen. Deshalb sind sich alle Experten, die sich mit extraterrestrischer Kommunikation beschäftigt haben, darüber einig, dass eine Übertragung der (sagen wir ersten, aus dem Bereich bis 100 stammenden) Primzahlen auf möglichst vielen interessanten Frequenzen (dazu gehört z. B. die Absorptionsfrequenz von Wasserstoff) eine hervorragende Möglichkeit der interstellaren Kommunikation bietet. Die Information könnte z. B. in gepulsten ‚Paketen‘ stecken, wobei die zeitlichen Abstände zwischen den Pulsen proportional zu den Distanzen der Primzahlen sind.

Genau dies passiert in dem Film ‚Contact‘⁸¹ aus dem Jahr 1997 mit Jodie Foster in der Hauptrolle.

81

https://web.archive.org/web/20071125172406/http://www.cisci.net/film.php?lang=2&display=5&topic=Astronomie%20und%20Astrophysik&seq_id=42&film=26

17.1 DIE ARECIBO-BOTSCHAFT

Am 16. November 1974 sandten Wissenschaftler des Arecibo-Observatoriums in New-Mexico eine Nachricht der Menschheit ins Weltall. Genau genommen in Richtung des 25000 Lichtjahre entfernten Kugelsternhaufens M13, unter Astronomen auch unter der Bezeichnung NGC 6205 bekannt. Dieser Kugelsternhaufen ist in einer klaren Nacht mit bloßem Auge sichtbar und befindet sich im Sternbild des Herkules zwischen den Sternen η und ξ , oberhalb von ξ Herculis:

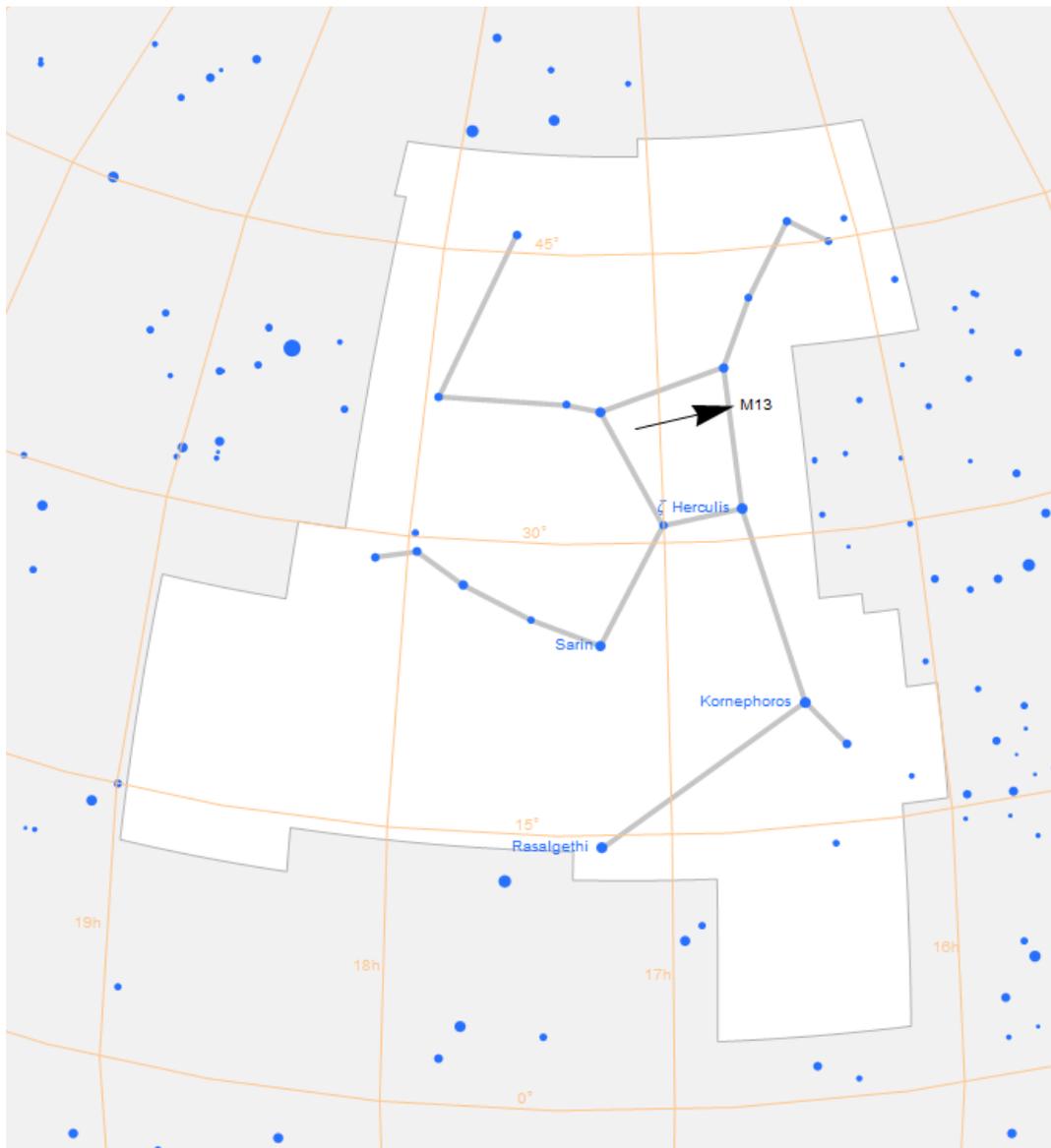


Abbildung 134: Sternbild Herkules, mit Kugelsternhaufen M13 (Ziel der Arecibo-Botschaft)

```
Mathematica:  
ConstellationData["Herculis", "ConstellationGraphic"]
```

Die Botschaft war digital und bestand aus 1679 Nullen und Einsen. Da der Kugelsternhaufen in astronomischen Verhältnissen relativ nah bei unserem Sonnensystem liegt und eine hohe Sternendichte aufweist (er besteht aus etwa 300000

Die Arecibo-Botschaft

Einzelsternen) schien er als Ziel für eine Botschaft an außerirdische Lebensformen ideal. Ein außerirdischer Empfänger müsste zunächst die Länge der Botschaft (1679 Bit) als Produkt der Primzahlen 23 und 73 erkennen und somit als zweifarbige Bild mit den Abmessungen 73*23 interpretieren. Dann ergibt sich aus der binären Sequenz folgendes Bild:

Nur wenn die Sequenz der Arecibo-Botschaft als Bild mit den Abmessungen der Primfaktorenzerlegung dargestellt wird, läßt sich die Botschaft entschlüsseln: Oben befindet sich zunächst das grundlegende ‚Alphabet‘ der binären Codierung der Zahlen 1 bis 10. Dies stellt sozusagen eine Art ‚Leseanleitung‘ für die folgenden Darstellungen. In dem Bild folgen danach Informationen über unsere chemischen Elemente, Aminosäure-Nucleotide, DNS-Struktur, die Menschheit, unseren Planeten Erde, usw...

Genauere Informationen findet der Leser im Internet.

Der Mathematica Code findet sich im Anhang.

Mit einer Antwort auf die Botschaft aus dem Sternhaufen M13 oder seiner ‚kosmischen Nachbarschaft‘ wäre erst in ca. 50000 Jahren zu rechnen.

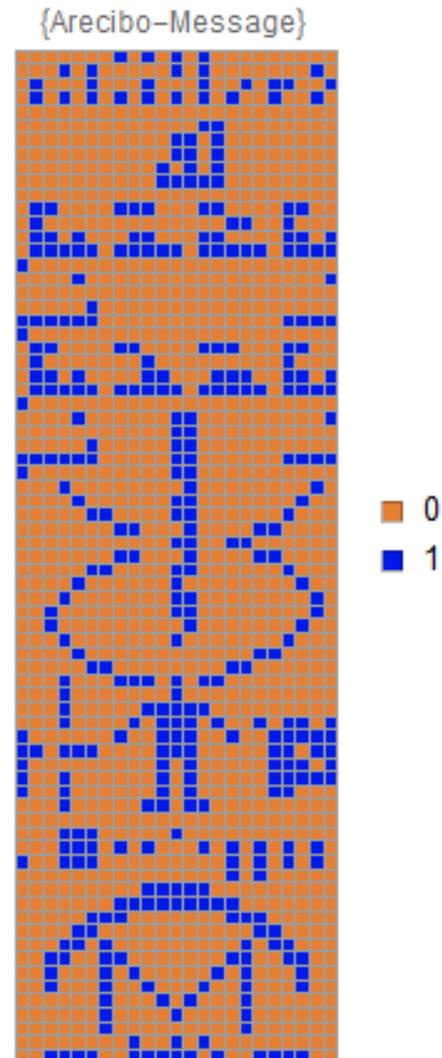


Abbildung 135: Arecibo-Message

18 SONSTIGES

18.1 DIE ZAHL 12

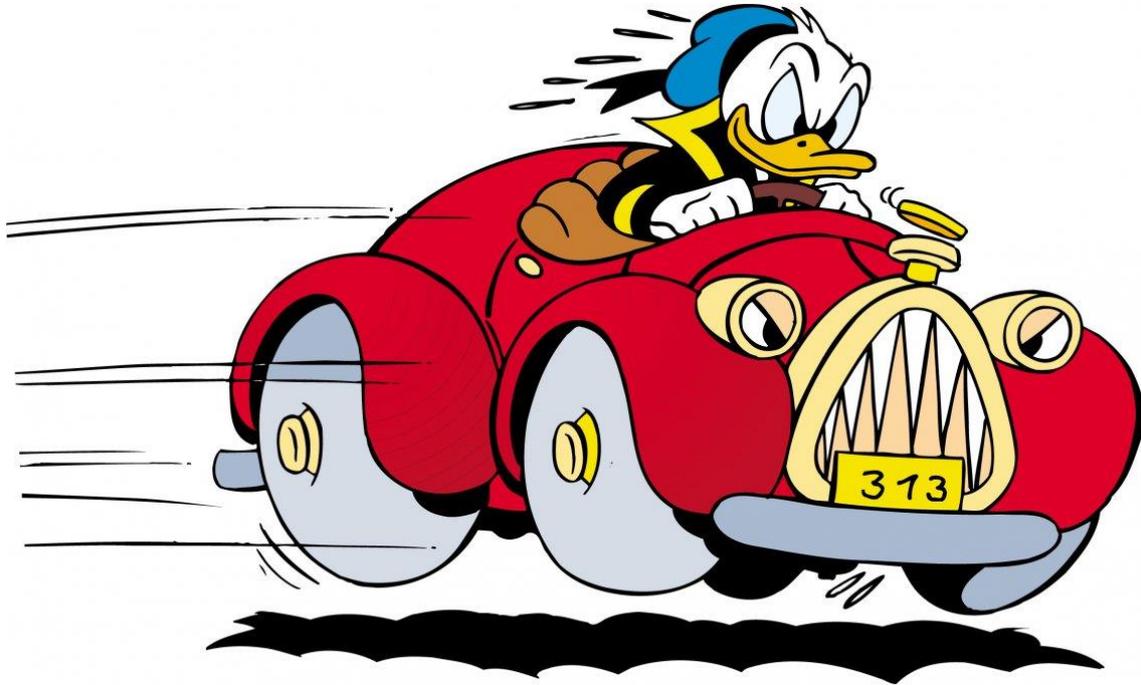
Die Zahl 12 besitzt viele besondere Eigenschaften:

- 1) Sie ist/war Bestandteil von Maß-und Zähl Systemen in früheren Zeiten. Als Beispiel seien hier nur die englische 12-Pence Münze genannt, und die Zählinheit ‚Dutzend‘.
- 2) Das Jahr besitzt 12 Monate, der Tag hat zwei Mal 12 Stunden.
- 3) Jesus hatte 12 Apostel; Israel besaß 12 Stämme.
- 4) Die Oktave besitzt 12 Halbtöne.
- 5) Sie ist die kleinste ‚abundante‘ Zahl (abundante Zahlen sind Zahlen, deren echte Teilersumme größer ist als die Zahl selbst...)
- 6) Dreidimensionale Kusszahl ⁸²
- 7) Es gibt 12 Tierkreiszeichen.
- 8) Es gibt 12 Olympische Götter.
- 9) Die 12-te Fibonacci-Zahl ist unter den unendlich vielen Fibonacci-Zahlen die einzige in dieser Folge vorkommende Quadratzahl und hat den Wert $12^2 = 144$
- 10) Sie ist die kleinste erhabene Zahl (es sind z. Zt. zwei erhabene Zahlen bekannt). Hinweis: Eine erhabene Zahl ist eine Zahl bei der die Summe ihrer Teiler und auch die Anzahl ihrer Teiler vollkommene Zahlen sind. Die zweite bekannte erhabene Zahl lautet:
6086555670238378989670371734243169622657830773351885970528324860512791691264
- 11) Sie spielt in der Musik eine besondere Rolle: Das 12-taktige Blues-Schema.
- 12) Die Zahl 12 ist die einzige Zahl n , für die der bemerkenswerte Zusammenhang $n = \frac{r_4(n)}{8}$ gilt (wobei $r_4(n)$ die Anzahl der vierdimensionalen Gitterpunkte eines quadrierten Radius von n darstellt).
- 13) Das Weltbild des Physikers Burkhard Heim enthält genau 12 Dimensionen.
- 14) Sie taucht auf in Ramanujans magischer Formel: $1 + 2 + 3 + 4 + \dots = -\frac{1}{12}$
- 15) Der 12.te Mersenne Primzahlexponent (127) besitzt die auffällig kurze OCRON-Darstellung "2PPPP" und enthält weder den ^ noch den * -Operator (siehe Kapitel 10.1).

⁸² <https://de.wikipedia.org/wiki/Kusszahl>

18.2 DIE ZAHL 313

Auch in Comics kommen Primzahlen vor! Wer kennt nicht Donald Ducks knallrotes Enten-Cabrio („1934 Belchfire Runabout“) mit der Nummer 313?



Die 313 ist in mehrfacher Hinsicht eine besondere Zahl:

- Sie ist eine 3-stellige palindromische Primzahl (ergibt vorwärts und rückwärts gelesen den gleichen Wert)
- Sie ist in Binärdarstellung (100111001) palindromisch und 100111001 dezimal interpretiert ergibt ebenfalls eine palindromische Primzahl!

Die Internetseite <https://primes.utm.edu/curios/page.php/313.html> listet über weitere 40 besondere Eigenschaften dieser Zahl...

18.3 PRIMZAHLEN UND KUNST

Das folgende Bild zeigt die Primzahlen in der Gausschen Zahlenebene, nachdem die zweifarbige (Rot/Gelb)-Darstellung fouriertransformiert, im Frequenzbereich durch Division mit den Frequenzen ‚integriert‘, und anschließend wieder zurücktransformiert wurde. Eine Spielerei zwar, jedoch sieht man, das mit wenig mathematischen Mitteln aus Primzahlen landschaftsähnliche Grafiken erzeugt werden können.

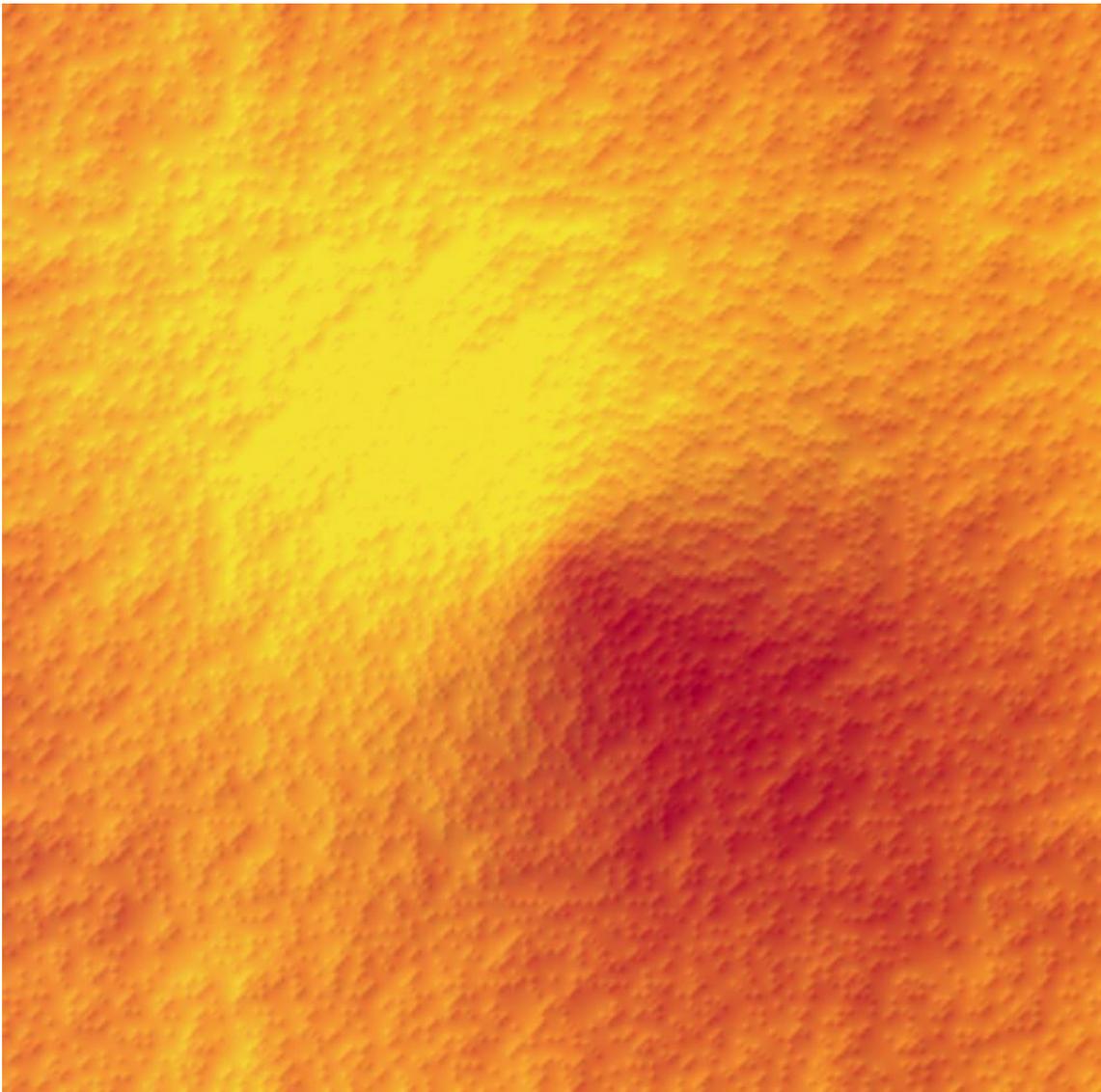


Abbildung 136: Gausssche Primzahlen, mit Hilfe von Fouriertransformationen ‚gefiltert‘

19 SCHLUSSBETRACHTUNG

Primzahlen faszinieren uns Menschen, seit wir Mathematik betreiben. Obwohl wir inzwischen sehr viel über sie wissen, haben sie nichts von ihrer Faszination verloren. Sie durchdringen viele Gebiete aller möglichen Wissenschaften und sind auch in kulturellen Bereichen, wie z. B. der Poesie und auch im Wirtschaftsleben zu finden. Wir haben viel über sie verstanden, nicht jedoch die eigentliche ‚Botschaft‘, die in ihnen steckt.

Es bleiben noch viele Geheimnisse zu lösen!

20 ANHANG

20.1 DIE CATALANSCHER VERMUTUNG

Die Catalansche Vermutung besagt, dass es keine ganzzahligen Potenzen natürlicher Zahlen gibt, die sich genau um den Wert 1 unterscheiden, mit einer Ausnahme:

$$2^3 = 8 \text{ und } 3^2 = 9$$

Anders formuliert: Die einzige ganzzahlige Lösung der Gleichung

$$m^p - n^q = 1, \text{ mit } m, n, p, q > 1 \text{ lautet } m = 3, n = 2, p = 2, q = 3 \quad (159)$$

Sie wurde im Jahr 2002 von Preda Mihailescu bewiesen.

Der Beweis gelang mit Hilfe der ‚Doppelten Wieferich-Primzahlen‘ (siehe 4.14).

20.2 STATISTISCHE AUFFÄLLIGKEITEN DER ENDZIFFERN IN DER PRIMZAHLFOLGE

Wie sehen die statistischen Auffälligkeiten aus, wenn wir noch mehr von den vorangegangenen Primzahlen mituntersuchen? Hier die Ergebnisse, wenn wir nicht nur die Vorgänger, sondern auch die Vor-Vorgänger berücksichtigen:

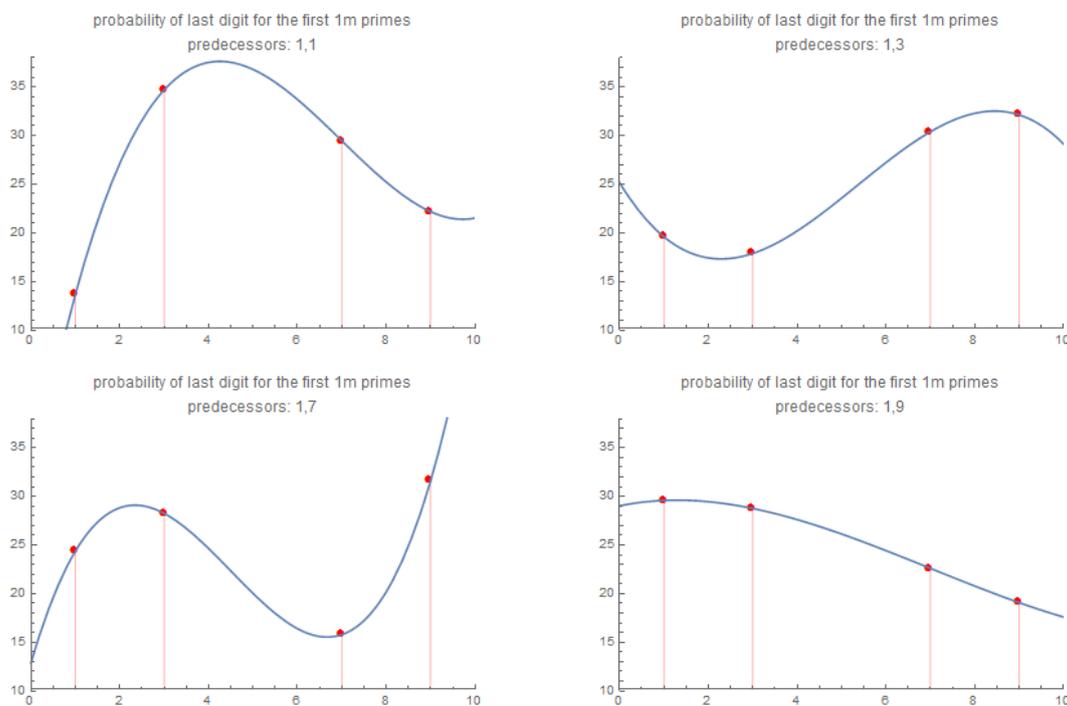


Abbildung 137: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:1,x)

Statistische Auffälligkeiten der Endziffern in der Primzahlfolge

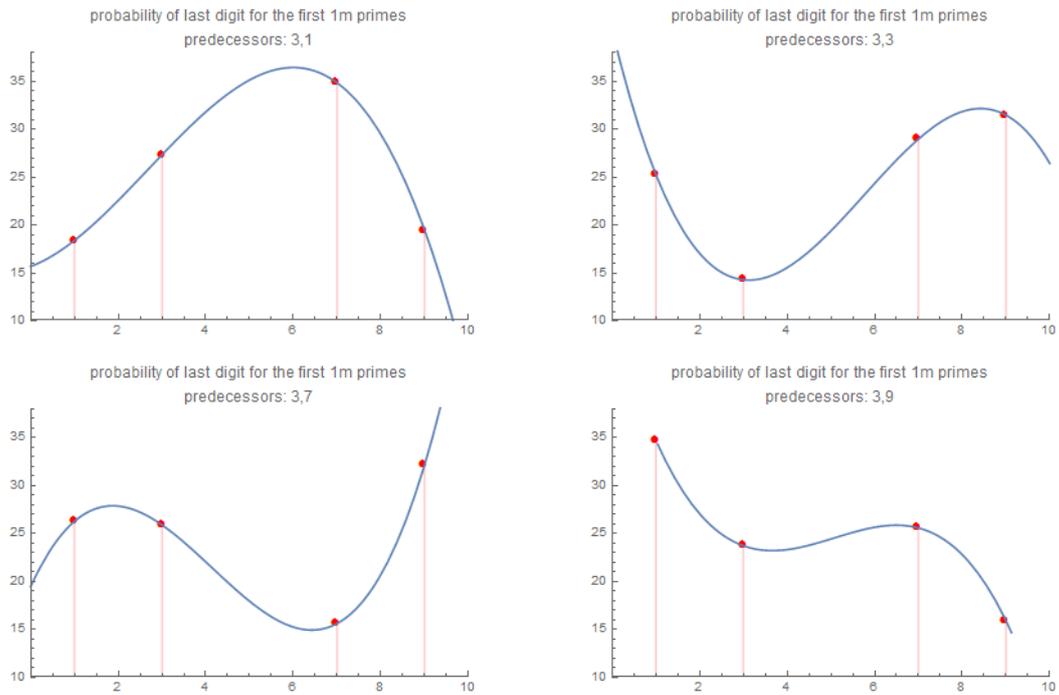


Abbildung 138: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:3,x)

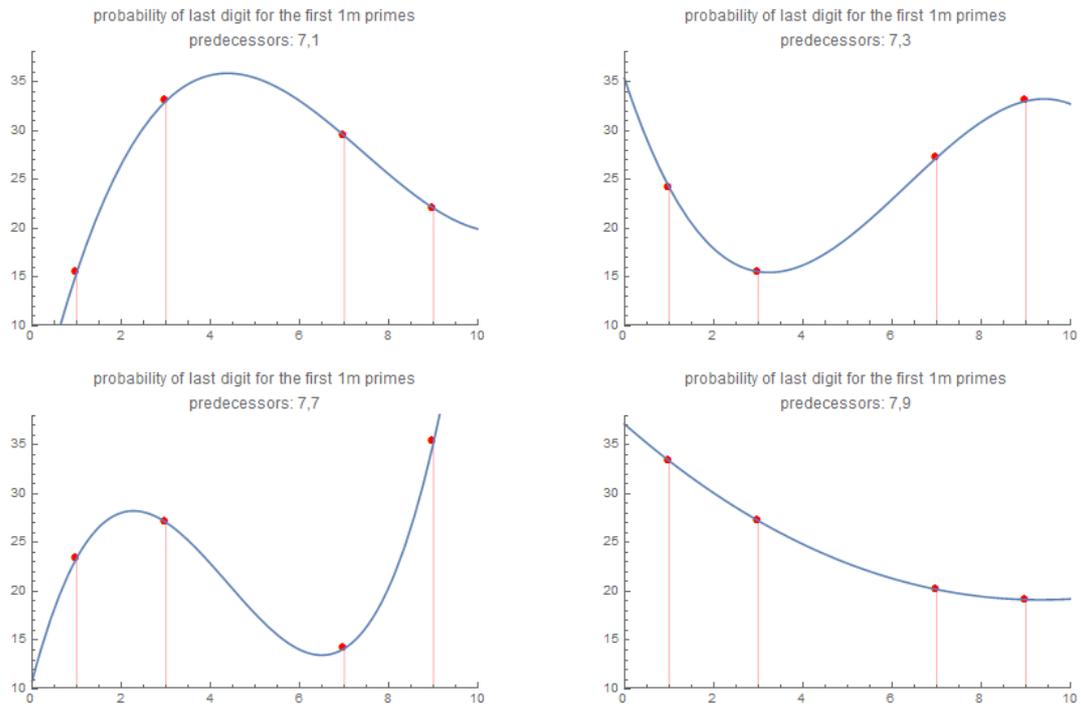


Abbildung 139: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:7,x)

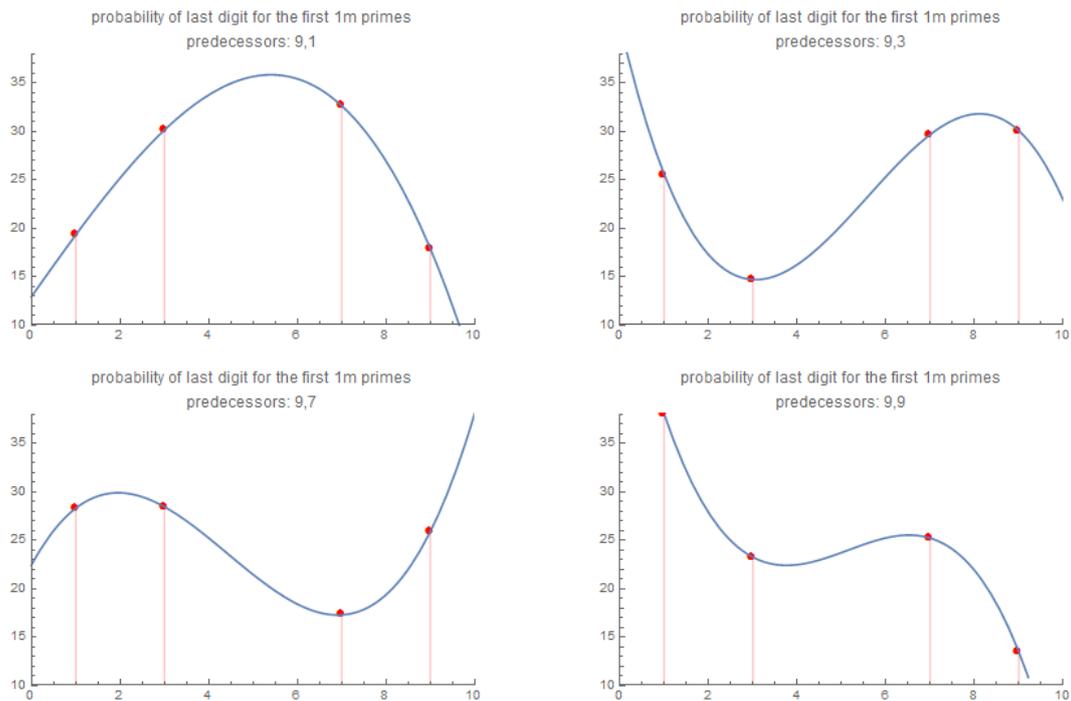


Abbildung 140: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:9,x)

Man sieht, dass die Tendenz der Endziffern, sich nicht zu wiederholen, sich noch verstärkt. Z. B. beträgt die Wahrscheinlichkeit, dass nach zwei Primzahlen mit der Endziffer 9 die nächste Primzahl wieder eine Endziffer 9 hat, nur noch 13.48 %.

20.3 EINE INTERESSANTE FOLGE: DIE PERRIN-FOLGE

Die Perrin-Folge (auch Skiponacci-Folge genannt) ist eine interessante Kuriosität: Ihre rekursive Definition lautet:

$$a(n) = a(n-2) + a(n-3), \quad \text{mit } a(0) = 3, a(1) = 0, a(2) = 2 \quad (160)$$

Sie wurde eigentlich von Edouard Lucas im Jahr 1878 entdeckt. Das Besondere ist, dass das p -te Folgenglied durch p teilbar ist (oder $\text{mod } p \ 0$ ergibt), wenn p eine Primzahl ist.

Mathematica-Code:

```
LinearRecurrence[{0, 1, 1}, {3, 0, 2}, 50]
```

```
{0, 2, 3, 2, 5, 5, 7, 10, 12, 17, 22, 29, 39, 51, 68, 90, 119, 158, 209, 277, 367, 486, 644, 853, 1130, 1497, 1983, 2627, 3480, 4610, 6107, 8090, 10717, 14197, 18807, 24914, 33004, 43721, 57918, 76725, 101639, 134643, 178364, 236282, 313007, 414646, 549289, 727653, 963935}
```

Oder, besser sichtbar (mit ,0' an den Primzahl-Positionen):

```
reduced=Mod[LinearRecurrence[{0, 1, 1}, {3, 0, 2}, {2, 50}], Range[2, 50]-1]
```

Eine interessante Folge: Die Perrin-Folge

```
{0, 0, 0, 2, 0, 5, 0, 2, 3, 7, 0, 5, 0, 9, 8, 10, 0, 14, 0, 17, 10, 2, 0, 13, 5, 15, 12, 23, 0, 20, 0, 26, 25, 19, 12, 2, 0, 21, 3, 5, 0, 33, 0, 2, 32, 2, 0, 21, 7}
```

Holen wir uns die ‚0‘-Positionen:

```
Flatten[Position[reduced, 0]]  
{1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47}
```

Wer jetzt denkt, dass diese Methode immer Primzahlen liefert, täuscht sich leider. Das erste Gegenbeispiel $271441 = 521^2$ für eine Zusammengesetzte Zahl ist jedoch schon sehr groß und lässt auf sich warten (statt der roten 0 sollte eigentlich ein Wert > 0 stehen).

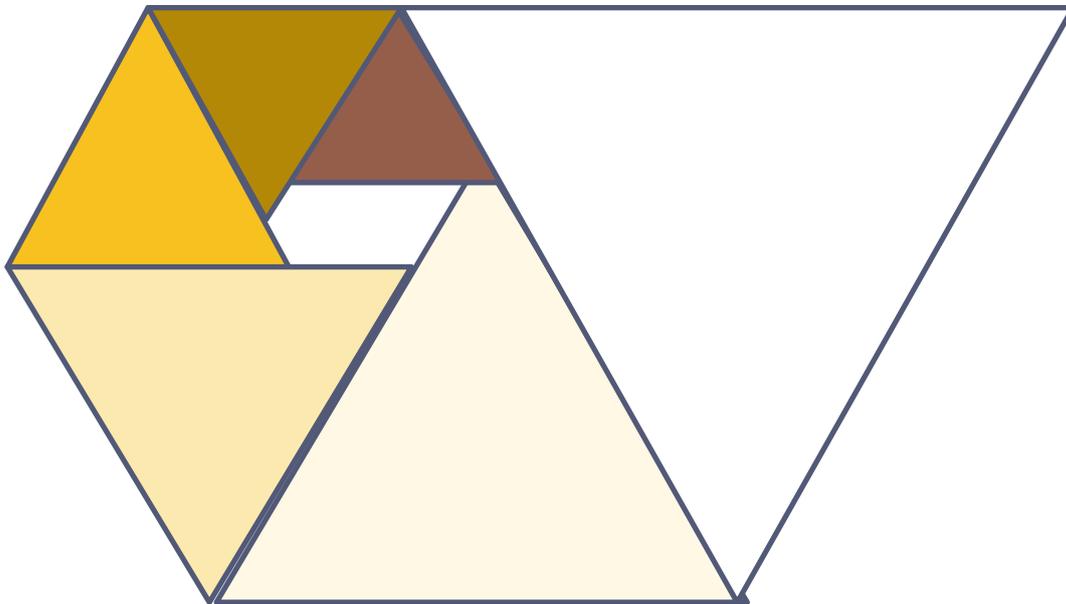
Mathematica:

```
Mod[LinearRecurrence[{0, 1, 1}, {3, 0, 2}, {271440, 271445}], {271439, 271440, 271441, 271442, 271443, 271444}]  
{107778, 199578, 0, 135723, 3, 112577}
```

Die ‚zusammengesetzten Zahlen‘ der Perrin-Folge, für die n ein Teiler von P_n ist, heißen **Perrin Pseudo-Primzahlen**. Z. Zt. sind 658 von Ihnen bekannt, die kleinste lautet $271441 = 521^2$ (Stand Dezember 2015).

John Grantham bewies 2010, dass es unendlich viele Perrin Pseudo-Primzahlen gibt.⁸³

Die Perrin-Serie ist eng mit der Folge von geometrisch wachsenden gleichseitigen Dreiecken verwandt:



Die Seitenlängen der gleichseitigen Dreiecke folgen der Perrin-Folge sowie einer zweiten Rekursions-Folge: $a(n) = a(n-1) + a(n-5)$

⁸³ <http://www.pseudoprime.com/pseudo3.pdf>

Das charakteristische Polynom der Perrin-Folge lautet somit:

$$(x^3 - x - 1) \text{ und } (x^5 - x^4 - 1)$$

Die Nullstelle des ersten Polynoms kann als verschachtelter unendlicher Ausdruck von 3. Wurzeln geschrieben werden:

$$r = \sqrt[3]{1 + \sqrt[3]{1 + \sqrt[3]{1 + \sqrt[3]{1 + \dots}}}} = 1.324717957244746$$

Die Perrin-Folge kann auch als geschlossener Ausdruck geschrieben werden: Mit

$$\Theta = \arccos\left(\frac{-r^{\frac{3}{2}}}{2}\right) \text{ gilt:}$$

$$s_n = r^n + 2 \frac{\cos(n\Theta)}{r^{\frac{n}{2}}}$$

In vielerlei Hinsicht erscheint die Perrin-Folge sogar noch interessanter als die Fibonacci-Folge. Sie besitzt noch eine Vielzahl von merkwürdigen Eigenschaften, auf die hier jedoch nicht eingegangen werden soll. Weitergehende Informationen findet der Leser im Internet.⁸⁴

20.4 WEITERE VERMUTUNGEN ÜBER PRIMZAHLEN

Die Goldbach-Vermutung:

Die Goldbach-Vermutung besagt, dass jede natürliche gerade Zahl $n > 2$ als Summe zweier Primzahlen geschrieben werden kann. Die Vermutung wurde numerisch für alle $n < 4 \cdot 10^{18}$ verifiziert. (Stand Okt. 2015). Die ‚erweiterte Goldbach-Vermutung‘ gibt eine Abschätzung für die Anzahl von Darstellungen R_g einer Zahl n als Summe von 2 Primzahlen:

$$R_g(n) \approx 2\Pi_2 \prod_{\substack{k=2 \\ p_k|n}} \frac{p_k - 1}{p_k - 2} \int_2^n \frac{dx}{(\ln x)^2} = 2\Pi_2 \prod_{\substack{k=2 \\ p_k|n}} \frac{p_k - 1}{p_k - 2} \left[\text{li}(x) - \frac{x}{\ln(x)} \right]_2^n \quad (161)$$

Mathematica-Code (Aus oeis.org):

```
a[n_] := Length @ Select[PowersRepresentations[2 n, 2, 1], (#[[1]] == 1 || PrimeQ[#[[1]])] && (#[[2]] == 1 || PrimeQ[#[[2]])] &]; Array[a, 98] (* Jean-François Alcover, Apr 11 2011 *)
```

⁸⁴ <http://www.mathpages.com/home/kmath345/kmath345.htm>

Die nächstmögliche Primzahl liegt bei $p+12$ beim Index 13: (Pentupel)

P	2	3	P	2	3	5	2	P	2	3	P	2	5	3	2	P	2	3	5	2	3	2	2	2	3	2	5	2	3	2	2	2	3	2	5	2	3	2	5
----------	---	---	----------	---	---	---	---	----------	---	---	----------	---	---	---	---	----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Für die nächste Siebung (Teilbarkeit durch 7) lässt sich noch keine eindeutige Bedingung finden (es ist nicht bestimmbar, welche der ersten 7 Stellen durch 7 teilbar sein muss, da wir noch Mehrdeutigkeit haben (Kandidaten: Stellen Nr. 4 und 5). **Also liegt die nächstmögliche Primzahl bei $p+18$ beim Index 19: (6-Tupel)**

P	2	3	P	2	3	5	2	P	2	3	P	2	5	3	2	P	2	3	5	2	3	2	P	2	3	5	2	3	2	2	2	3	2	5	2	3	2	5
----------	---	---	----------	---	---	---	---	----------	---	---	----------	---	---	---	---	----------	---	---	---	---	---	---	----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Von Gruppe $p-\frac{2}{3}-p-2-\frac{3}{5}-2-p$ muss eine durch 7 teilbar sein. Das kann nur die 4-Stelle mit der ,2' sein. Grund: Die 2-te Stelle $\frac{2}{3}$ ist ausgeschlossen, da dann auch die 9-te Stelle durch 7 teilbar sein müsste, die ist jedoch mit **p** belegt. Das gleiche gilt für die 5-te Stelle $\frac{2}{3}$, da dann auch die 19-te Stelle durch 7 teilbar sein müsste, die jedoch auch mit **p** belegt ist. Das gleiche gilt für die 6-te Stelle ,2', da dann auch die 13-te Stelle durch 7 teilbar sein müsste, die jedoch auch mit **p** belegt ist. Folglich bleibt als einzige Möglichkeit die 4-te Stelle mit der ,2'. Wir streichen alle durch 7 teilbaren Stellen:

P	2	3	P	2	7	3	5	2	P	2	3	P	2	5	3	7	2	P	2	3	5	2	3	7	P	2	3	5	2	3	7	2	2	2	3	5	2	7	2	5
----------	---	---	----------	---	---	---	---	---	----------	---	---	----------	---	---	---	---	---	----------	---	---	---	---	---	---	----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Die nächstmögliche Primzahl liegt bei $p+20$ beim Index 21: (7-Tupel)

P	2	3	P	2	7	3	5	2	P	2	3	P	2	5	3	7	2	P	2	3	5	2	3	7	P	2	3	5	2	3	7	2	2	2	3	5	2	7	2	5
----------	---	---	----------	---	---	---	---	---	----------	---	---	----------	---	---	---	---	---	----------	---	---	---	---	---	---	----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Für die nächste Siebung (Teilbarkeit durch 11) lässt sich noch keine eindeutige Bedingung finden (es ist nicht bestimmbar, welche der ersten 11 Stellen durch 11 teilbar sein muss, da wir noch Mehrdeutigkeit haben (Kandidaten: Stellen Nr. 4,5,6,11). **Also liegt die nächstmögliche Primzahl bei $p+26$ beim Index 27: (8-Tupel)**

P	2	3	P	2	7	3	5	2	P	2	3	P	2	5	3	7	2	P	2	3	5	2	3	7	P	2	3	5	2	3	7	2	2	2	3	5	2	7	2	5
----------	---	---	----------	---	---	---	---	---	----------	---	---	----------	---	---	---	---	---	----------	---	---	---	---	---	---	----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Für die nächste Siebung (Teilbarkeit durch 11) lässt sich noch keine eindeutige Bedingung finden (es ist nicht bestimmbar, welche der ersten 11 Stellen durch 11 teilbar sein muss, da wir noch Mehrdeutigkeit haben (Kandidaten: Stellen Nr. 4,6,11). **Also liegt die nächstmögliche Primzahl bei $p+30$ beim Index 31: (9-Tupel)**

20.7 WEITERE ABBILDUNGEN ZU RG-FOLGEN

Hier ein paar Beispiele zu RG-Folgen vom Typ4 (EGOCRON4):

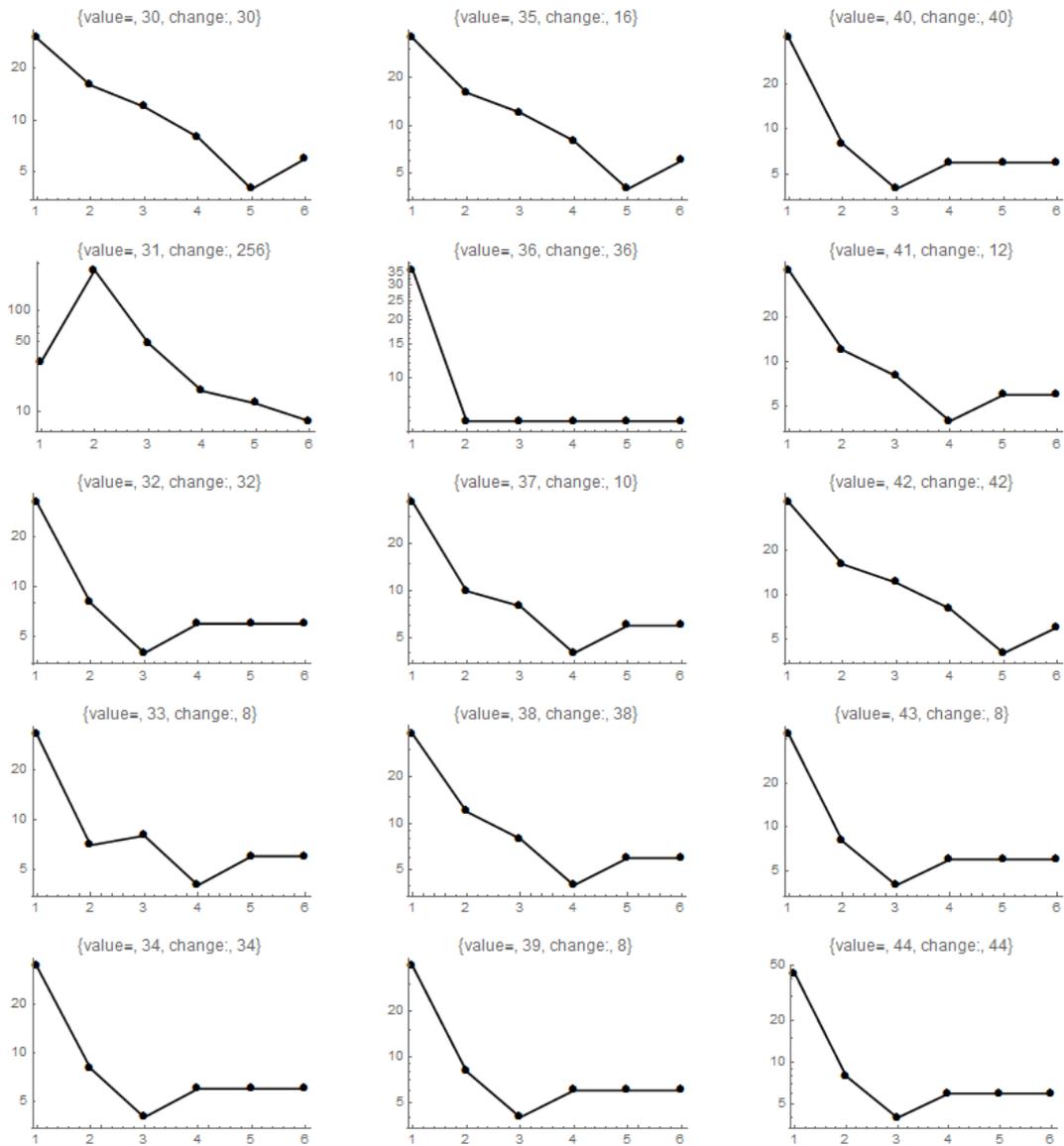


Abbildung 141: RG-Folgen von Typ4-EGOCRONs in Richtung positiver Indices (Werte 30-44)

Weitere Abbildungen zu RG-Folgen

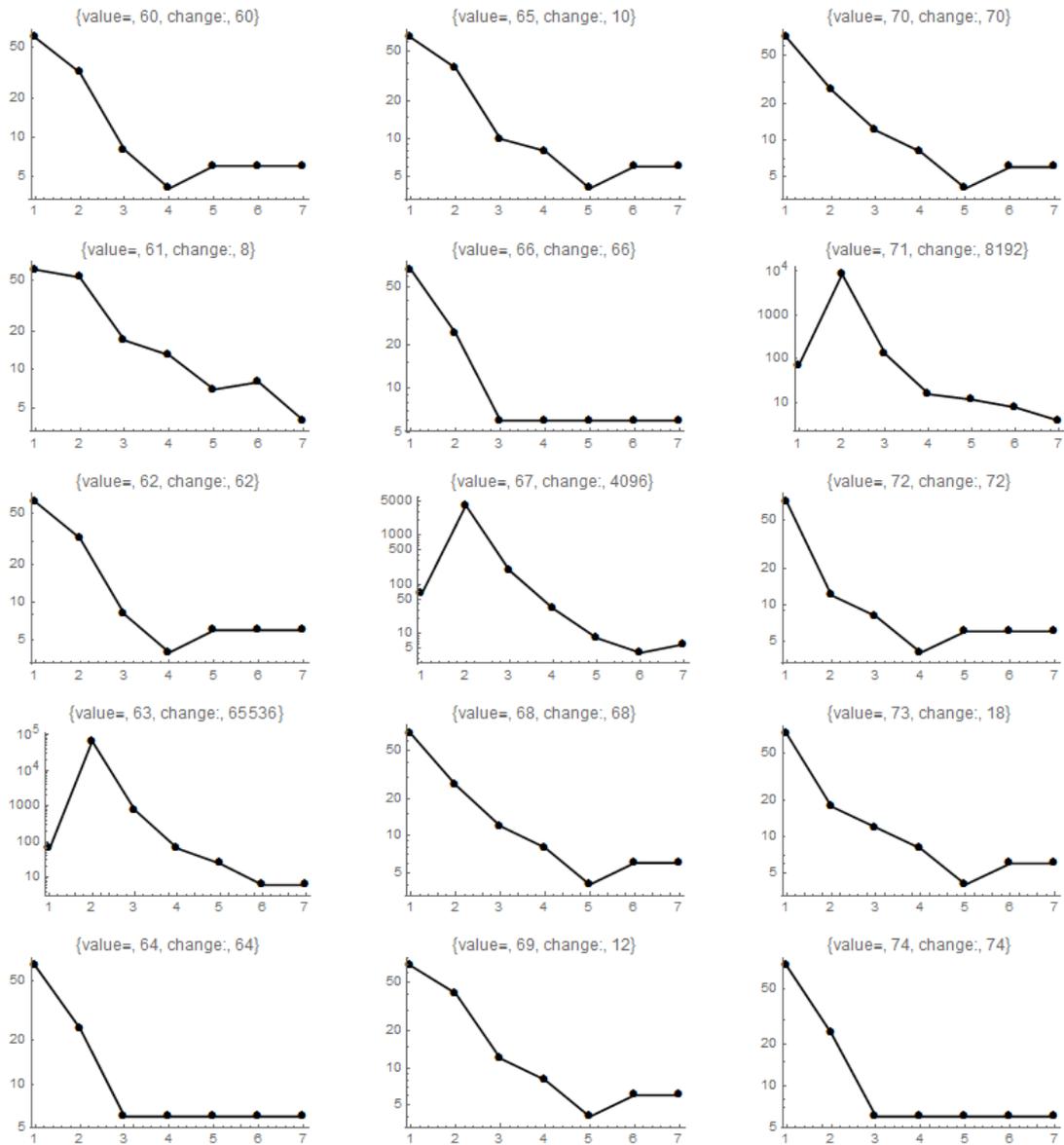


Abbildung 142: RG-Folgen von Typ4-EGOCRONs in Richtung positiver Indices (Werte 60-74)

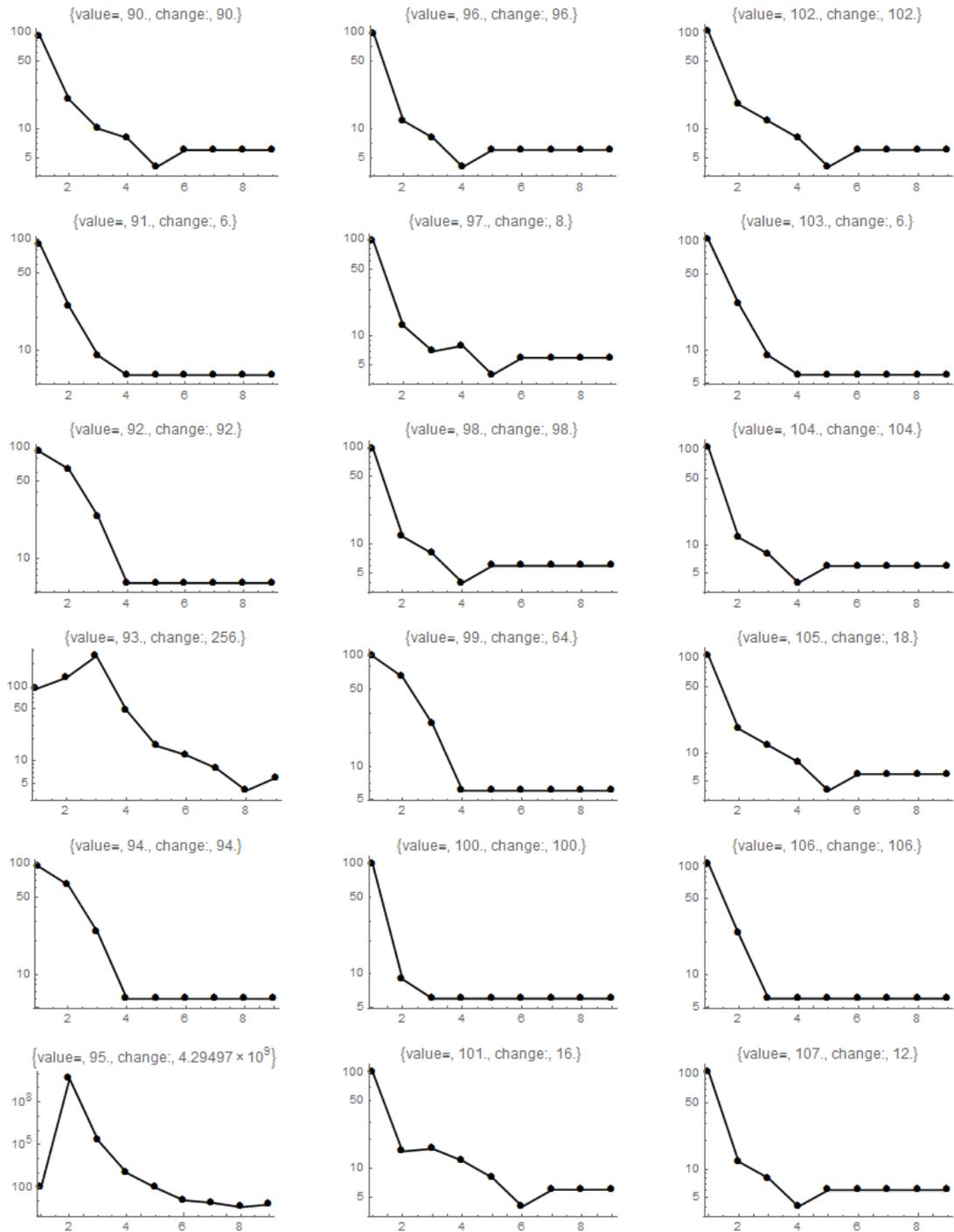


Abbildung 143: RG-Folgen von Typ4-EGOCRONs in Richtung positiver Indices (Werte 90-107)

20.8 VIRTUELLE OCRONS

Wenn wir uns OCRONS vom Typ 4 ansehen, dann sehen wir, dass es OCRONS gibt, in denen das Symbol ‚*‘ nicht vorkommt, d.h. die nur aus den Symbolen ‚2‘, ‚P‘ und ‚^‘ bestehen. Dies sind Primzahlen oder Potenzen von Primzahlen. Für die folgenden Betrachtungen brauchen wir auch die zur Basis 2 exponierten OCRONS. Wie der Leser inzwischen weiß, entsteht ein zur Basis 2 exponiertes OCRON durch Voranstellen einer ‚2‘ und Anhängen von ‚^‘ (unten in grüner Farbe). Hier ein paar Beispiele von OCRONS und deren exponierten Versionen (das ‚unexponierte‘ in schwarzer oder blauer Farbe):

2 (4) :	22^
3 (8) :	22P^
4 (16) :	222^^
5 (32) :	22PP^
6 (64) :	22P2*^, 22P^2^
7 (128) :	222^P^
8 (256) :	222P^^
9 (512) :	22P2^^
10 (1024) :	22PP2*^, 22PP^2^
11 (2048) :	22PPP^
12 (4096) :	22P22^*^, 222^^2P^, 22^2P^2^
13 (8192) :	22P2*P^ (keine ‚*‘-freie OCRON-Darstellungen)
13 (2 ⁸¹⁹²) :	222P2*P^^, 222PP^^222P^^, 222^^22PPP^^
14 (16384) :	222^P2*^, 22^22^P^
15 (32768) :	22PP2P*^, 22P^2PP^
16 (65536) :	2222^^^
17 (131072) :	222^PP^
18 (262144) :	22P2^2*^, 22^2P2^^
19 (524288) :	222P^P^

Die in blau dargestellten OCRONS sind ‚*‘-freie OCRONS, die nach den OCRON-Regeln einfach abgearbeitet werden können. Die in rot dargestellten OCRONS sind ‚nicht wohlgeformt‘, d.h. sie ergeben, als OCRONS vom Typ 4 interpretiert, keinen Sinn. Werden sie jedoch zur Basis 2 exponiert (oder im Fall der Zahl 13 sogar 2-mal exponiert), stellen sie wohlgeformte, interpretierbare OCRONS dar. Die roten, unexponierten ‚OCRONS‘ wollen wir ‚virtuelle OCRONS‘ nennen, da sie erst einen Sinn ergeben wenn sie ein oder mehrmalig zur Basis 2 exponiert werden. Um den Wert eines virtuellen OCRONS zu erhalten, muss der Zahlenwert des n mal exponierten OCRONS wieder n mal mit dem 2-er Logarithmus („de“-)logarithmiert werden.

Gleichwertige, ‚*‘-freie OCRONS zu finden stellt sich als nicht-triviale Aufgabe dar, da die Menge der zu diesem OCRON gehörenden entarteten OCRONS nach ‚*‘-freien OCRONS durchsucht werden muss.

Folgender Satz ist eine noch unbewiesene Vermutung:

Jede OCRON-Typ4-Darstellung einer natürlichen Zahl $n \geq 2$ ist entweder ,*-frei oder es gibt in den höheren Exponierungen des OCRONs gleichwertige, entartete ,*-freie OCRON-Darstellungen.

Falls dieser Satz richtig ist, hätten wir eine OCRON-Representation aller natürlichen Zahlen ≥ 2 , die nur aus den OCRON-Symbolen „2“, „P“, und „^“ besteht. Dies wäre eine Beschreibung, die ohne den „multiplikativen“ Operator „*“ auskommt.

Virtuelle OCRONs haben interessante Eigenschaften. Hier zunächst eine Tabelle mit einigen entarteten, virtuellen OCRONs im Bereich von 2 bis 40:

Tabelle 28: Entartete virtuelle OCRONs. Primzahlen und -Potenzen in rot (Ordnung: exp)

N	GC(P=1,2=2,^=0), OCRON, (exp)	n	GC(P=1,2=2,^=0), OCRON,(exp)
1	-	21	1774, $2P^22^P$ (1) 1978, 22^P^2P (1) 13834, 2^{2222P^P} (2)
2	2, 2 (0)	22	553, 2^2PPP (1) 605, $2PPP^2$ (1)
3	7, $2P$ (0) 1484, 2^{2222} (3) 2375, P^{2222} (3)	23	208, $2P2^P$ (0) 164832, $22P^P^{2222}$ (2)
4	24, 22^2 (0) 20, 2^2 (1) 170, 2^{22} (2) 13928, 2^P^{22P2} (3) 15388, $2P^{222P}$ (3) 24569, P^2^{2222} (3) 25541, $P^{22^{2222}}$ (3) 28619, PPP^{2222} (3) 40049, 2^{2222P^22} (3) 40903, $2^{22^{222P}}$ (3)	24	1776, $2P^22P^2$ (1) 2032, $22P^{22P}$ (1) 4921, $2^2^{2^2P}$ (1) 4925, $2^2^{2P^2}$ (1) 4961, $2^2P^2^2$ (1) 4965, $2^2P^{22^2}$ (1) 5029, $2^{22^{22P}}$ (1) 5285, $2P^2^{2^2}$ (1) 5289, $2P^2^{22^2}$ (1) 5321, $2P^{22^{2^2}}$ (1)
5	22, $2PP$ (0) 511, 2^{22P} (2) 575, $2P^{22}$ (2) 1520, 2^{2^22} (2) 8980, PP^{22P2P} (3) 33611, $P2^P^{22P2}$ (3) 41783, 2^P^{22PP2} (3)	25	204, $2PP2^2$ (0) 1804, $2PP^2PP$ (1) 13816, 2^{222P2^P} (2)
6	61, 2^2P (1) 65, $2P^2$ (1) 1532, 2^{222^2} (2) 1536, 2^{2222^2} (2) 1628, 2^2^{222} (2) 1726, $2P^{22P}$ (2) 1952, 22^{222} (2) 4561, 2^{2^22P} (2) 5161, $2P^{2^22}$ (2)	26	46621, $2P^{22P2^P}$ (2)
7	73, 22^P (0) 1534, 2^{222PP} (2) 1790, $2PP^{22}$ (2) 4885, 2^2^{222P} (2) 5177, $2P^{22^2}$ (2) 5181, $2P^{2222^2}$ (2) 5857, 22^{222P} (2) 7252, $P^{222PP2P}$ (3) 13682, $2^{2^222^2}$ (2) 13686, $2^{2^2222^2}$ (2)	27	210, $2P2P^2$ (0) 1770, $2P^2P2^2$ (1) 1870, $2P2^{22P}$ (1) 5299, $2P^2P^2P$ (1) 13812, 2^{222PP2^2} (2)
8	75, $22P^2$ (0) 182, 2^2^2 (1) 186, 2^{22^2} (1)	28	4933, $2^2^{22^P}$ (1) 5033, $2^{22^2P^2}$ (1) 5905, 22^{22^2P} (1)

Virtuelle OCRONS

	218, 22^2 (1) 4597, 2^22^2P (2) 4601, 2^22P^2 (2) 4949, 2^2P^22 (2) 5179, 2P^22PP (2) 5273, 2P^2^22 (2) 5371, 2PP^22P (2)		5933, 22^P^2^2 (1) 5937, 22^P^22^ (1)
9	69, 2P2^ (0) 196, 2P^2P (1) 4609, 2^222^P (2) 5921, 22^P^22 (2) 14656, 2^2^22PP (2) 14848, 2^2P^22P (2) 15532, 2P^22^2P (2) 15536, 2P^22P^2 (2) 15820, 2P^2^22P (2) 16112, 2PP^22^2 (2)	29	13818, 2^222P2P^ (2) 17018, 2P2P^22 (2)
10	184, 2^2PP (1) 200, 2PP^2 (1) 4611, 2^222P^ (2) 6083, 22P^22 (2) 13790, 2^222^2^2 (2) 13794, 2^222^22^ (2) 13826, 2^2222^2 (2) 14750, 2^2^2^22 (2) 15074, 2^22^22 (2) 15074, 2^22^22 (2)	30	4963, 2^2P^2PP (1) 4975, 2^2PP^2P (1) 5287, 2P^2^2PP (1) 5303, 2P^2PP^2 (1) 5407, 2PP^2^2P (1) 5411, 2PP^2P^2 (1)
11	67, 2PPP (0) 4605, 2^222P2^ (2) 5597, 2P2^22 (2) 13804, 2^222P^2P (2) 15546, 2P^222P^ (2) 15884, 2P^2P^22 (2) 18250, 22P^22P (2) 41061, 2^2^2^22P^ (2) 43975, 2^2^222^P (2)	31	202, 2PPPP (0) 153168, 2P2P^222^ (2)
12	547, 2^2^2P (1) 551, 2^2P^2 (1) 587, 2P^2^2 (1) 591, 2P^22^ (1) 655, 22^2P (1) 13792, 2^222^2PP (2) 13808, 2^222PP^2 (2) 14912, 2^2PP^22 (2)	32	228, 22PP^ (0) 1680, 2^222^2 (1) 1692, 2^222^2 (1) 2000, 22^2^2 (1) 2108, 222^2 (1) 4935, 2^2^22P^ (1) 5051, 2^22P^2 (1) 5907, 22^22P^ (1) 6095, 22P^2^2 (1) 6099, 22P^22^ (1)
13	4603, 2^222PPP (2) 5435, 2PPP^22 (2) 48351, 2PP^222P^ (2) 52719, 22^2^22P2^ (2)	33	1768, 2P^2PPP (1) 1816, 2PPP^2P (1) 13810, 2^222PPP (2)
14	559, 2^22^P (1) 659, 22^P^2 (1)	34	1678, 2^22^PP (1) 1982, 22^PP^2 (1) 13836, 2^2222PP^ (2)
15	589, 2P^2PP (1) 601, 2PP^2P (1)	35	5419, 2PP^22^P (1) 5935, 22^P^2PP (1)
16	222, 22^2^ (0) 234, 222^ (0) 561, 2^22P^ (1) 677, 22P^2 (1) 1640, 2^2^2^2 (1) 1644, 2^2^22^ (1) 1676, 2^22^2 (1) 1964, 22^2^2 (1) 1968, 22^22^ (1)	36	4929, 2^2^2P2^ (1) 4997, 2^2P2^2 (1) 5609, 2P2^2^2 (1) 5613, 2P2^22^ (1) 5901, 22^2P2^ (1) 14776, 2^2^2P^2P (1) 14884, 2^2P^2^2P (1) 14888, 2^2P^2P^2 (1)
17	220, 22^PP (0) 41413, 2^22P^2PP (2)	37	166288, 22PP^22PP (2)

	41425, $2^{22}2^{22}2^{22}2^{22}$ (2)		
18	555, $2^22^22^2$ (1) 623, $2^22^22^2$ (1) 1654, $2^22^22^22^2$ (1) 1762, $2^22^22^22^2$ (1) 1766, $2^22^22^22^2$ (1) 13830, $2^{22}2^{22}2^{22}$ (2) 13842, $2^{22}2^{22}2^{22}$ (2)	38	1684, $2^22^22^22^2$ (1) 2036, $2^22^22^22^2$ (1)
19	226, $2^22^22^2$ (0) 13828, $2^{22}2^{22}2^{22}$ (2)	39	498871, $2^{22}2^{22}2^{22}2^{22}2^{22}$ (2)
20	1642, $2^22^22^22^2$ (1) 1658, $2^22^22^22^2$ (1) 1802, $2^{22}2^{22}2^{22}$ (1) 1806, $2^{22}2^{22}2^{22}$ (1) 1966, $2^{22}2^{22}2^{22}$ (1)	40	5421, $2^{22}2^{22}2^{22}$ (1) 6097, $2^{22}2^{22}2^{22}$ (1) 14764, $2^22^22^22^22^2$ (1) 14780, $2^22^22^22^22^2$ (1) 14924, $2^22^22^22^22^2$ (1) 14928, $2^22^22^22^22^2$ (1)

Wir benennen virtuelle OCRONs mit einer zugehörigen Anzahl n von Exponierungen als ‚virtuelle‘ OCRONs der Ordnung n . Aus den Potenzrechengesetzen

$(2^a)^b = 2^{a*b}$ sowie $(2^{2^a})^{2^b} = 2^{2^{a+b}}$ lassen sich folgende Regeln für virtuelle OCRONs der Ordnung 1 und 2 feststellen:

Eine Zahl n , die sich als ‚*-freies OCRON darstellen lässt, ist (gleichzeitig) ein virtuelles OCRON der Ordnung 0. Dies trifft zu für alle Primzahlen und Primzahl-Potenzen, falls die Primzahl eine ‚*-freie Darstellung besitzt.

Jede zusammengesetzte Zahl, die sich als Produkt von unterschiedlichen ‚*-freien Faktoren schreiben lässt, lässt sich als virtuelles OCRON der Ordnung 1 und 2 darstellen, jedoch nicht als virtuelles OCRON der Ordnung 0.

Primzahlen entsprechen entweder virtuellen OCRONs der Ordnung 0 oder 2.

Primzahl-Potenzen (mit Potenzen ≥ 2) können als virtuelle OCRONs der Ordnungen 0, 1 und 2 dargestellt werden.

Virtuelle OCRONs der Ordnung 0,1 oder 2 beginnen immer mit dem Symbol ‚2‘.

Ab der Ordnung 3 können virtuelle OCRONs auch mit dem Symbol ‚P‘ beginnen.

Satz: Jede natürliche Zahl lässt sich als virtuelles OCRON der Ordnung 0, 1 oder 2 darstellen.

In ‚mathematische Alltagssprache‘ übersetzt lautet dieser Satz:

Jede natürliche Zahl $n > 1$ lässt sich nur durch Verwendung der Zahl 2, der Funktionen Prime() und Log() (zur Basis 2) sowie der arithmetischen Potenzierungs-Operation darstellen.

Man beachte, dass die arithmetischen Operationen ‚*‘ und ‚+‘ nicht benötigt werden! Der Beweis ist klar, da sich jede natürliche Zahl entweder aus einem Produkt von 2 oder mehreren ‚*-freien Faktoren, oder aus einer Summe von 2 oder mehreren ‚*-freien Summanden darstellen lässt, wobei wir unter ‚*-freien Faktoren bzw. Summanden solche verstehen wollen, die (‚*-freie) Primzahlen oder Primzahlpotenzen sind.

20.9 WEITERE UNGELÖSTE MATHEMATISCHE PROBLEME

20.9.1 EUKLID-MULLIN SEQUENZ

Diese Sequenz ist sehr einfach definiert:

Sei $a_1 = 2$, dann ist a_n der kleinste Primfaktor der Zerlegung:

$$\prod_{i=1}^{n-1} a_i + 1$$

Die ersten Werte der Euklid-Mullin-Sequenz lauten:

2, 3, 7, 43, 13, 53, 5, 6221671, 38709183810571, 139, 2801, 11, 17,
5471, 52662739, 23003, 30693651606209, 37, 1741, 1313797957, 887, 71,
7127, 109, 23, 97, 159227, 643679794963466223081509857, 103,
1079990819, 9539, 3143065813, 29, 3847, 89, 19, 577, 223, 139703, 457,
9649, 61, 4357,
87991098722552272708281251793312351581099392851768893748012603709343,
107, 127, 3313,
2274326891085895327549849150757748483866714395682604207544149407807612
45893, 59, 31, 211

Mathematica:

```
f[1]=2;f[n_]:=f[n]=FactorInteger[Product[f[i],  
{i,1,n-1}]+1][[1,1]];Table[f[n],{n,1,43}]
```

Es ist unbekannt, ob die Euklid-Mullin-Sequenz alle Primzahlen durchläuft. Es ist weiterhin unbekannt, ob das Problem, herauszufinden ob eine gegebene Primzahl in der Sequenz enthalten ist, zur Gruppe der berechenbaren⁸⁵ Probleme gehört. So ist es z. B. noch ungeklärt, ob die Zahl 41 zur Euklid-Mullin-Sequenz gehört.

⁸⁵ <https://de.wikipedia.org/wiki/Berechenbarkeit>

- Zyklisch (nicht 0): Die bisher bekannten Zyklen besitzen folgende Längen: 1,2,4,5,6,8,9,28 (Stand: Juni 2016). Zahlen mit Zyklus 1 sind (die bereits in 4.5 besprochenen) perfekten Zahlen. Solche mit Zyklus 2 heißen ‚befreundete‘ (‚amicable‘) Zahlen. Zahlen in den höheren Zyklen nennt man ‚gesellige‘ (‚sociable‘) Zahlen.
- ‚Open End‘ (OE). Manche Sequenzen wachsen ins Unendliche ohne dass wieder ein ‚Abstieg‘ erkennbar wäre.

Wenn wir die Sequenzen, die mit 0 enden, auch zur Klasse der zyklisch endenden Sequenzen zählen, gibt es im Grunde genommen nur zwei Typen, nämlich die zyklisch endenden und die nicht endenden Sequenzen.

Die Vermutung von Catalan lautet nun, dass jede Aliquot Sequenz irgendwann zyklisch endet, dass somit gar keine OE (nicht endende) Sequenzen existieren!

Unterhalb von 1000 gibt es z. Zt. 5 OE-Sequenzen und weitere 7 Sequenzen, die entweder Startwerte auf einer dieser 5 Sequenzen haben oder auf einer dieser 5 Sequenzen landen (Stand Juni 2016). Dies sind die sogenannten ‚Lehmer-Five‘. Hier die Startwerte der 12 Sequenzen unterhalb von 1000, deren ‚Schicksal‘ ungewiss ist:

276 (306,396, 696)

552 (888)

564 (780)

660 (828, 996)

966

Mit wachsender Rechenleistung der Computer ist in den letzten Jahren die Anzahl der OE-Sequenzen reduziert worden. Manche Sequenzen stoßen in schwindelerregend hohe Zahlenbereiche vor, bevor sie sich entschließen, wieder ‚abzusteigen‘ und ganz normal bei einer Primzahl zu enden. Jede natürliche Zahl, als Startwert genommen, besitzt somit eine ganz eigene, private Aliquot-Sequenz. Diese Sequenzen können höchst unterschiedlich aussehen. Sie können aus einer Zahl bestehen (wenn als Startwert eine vollkommene Zahl genommen wird), sie können jedoch auch aus Tausenden von Werten bestehen, bevor die Sequenz in einem Zyklus endet. In diesen Fällen ähnelt der Graph der entsprechenden Sequenz mehr einem Börsenkurs als einer zahlentheoretischen Funktion. Die längsten, bisher berechneten Sequenzen sind alle OE-Sequenzen, mit Längen von Tausenden von Folgengliedern. Die längste, derzeit berechnete Aliquot OE-Sequenz hat den Startwert 933436 und ist bis zum Term 12516 berechnet worden (Stand: Juni 2016).⁸⁶ Die längsten bisher gefundenen endenden Sequenzen haben Längen von über 70000 Gliedern. Die größten dabei erreichten Werte von Folgengliedern liegen über 10^{120} (gleiche Quelle). Bei OE-Sequenzen gibt es ‚Abstiege‘ von über 100 Zehnerpotenzen, bevor die Sequenz wieder ‚in die Unendlichkeit‘ hinaufstrebt. Auf der anderen Seite gibt es ‚Aufstiege‘ bis zu 120 Zehnerpotenzen, bevor manche Sequenzen wieder absteigen und bei einer Primzahl enden. Aus den bisher empirisch gefundenen Daten lässt sich abschätzen, dass z. Zt. etwa 1 % aller Zahlen OE-Sequenzen (‚Open End‘) besitzen.

Hinweis: Die meisten Informationen in diesem Kapitel stammen von folgenden Internetseiten:

⁸⁶ <http://aliquot.de/aliquote.htm#records>

<http://www.aliquot.de>, <http://factordb.com> (Markus Tervooren),
<http://christophe.clavier.free.fr/Aliquot/site/Aliquot.html>

Hier ein paar Graphen von Aliquot-Sequenzen. Zunächst die ‚Lehmer-Five‘ (Open-End Sequenzen mit Startwerten unterhalb von 1000):

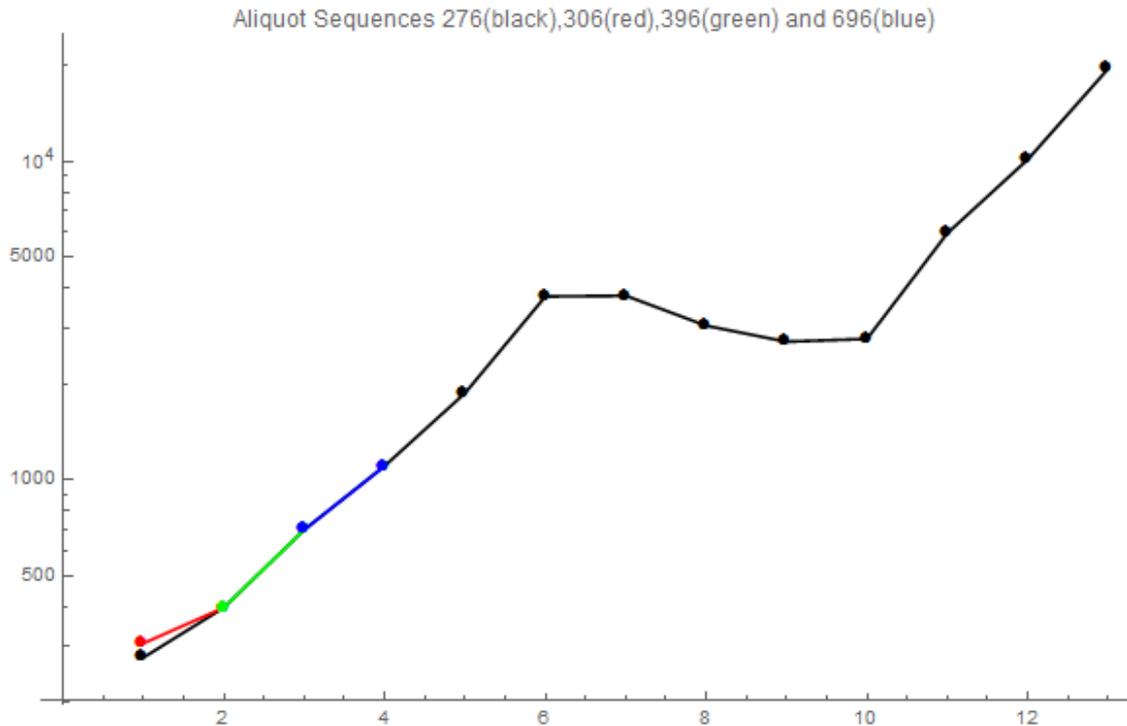


Abbildung 144: Die ersten 12 Werte der Aliquot Sequenzen 276,306,396,696. Ab dem 3. Wert sind die Sequenzen identisch

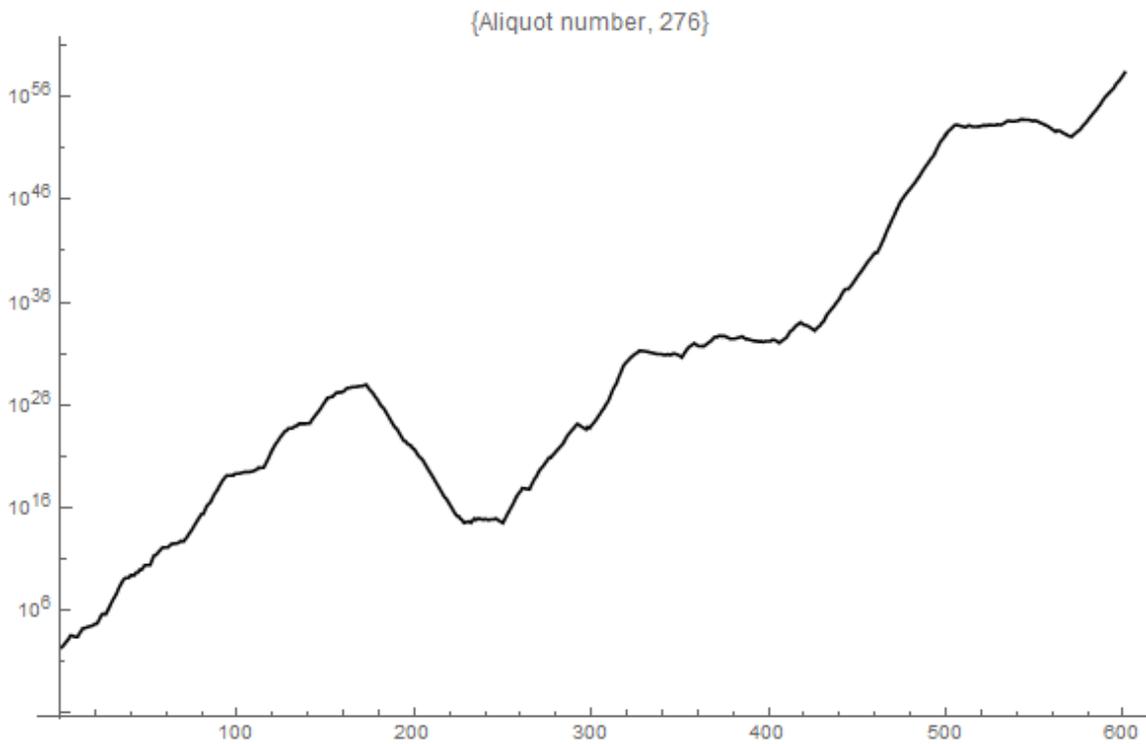


Abbildung 145: Aliquot-Sequenz 276 (OE, die ersten 600 Werte)

Weitere ungelöste mathematische Probleme

Mathematica:

(*Aliquot 276 OE*)

```
n=276;value=n;  
table=Table[value=DivisorSigma[1,value]-  
value,{i,1,600}];table=Prepend[table,n];  
ListLogPlot[table,PlotStyle->Black,Joined->True,ImageSize-  
>Large,PlotLabel->{"Aliquot number",n}]
```

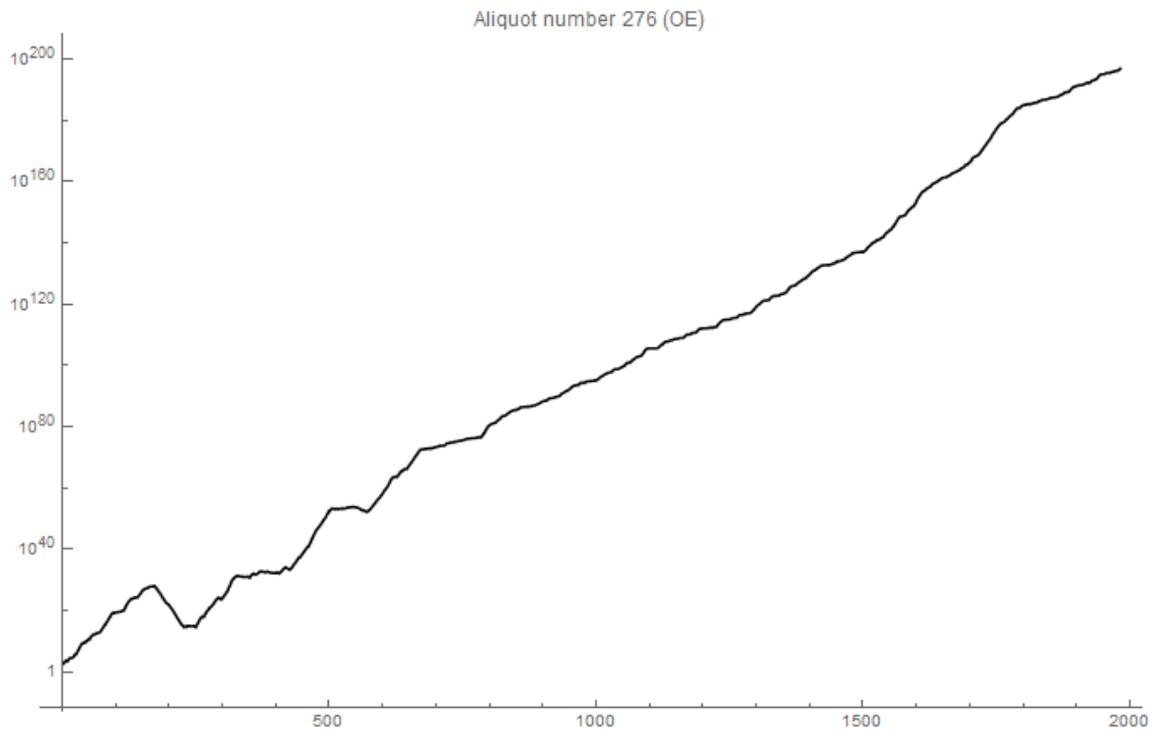


Abbildung 146: Aliquot-Sequenz 276 (OE, die ersten 1981 Werte)

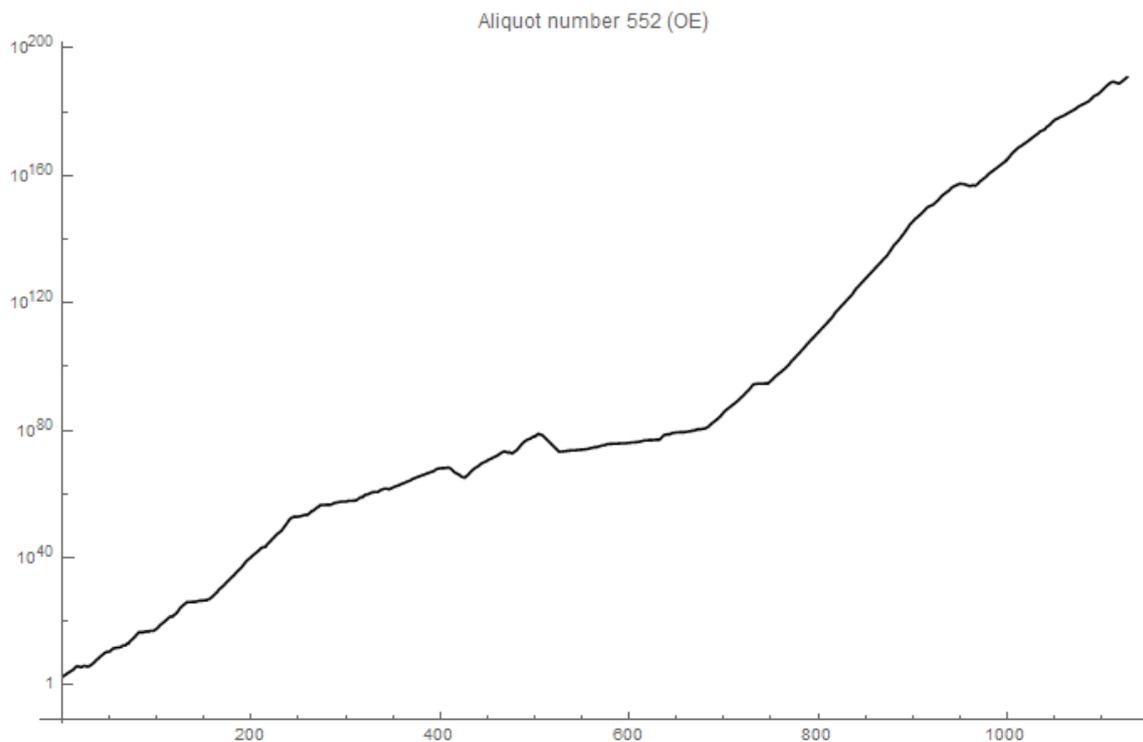


Abbildung 147: Aliquot-Sequenz 552 (OE, die ersten 1126 Werte)

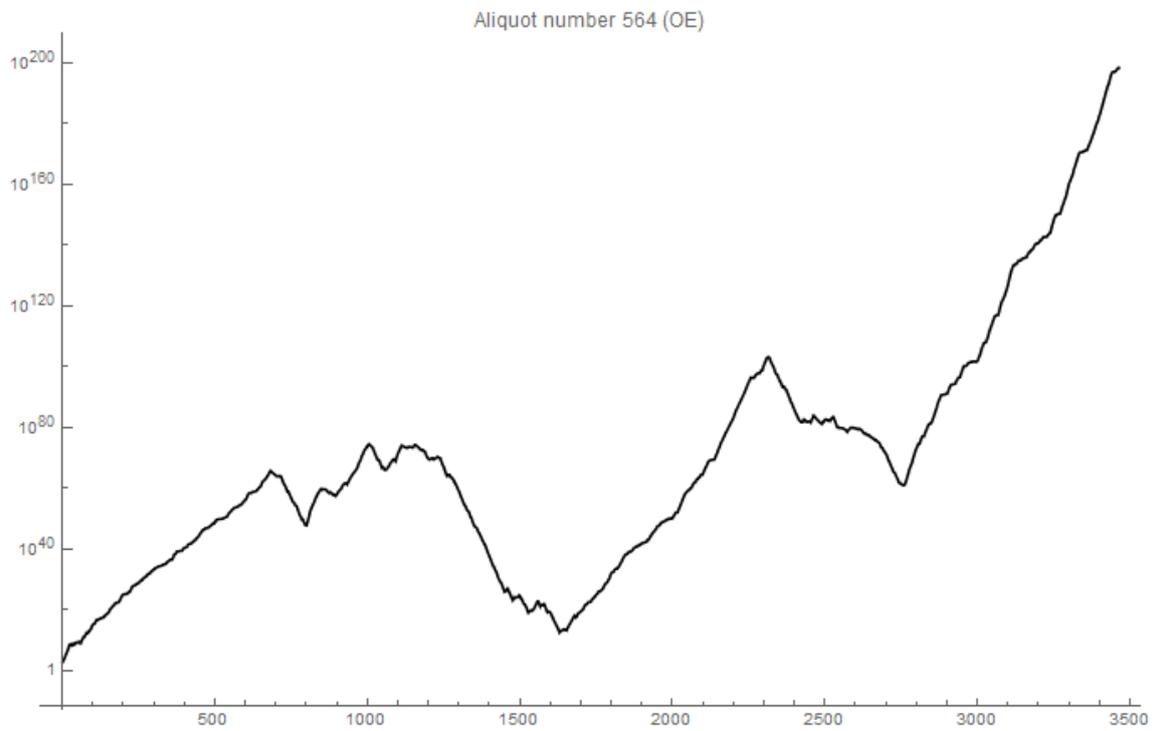


Abbildung 148: Aliquot-Sequenz 564 (OE, die ersten 3463 Werte)

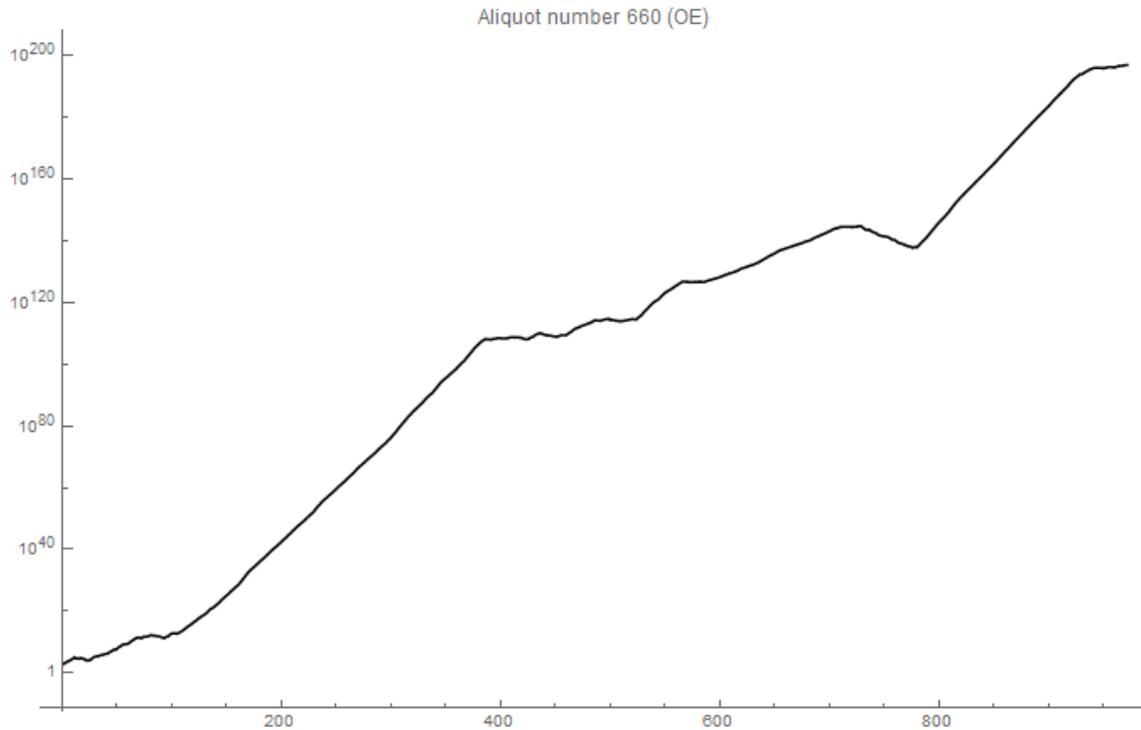


Abbildung 149: Aliquot-Sequenz 660 (OE, die ersten 971 Werte)

Weitere ungelöste mathematische Probleme

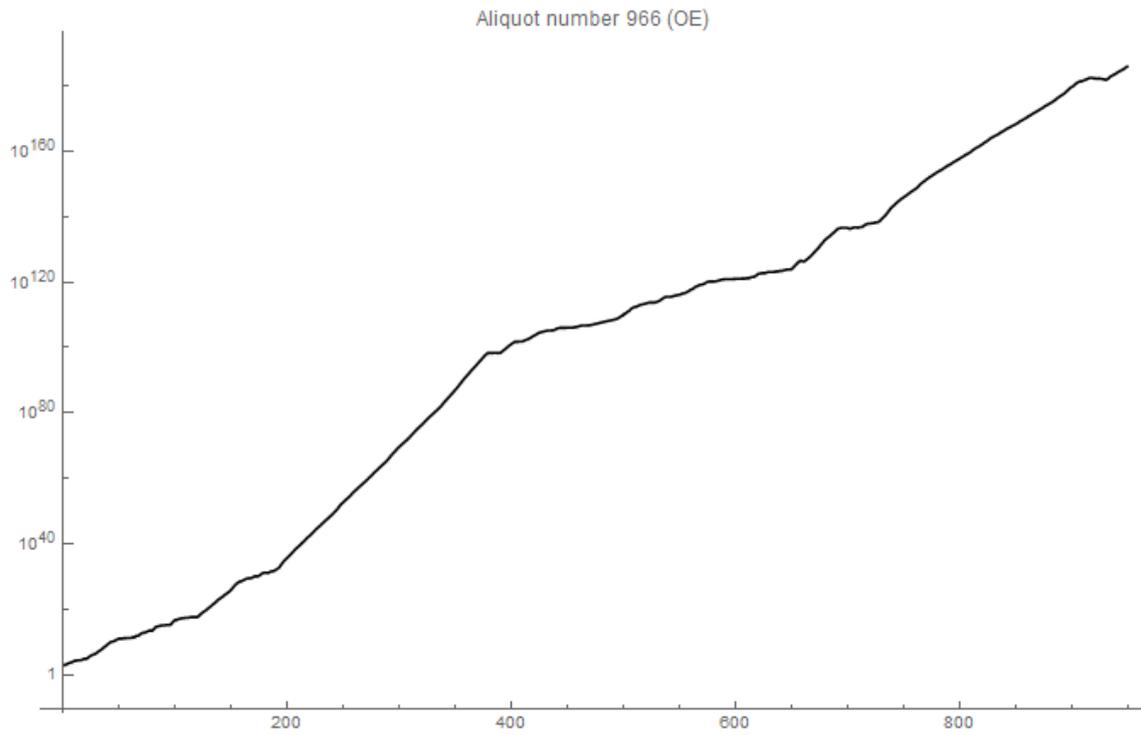


Abbildung 150: Aliquot-Sequenz 966 (OE, die ersten 948 Werte)

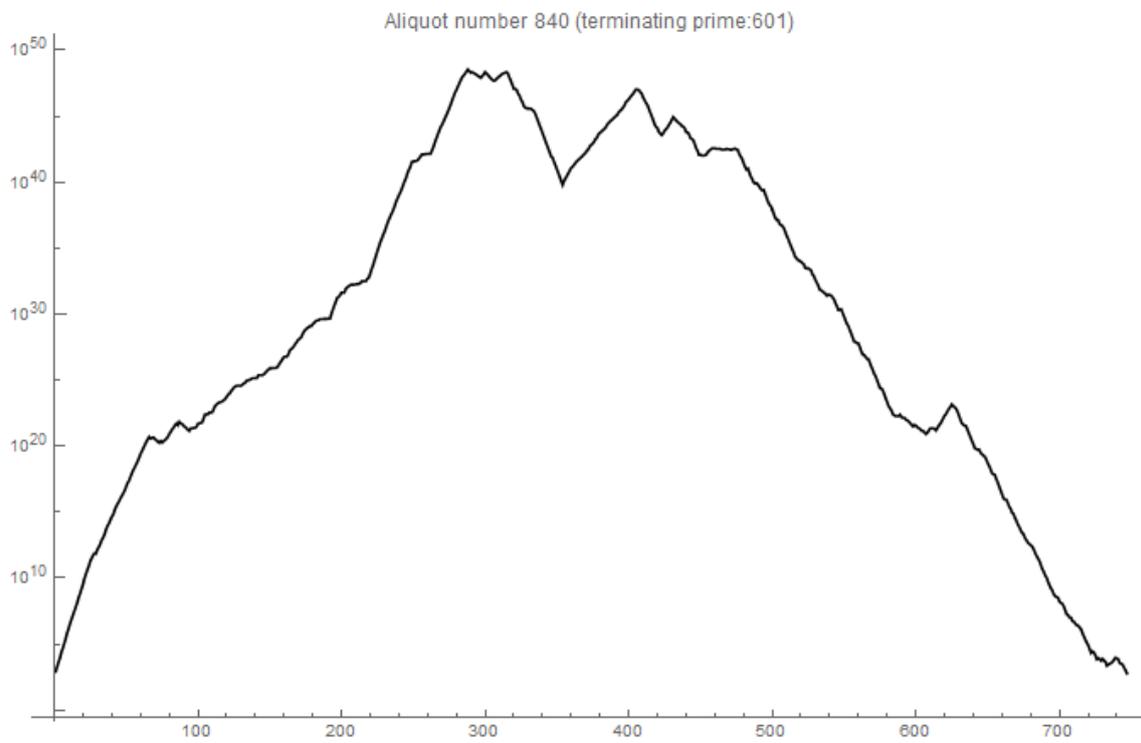


Abbildung 151: Aliquot-Sequenz 840 (Ende bei 601, 746 Werte)

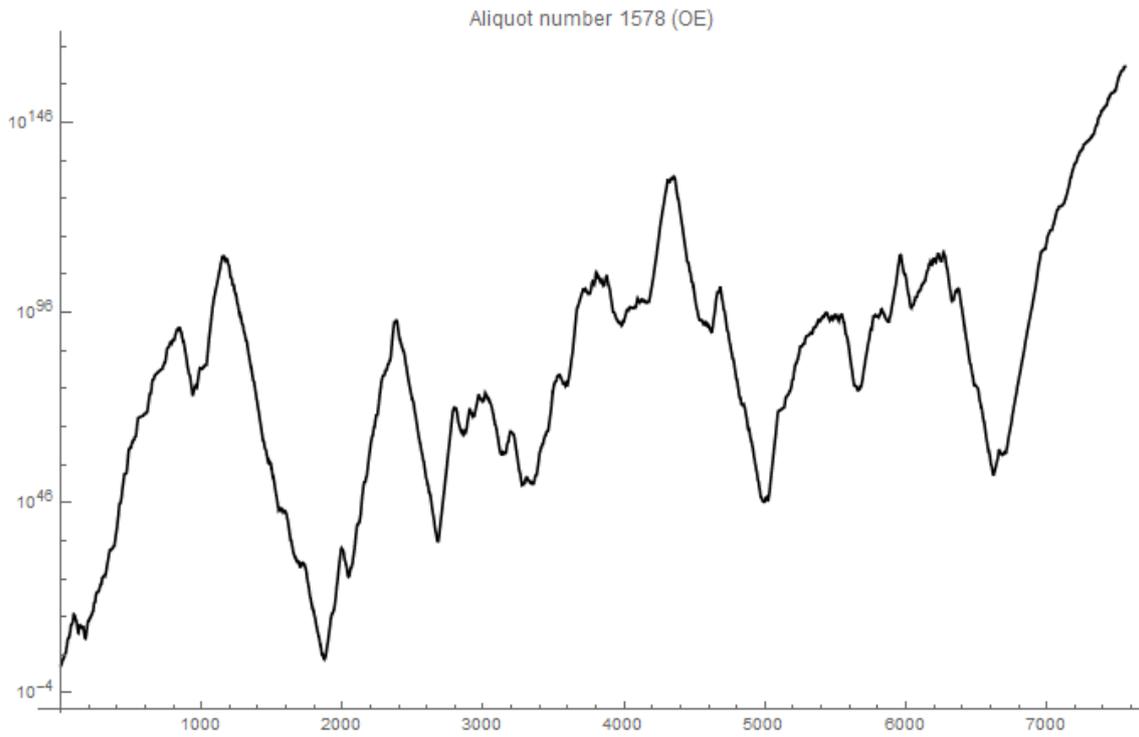


Abbildung 152 : Aliquot-Sequenz 1578 (OE, die ersten 7555 Werte)

Und hier noch ein paar Plots von endenden Aliquot-Sequenzen:

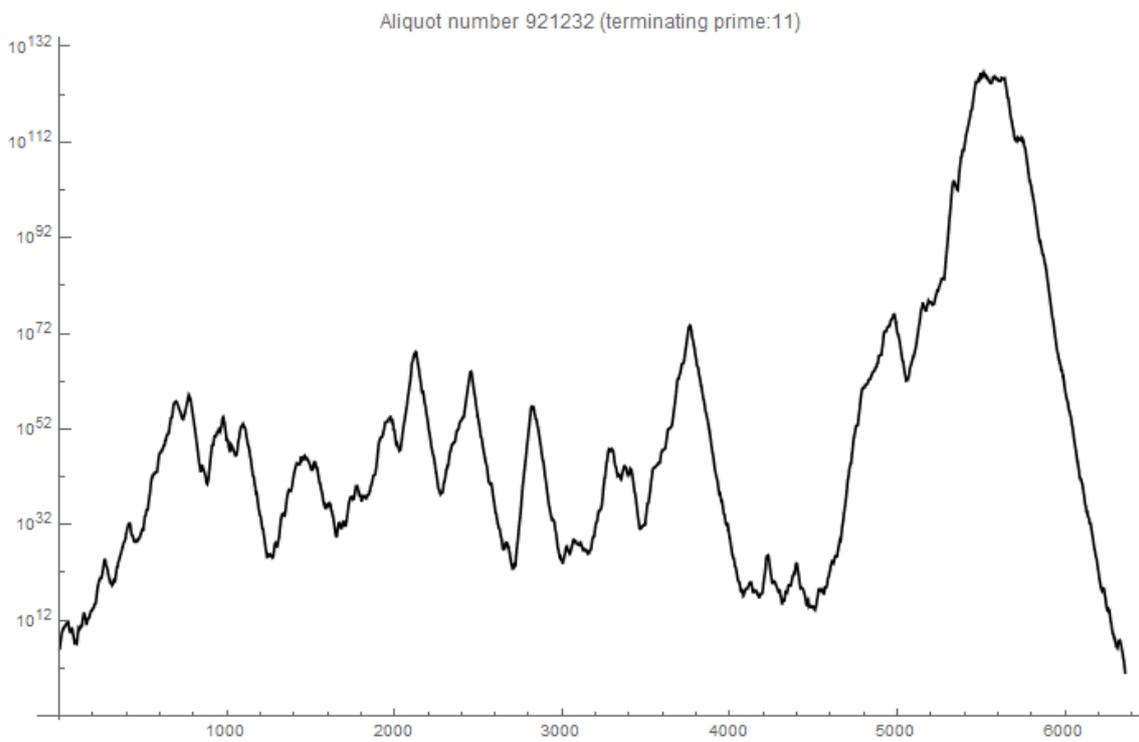


Abbildung 153: Aliquot-Sequenz 921232 (Ende bei 11, 6358 Werte)

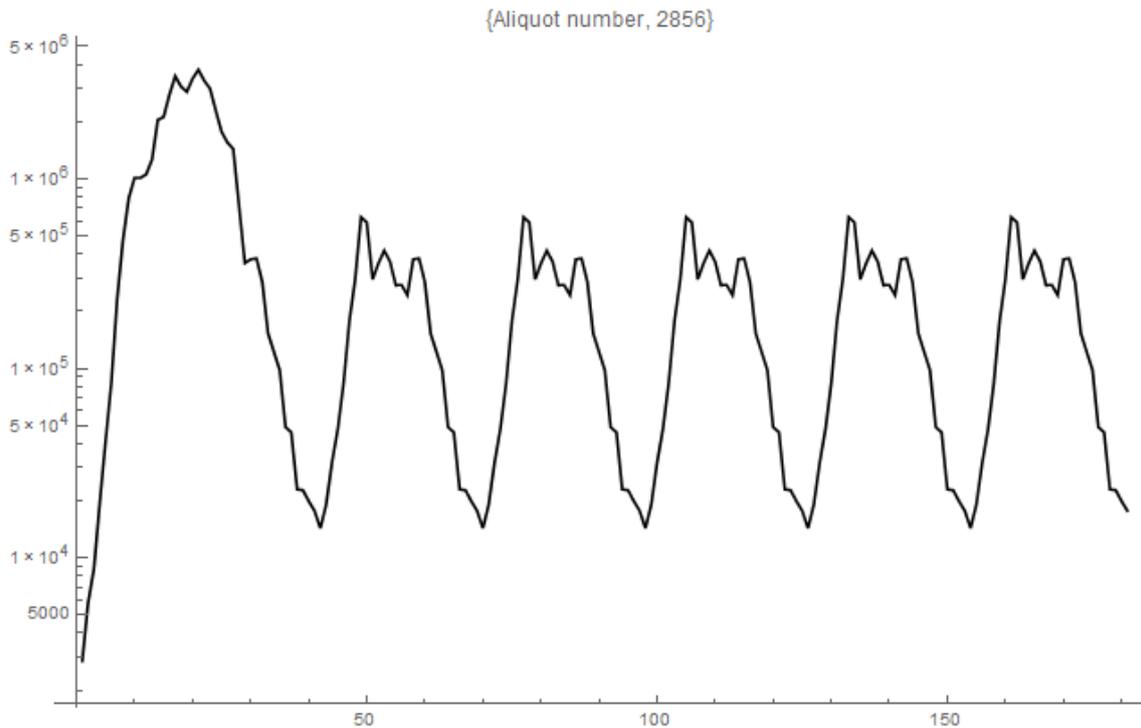


Abbildung 154: Aliquot-Sequenz 2856 (Ende bei Zyklus mit 28 Werten)

Innerhalb einer Aliquot-Sequenz können auch ‚fast perfekte‘ Zahlen auftreten, so z. B. in der terminierten Sequenz mit Startwert 840 bei den Indices 139/140 sowie 140/142:

{ 13938528443323550460883494, 13938528465780941432786**826**, 13938528465780941432786**838**, 23607694429544124013899**882**, 23607694429544124013899**894** }

Hierbei unterscheiden sich die Folgenglieder erst in der 26-ten Stelle mit einer Differenz von 12!

20.9.2.2 FAMILIEN VON ALIQUOT-SEQUENZEN

Alle (zu verschiedenen Startwerten gehörenden Aliquot-Sequenzen) gehören zur gleichen Familie, wenn sie auf die gleiche Art und Weise enden (d.h. mit dem gleichen Zyklus, mit der gleichen Primzahl, oder mit der gleichen OE-Sequenz). Eine Familie von Aliquot-Sequenzen lässt sich sehr schön als Baumstruktur darstellen. Hier ein paar Beispiele (die natürlich nur den unteren Zahlenbereich darstellen):

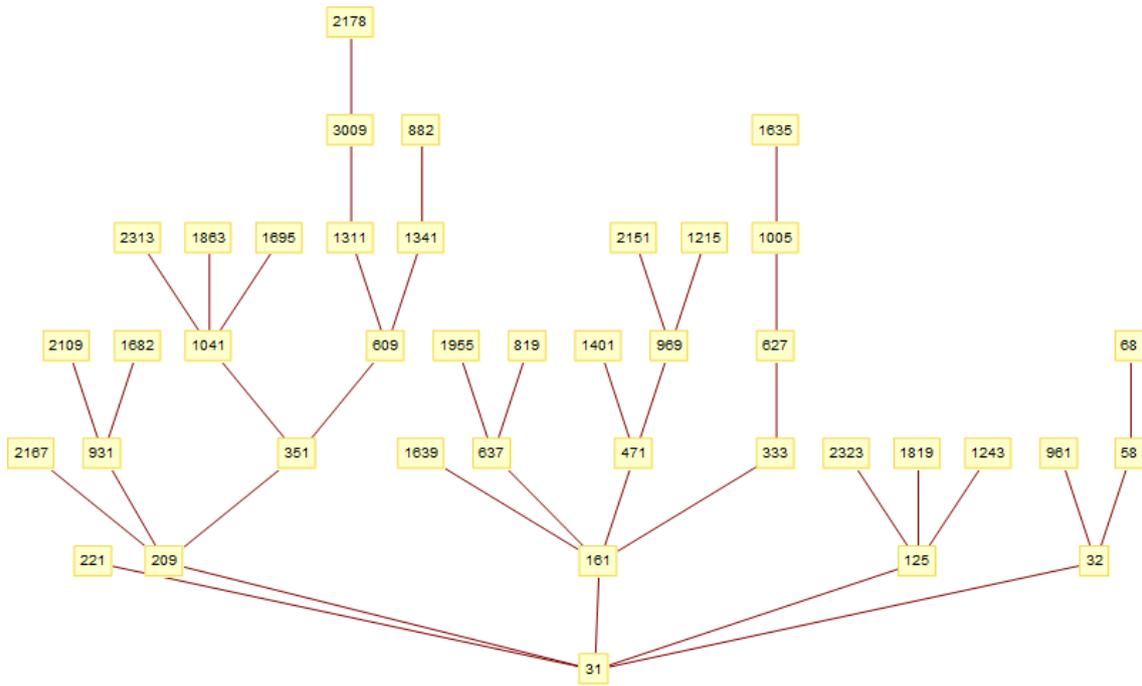


Abbildung 157: Familie von Aliquot-Sequenzen (Sequenzende bei Primzahl 31)

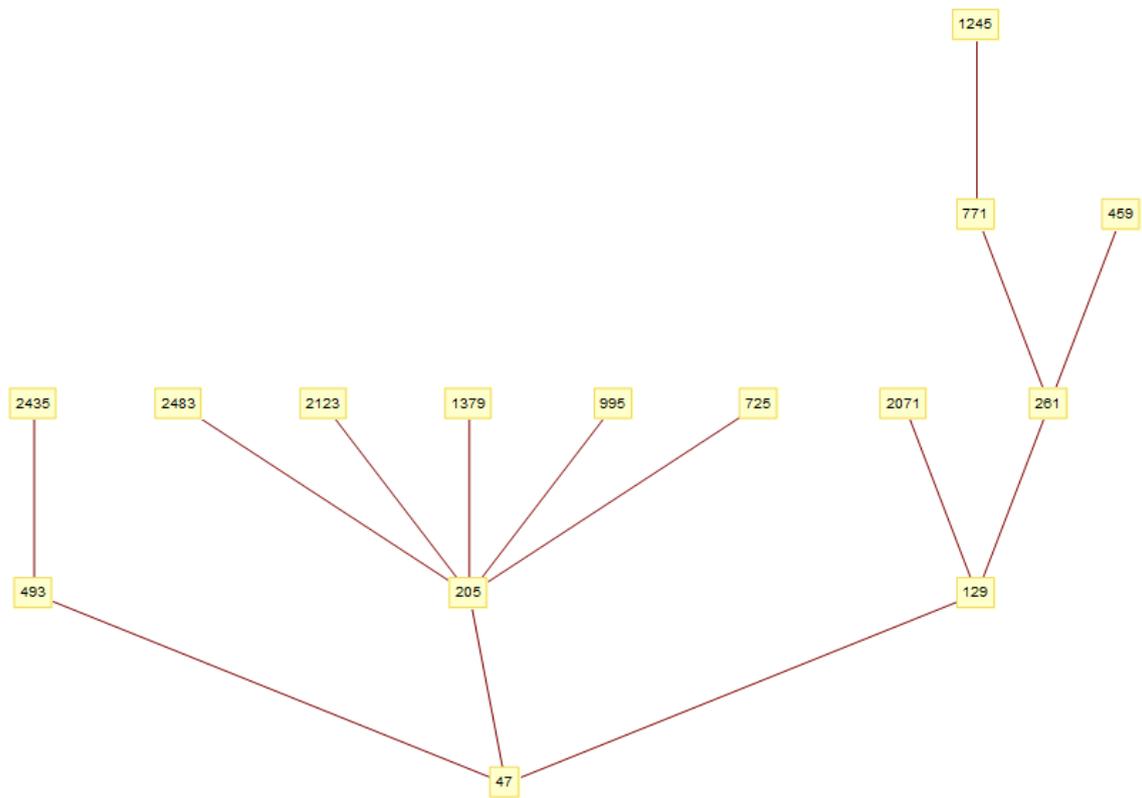


Abbildung 158: Familie von Aliquot-Sequenzen (Sequenzende bei Primzahl 47)

20.9.2.3 LÄNGEN VON ALIQUOT-SEQUENZEN

Bei der Berechnung der Längen von Aliquot-Sequenzen gilt folgende Konvention: Die Folge startet immer mit dem Startwert selbst. Es werden alle Folgenglieder bis zum (einschließlich) ersten, sich wiederholenden Wert gezählt. Da Primzahlen eine Aliquotsumme von 1 haben und die 1 von einer 0 gefolgt wird, besitzen alle Primzahlen eine Sequenzlänge von 3. Vollkommene Zahlen besitzen eine Sequenzlänge von 1. Für OE-Sequenzen wurde (zugegebenermaßen etwas willkürlich) eine Sequenzlänge von 10000 festgelegt.

Hier eine Liste der ersten 300 Sequenzlängen:

```
{2, 3, 3, 4, 3, 1, 3, 4, 5, 5, 3, 8, 3, 6, 6, 7, 3, 5, 3, 8, 4, 7, 3, 6, 2, 8, 4, 1, 3, 16, 3, 4, 7, 9,
4, 5, 3, 8, 4, 5, 3, 15, 3, 6, 8, 9, 3, 7, 5, 4, 5, 10, 3, 14, 4, 6, 4, 5, 3, 12, 3, 10, 4, 5, 4, 13,
3, 6, 5, 7, 3, 10, 3, 6, 6, 6, 4, 12, 3, 8, 6, 7, 3, 7, 4, 10, 8, 8, 3, 11, 5, 7, 5, 5, 3, 10, 3, 4, 5
, 6, 3, 19, 3, 8, 9, 7, 3, 11, 3, 8, 4, 10, 3, 18, 4, 6, 5, 11, 3, 13, 9, 6, 9, 7, 4, 17, 3, 4, 4, 7,
3, 12, 5, 8, 10, 9, 3, 179, 3, 6, 6, 7, 3, 10, 5, 7, 7, 12, 3, 178, 3, 13, 7, 9, 4, 9, 3, 8, 5, 12,
4, 5, 3, 8, 10, 11, 3, 176, 7, 10, 4, 10, 3, 17, 4, 6, 5, 8, 3, 53, 3, 10, 5, 7, 4, 16, 4, 13, 4, 1
1, 3, 14, 3, 7, 7, 5, 3, 15, 3, 5, 4, 9, 4, 11, 4, 8, 10, 8, 4, 53, 3, 12, 7, 9, 6, 11, 5, 11, 5, 2,
4, 177, 3, 18, 9, 7, 3, 9, 3, 10, 8, 12, 3, 176, 4, 8, 4, 8, 3, 12, 3, 4, 10, 12, 4, 16, 8, 13, 9,
12, 3, 18, 5, 8, 6, 7, 3, 15, 9, 12, 5, 9, 3, 32, 4, 10, 6, 9, 3, 14, 3, 13, 5, 7, 4, ???, 3, 8, 4,
17, 3, 18, 3, 2, 8, 12, 6, 11, 6, 13, 4, 8, 3, 17, 5, 8, 6, 14, 4}
```

Im Bereich ab 80 und höher scheint es ‚Bänder‘ zu geben, für die keine Sequenzlängen existieren.

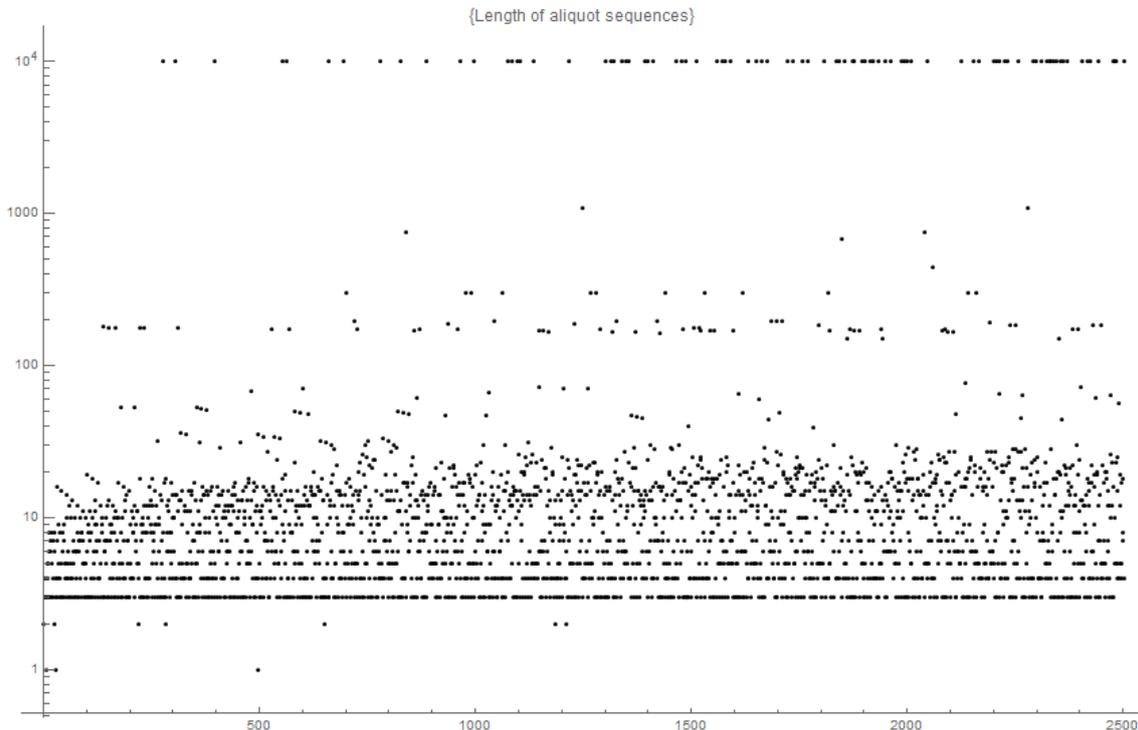


Abbildung 159: Aliquot Sequenzlängen bis n=2500, OE-Sequenzen sind mit einer Länge von 10000 dargestellt.

Das Mathematica-Programm, mit dem die Längen berechnet wurden, findet sich im Anhang.

20.9.2.4 ENDWERTE VON ALIQUOT-SEQUENZEN

Interessant ist auch die Frage, bei welchen Zahlenwerten Aliquot-Sequenzen enden. Da die meisten Sequenzen mit einer 0 enden (mit den Vorgängern einer Primzahl und einer 1), wäre ein solcher Graph höchst langweilig, wenn wir tatsächlich das letzte Glied (entsprechend unserer im letzten Kapitel eingeführten Längenkonvention) hernehmen würden. Wir untersuchen die ‚interessanten‘ Werte und nehmen für die Endwerte folgende Konvention: Bei Sequenzen, die mit 0 enden, nehmen wir die zwei Plätze vorher erscheinende Primzahl, bei zyklischen Enden nehmen wir das erste Element der am Schluss erscheinenden Periode. Bei OE-Sequenzen wählen wir (mangels genauerer Information) den Wert 1. Hier eine Liste mit den ersten 300 Endwerten:

```
{1, 2, 3, 3, 5, 6, 7, 7, 3, 7, 11, 3, 13, 7, 3, 3, 17, 11, 19, 7, 11, 7, 23, 17, 6, 3, 13, 28, 29,
3, 31, 31, 3, 7, 13, 17, 37, 7, 17, 43, 41, 3, 43, 43, 3, 3, 47, 41, 7, 43, 11, 3, 53, 3, 17, 41
, 23, 31, 59, 43, 61, 7, 41, 41, 19, 3, 67, 31, 13, 43, 71, 3, 73, 43, 7, 41, 19, 3, 79, 41, 43
, 43, 83, 37, 23, 3, 3, 41, 89, 3, 11, 41, 13, 43, 6, 37, 97, 73, 23, 19, 101, 3, 103, 41, 3, 4
1, 107, 43, 109, 41, 41, 43, 113, 3, 29, 43, 19, 7, 6, 12161, 3, 41, 3, 19, 31, 3, 127, 127,
47, 41, 131, 43, 13, 43, 3, 43, 137, 59, 139, 37, 11, 43, 6, 3, 13, 41, 43, 7, 149, 59, 151,
7, 43, 43, 37, 37, 157, 43, 23, 43, 31, 71, 163, 41, 3, 3, 167, 59, 7, 43, 89, 43, 173, 3, 73
, 37, 41, 41, 179, 601, 181, 43, 19, 37, 43, 3, 29, 7, 131, 43, 191, 43, 193, 19, 11, 37, 19
7, 3, 199, 59, 71, 41, 37, 43, 47, 41, 3, 43, 31, 601, 211, 3, 7, 41, 7, 73, 17, 43, 19, 220,
31, 59, 223, 41, 41, 43, 227, 41, 229, 41, 43, 43, 233, 59, 53, 37, 83, 19, 239, 12161, 24
1, 157, 3, 43, 97, 3, 3, 43, 3, 43, 251, 59, 13, 41, 41, 41, 257, 3, 3, 43, 47, 43, 263, 59, 5
9, 41, 13, 43, 269, 3, 271, 43, 73, 37, 97, 1, 277, 43, 137, 41, 281, 163, 283, 284, 11, 43
, 7, 3, 11, 43, 101, 43, 293, 163, 19, 37, 19, 7, 37}
```

Als Erstes fällt auf, dass die meisten Endwerte aus Primzahlen bestehen. Die wenigen zusammengesetzten Zahlen gehören zu Sequenzen, die zyklisch enden. Bemerkenswert ist die Beobachtung, dass die Primzahl 5 als Endwert nur ein einziges Mal vorkommt (nämlich auf Platz 5). Im Bereich zwischen 1 und 300 gibt es nur eine einzige OE-Sequenz (markiert durch die rote ‚1‘).

In der graphischen Darstellung fallen zwei Linien auf, die durch Punkthäufungen entstehen. Die gekrümmte Linie: Diese markiert die Primzahlen. Die geraden Linien, parallel zur X-Achse bei den Werten 41, 43 und 59: Hier ist deutlich eine unerklärliche Häufung zu sehen. Die OE-Sequenzen erscheinen diesmal alle auf einer Geraden beim Wert 1. Im Durchschnitt enden fast 8 % aller Aliquot-Sequenzen beim Wert 43, ca. 5 % beim Wert 59 und 5 % bei 41. Andere Werte wie 5 oder die 28 kommen dagegen nur ein einziges Mal vor!

Weitere ungelöste mathematische Probleme

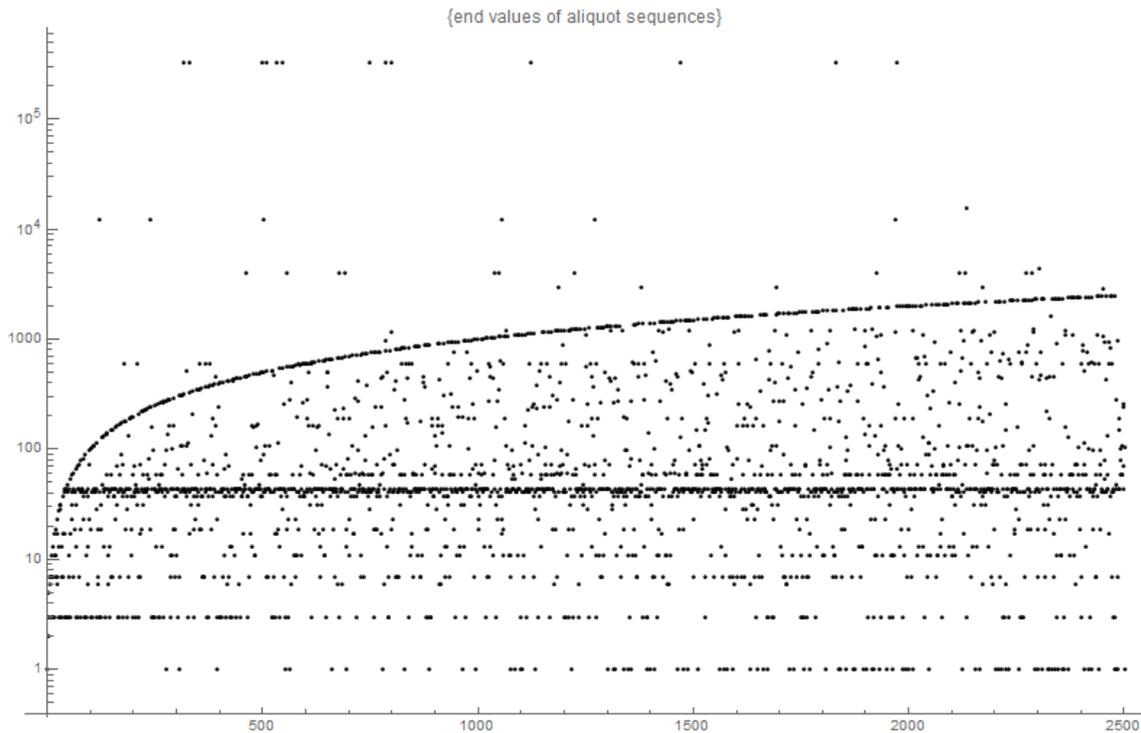


Abbildung 160: Endwerte von Aliquot-Sequenzen für Startwerte bis 2500

Man mag sich fragen, welche Aliquot-Startwerte denn zyklische Endwerte (einschließlich der perfekten Zahlen mit Zykluslänge 1) ergeben. Plottet man diese Startwerte einfach in aufsteigender Reihenfolge, dann sieht man, dass ihre ‚Dichte‘ im Durchschnitt konstant bleibt, denn der Anstieg ist mit guter Näherung linear (die Gleichung der Fit-Gerade lautet: $f(x) = 14.512 + 40.8404 x$).

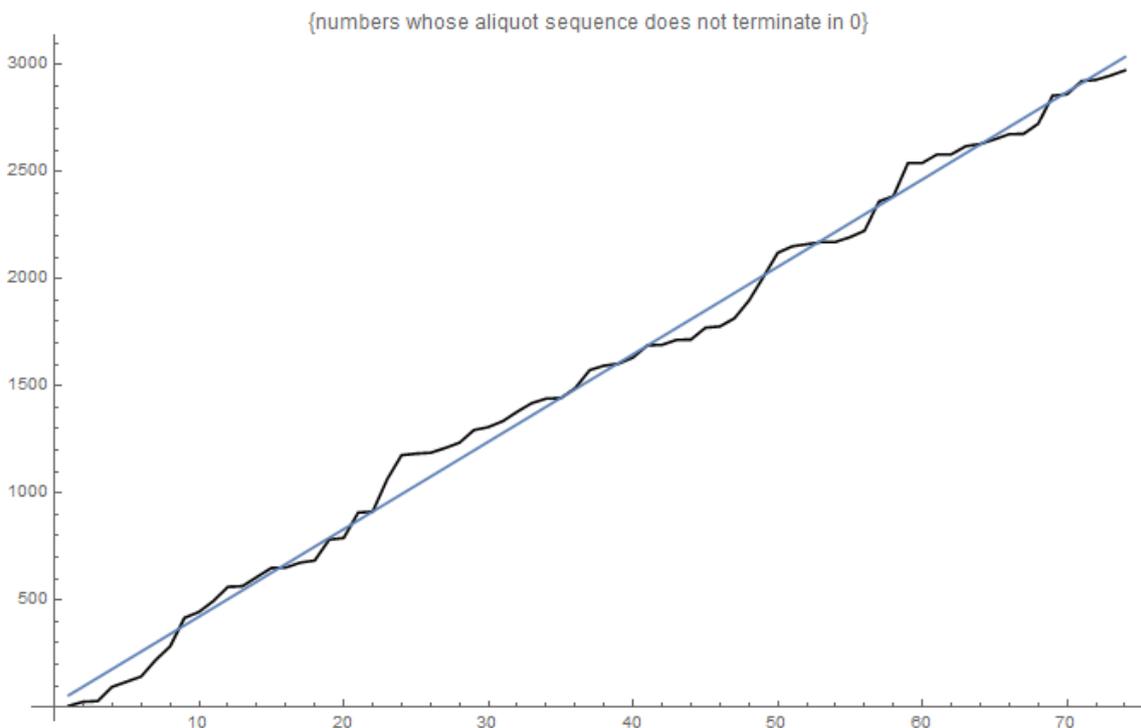


Abbildung 161: Startwerte von Aliquot-Sequenzen, die ein zyklisches Ende ergeben

20.9.2.5 DIFFERENZEN UND QUOTIENTEN VON ALIQUOT-SEQUENZEN

Die nachfolgenden Betrachtungen wurden am Beispiel der Aliquot-Sequenz mit dem Startwert 840 illustriert. Sie gelten jedoch allgemein genauso für die meisten anderen Aliquot-Sequenzen.

Betrachten wir die Differenzen von jeweils zwei aufeinanderfolgenden Folgengliedern, so fällt zunächst auf, dass diese vom Betrag her in den meisten Fällen fast in der gleichen Größenordnung wie die Folgenglieder selbst liegen. Eine Ausnahme bilden die ‚fast-perfekten‘ Zahlen (siehe Kapitel 20.9.2.1). Mehr noch: Ein Plot der Differenzen zeigt eine gewisse ‚Forminvarianz‘ gegenüber der originalen Aliquot-Sequenz. Diese Forminvarianz bleibt auch bei Differenzen höherer Ordnung bestehen (vom Autor getestet bis Differenzenordnungen von über 20). Bei den Plots wurden die logarithmischen Werte der Differenzen genommen. Die Forminvarianz wird noch besser sichtbar, wenn die negativen Differenzen ‚nach oben geklappt‘ (somit die Absolutbeträge) genommen werden:

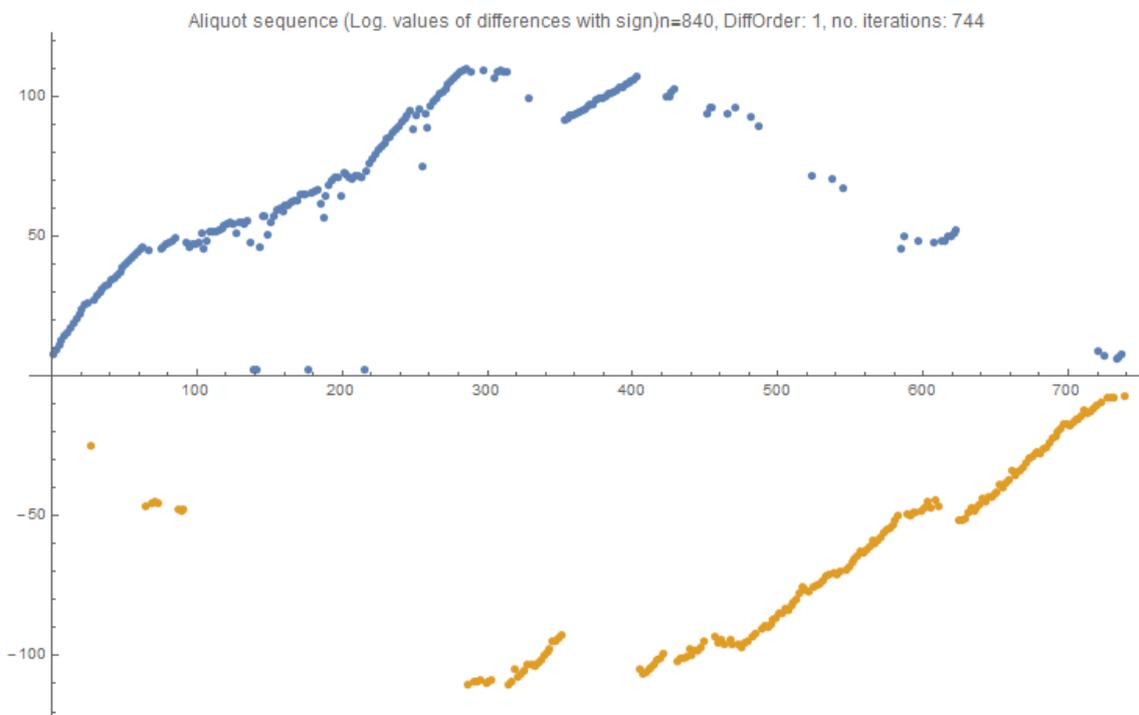


Abbildung 162: Aliquot-Sequenz: logarith. Differenzen für Startwert $n = 840$, mit Vorzeichen

Mathematica-Programm: Bitte den Autor kontaktieren.

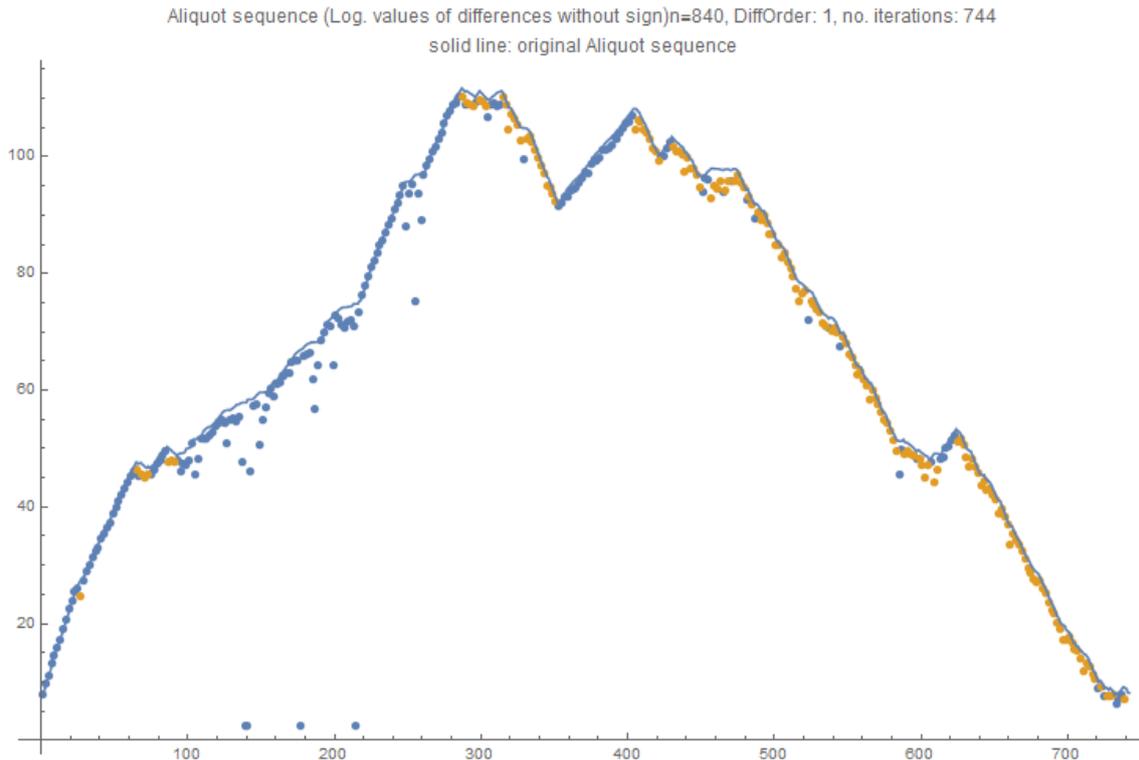


Abbildung 163; Aliquot: logarith. Differenzen für Startwert $n=840$, ohne Vorzeichen

Noch interessanter wird es, wenn wir nicht die logarithmischen Werte der Differenzen, sondern die Differenzen der logarithmischen Werte betrachten, was dem Quotienten zweier aufeinanderfolgenden Werten entspricht. Es existieren Häufungspunkte, die ungefähr bei den Werten $\ln\left(\frac{1}{2}\right)$, $\ln\left(\frac{3}{4}\right)$, 0 , $\ln\left(\frac{5}{4}\right)$ und 1 liegen, was Quotientenwerten von $\frac{1}{2}$, $\frac{3}{4}$, 1 , $\frac{5}{4}$ und e entspricht. Weiterhin lässt sich feststellen, dass die ‚Anstiege‘ innerhalb der Sequenzen im Durchschnitt steiler als die ‚Abstiege‘ ausfallen. Bei den Steigungen im Anstieg gibt es keine ausgeprägten Häufungspunkte, dafür jedoch eine obere Schranke von 1 (bzw. e) (von gelegentlichen kleinen Ausrutschern bei OE-Sequenzen, die knapp über 1 liegen, abgesehen). Beim Abstieg haben wir eine untere Schranke von -0.693 (bzw. 0.5). Das bedeutet, dass die Glieder von Aliquot-Sequenzen nicht schneller als mit einem Faktor e wachsen können, bzw. mit einem Faktor 0.5 kleiner werden können. Umgangssprachlich: Der Abstieg geht immer langsamer als der Aufstieg, und dennoch steigen fast alle Aliquot-Sequenzen wieder ab und kommen bei kleinen Werten ‚zur Ruhe‘. Hinweis: All dies sind nur rein empirische Betrachtungen, ohne Anspruch auf strenge, mathematische Gültigkeit.

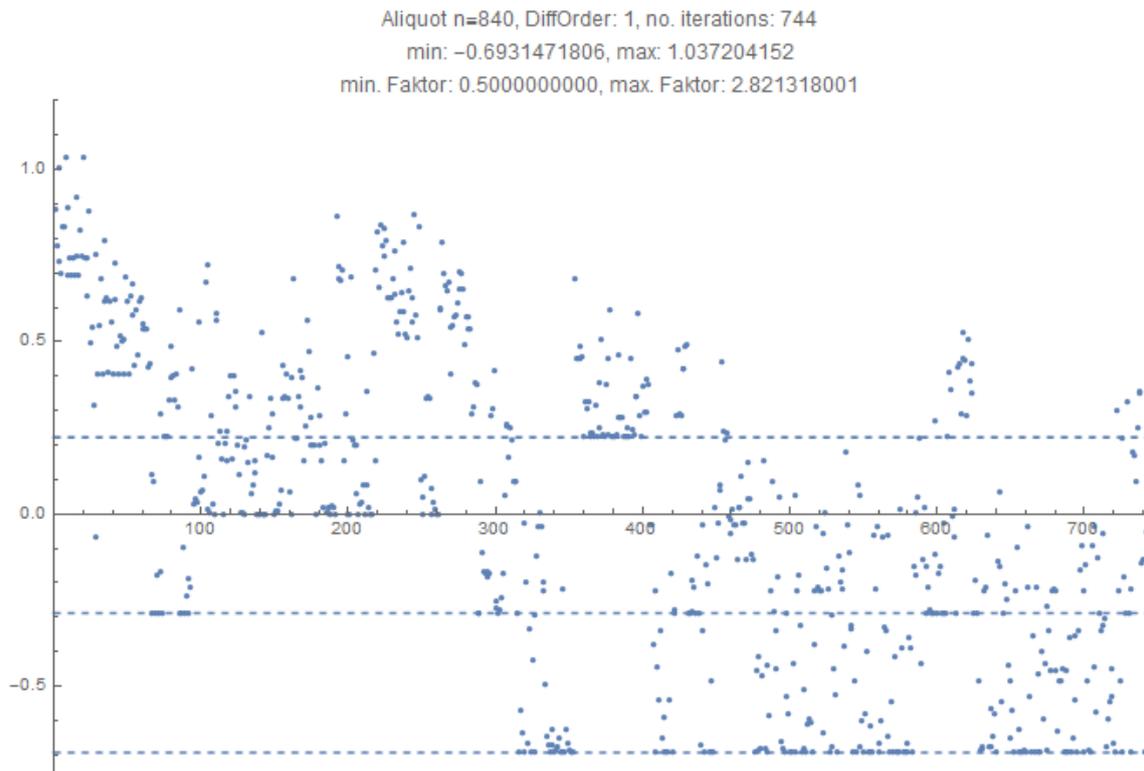


Abbildung 164: Aliquot: Differenzen der logarith. Werte für Startwert n= 840

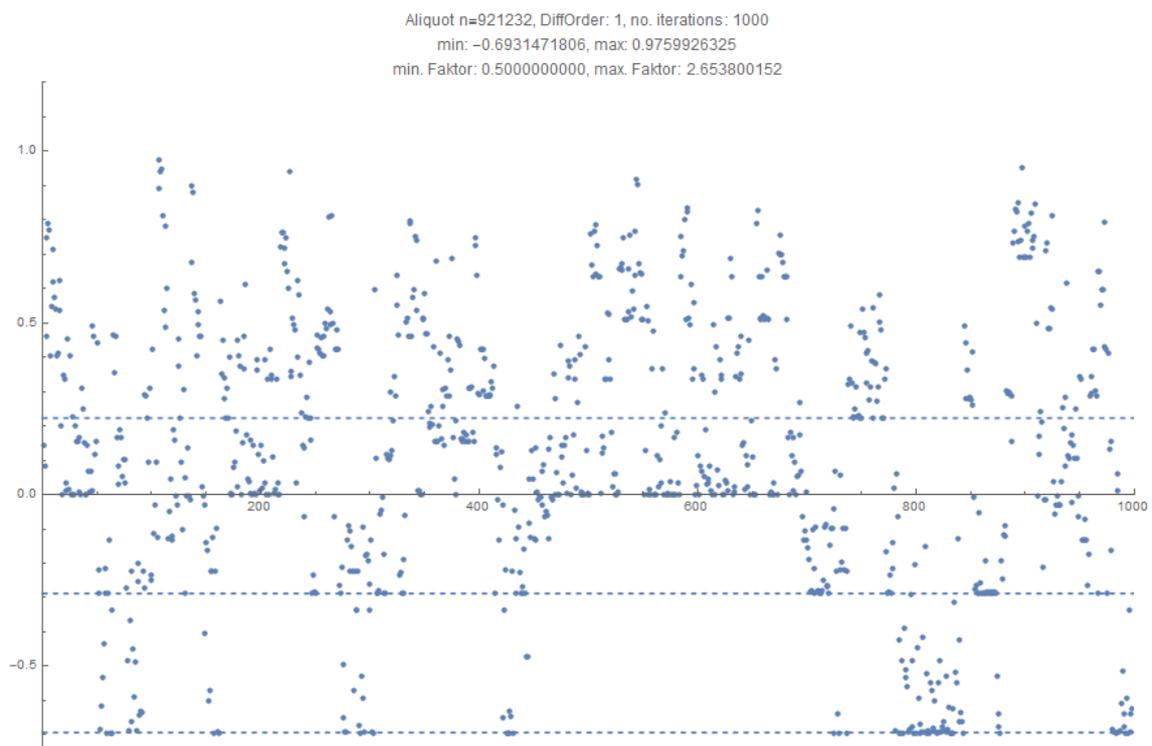


Abbildung 165: Aliquot: Differenzen der logarith. Werte für Startwert n= 921232

Mathematica:
 (Programm in Anhang)

Weitere ungelöste mathematische Probleme

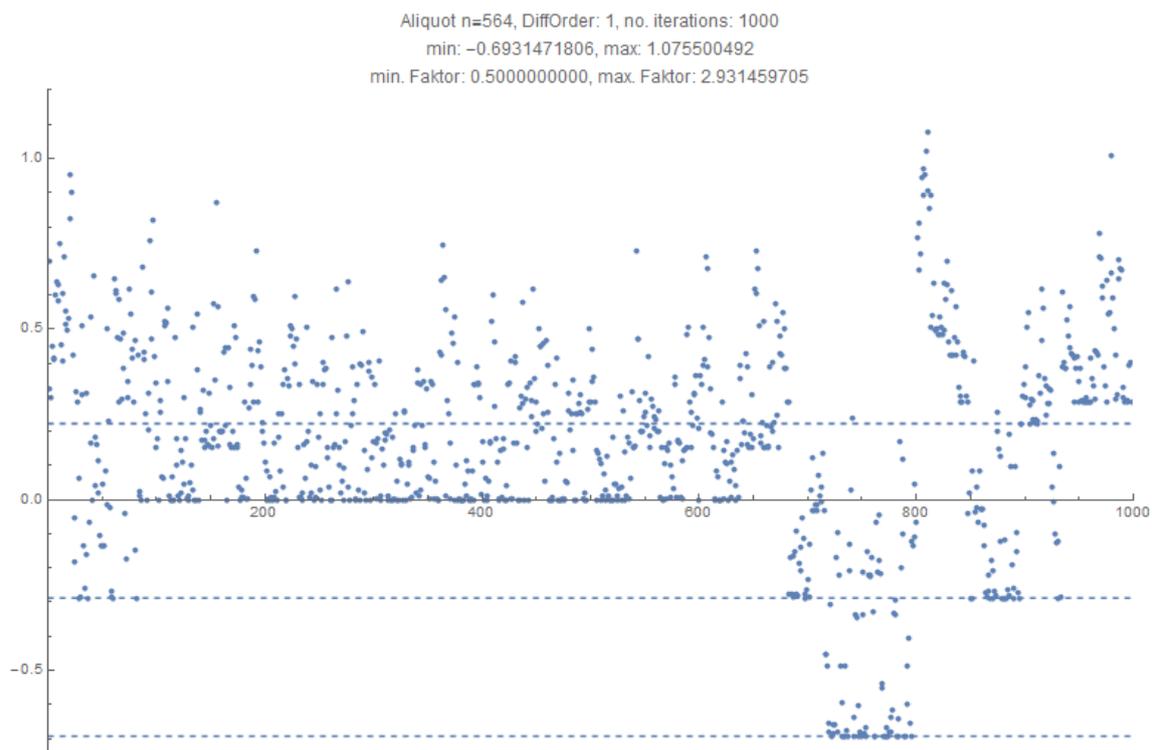


Abbildung 166: Aliquot: Differenzen der logarith. Werte für Startwert $n=564$ (OE)

20.9.3 FAKTORISIERUNG VON ZAHLEN

Mit Mathematica lassen sich relativ große Zahlen faktorisieren:
Zum Beispiel folgende 68-stellige Zahl:

```
CenterDot@@(Superscript@@@  
FactorInteger[24284712165828060817808704394685584572191120513988  
451223045457718339])
```

Liefert nach 1.5 Sekunden Rechenzeit (auf einem 2.6 GHz Quad-Core Intel-PC)
folgende Primfaktorenzerlegung:

$$29996224275833^2 \cdot 29996224275851^3$$

Es gibt mit Mathematica weiterhin beliebig viele Methoden, um Zahlen zu faktorisieren
(ob diese sehr effektiv sind, sei dahingestellt), z. B.:

```
n=1037;
```

```
Solve[x*y== n&& x>1&& x<n&& y>1&& y<n, {x,y}, Integers]
```

oder:

```
FindInstance[x*y== n&& x>1&& x<n&& y>1&& y<n, {x,y}, Integers]
```

ergibt:

```
{{x->17, y->61}}
```

20.9.3.1 DIE WICHTIGSTEN FAKTORISIERUNGSMETHODEN

Die gegenwärtig (Stand Feb. 2016) wichtigsten Faktorisierungsmethoden sind hier aufgelistet, ohne auf deren Algorithmen oder Implementierung einzugehen. In der Praxis werden meist mehrere Methoden verwendet, je nach dem in welchem Zahlenbereich man sich befindet. Es werden meist auch mehrere Verfahren miteinander kombiniert. So findet am Beginn eines Faktorisierungsverfahrens meist ein Test mit vergleichsweise kleinen Faktoren (Probe-Division) statt, um kleine Faktoren schnell zu finden, bevor dann die ‚schweren Geschütze‘ wie die ECM-Methode oder ‚Zahlenkörpersieb‘ angeworfen werden.

Faktorisierungsmethoden:

- Probedivision (zum Herausfinden kleiner Faktoren)
- Fermatsche Faktorisierungsmethode
- Pollard ‚p-1‘ Methode
- Pollard ‚rho‘-Methode (sucht nach $x \equiv y \pmod{p}$ in einer Pseudo-Zufallsszählensequenz)
- Williams’s ‚p+1‘-Methode
- Methode von Pomerance
- Kettenbruch-Methoden
- ECM-Methode der elliptische Kurven
- -Shanks‘ SQUFOF-Methode
- Quadratische Siebmethoden
- Zahlenkörpersieb

Diese Verfahren sind ausführlich in zahlreichen Internet-Seiten dokumentiert, deshalb soll hier nicht darauf eingegangen werden.

Wegen ihrer Einfachheit und Schönheit sei hier Fermats Methode kurz dargestellt:

Sei n unsere zu faktorisierende Zahl. Der Algorithmus funktioniert nur für ungerade Zahlen n . Wir testen den Ausdruck $(\lceil\sqrt{n}\rceil + i)^2 - n$ (durch Inkrementieren von i , beginnend von $i = 0$) solange, bis er eine Quadratzahl y^2 ergibt:

$(\lceil\sqrt{n}\rceil + i)^2 - n = y^2$. Mit $x = \lceil\sqrt{n}\rceil + i$ ergibt dies: $x^2 - n = y^2$, bzw.

$n = (x + y)(x - y)$. Damit haben wir zwei Faktoren von n gefunden.

Hier ein Beispiel: $n=1037$. Dann ist $\lceil\sqrt{1037}\rceil = 33$. Wir bekommen dann folgende Sequenz:

$$\begin{aligned} (33 + 0)^2 - 1037 &= 52 \\ (33 + 1)^2 - 1037 &= 119 \\ (33 + 2)^2 - 1037 &= 188 \\ (33 + 3)^2 - 1037 &= 259 \\ (33 + 4)^2 - 1037 &= 332 \\ (33 + 5)^2 - 1037 &= 407 \\ (33 + 6)^2 - 1037 &= 484 (= 22 * 22) \end{aligned}$$

Damit hat x den Wert 39 und y den Wert 22 und somit haben wir die beiden Faktoren $p = 39 + 22 = 61$ und $q = 39 - 22 = 17$. $1037 = 17 * 61$.

```
Mathematica:
n=17*61;sqN=Ceiling[Sqrt[n]];value=2;
For[i=0,i<n&&IntegerQ[Sqrt[value]]==False,i++,
Print[i,"->",value=(sqN+i)^2-n];i--;
y=Sqrt[value]; x=sqN+i;
p=x+y; q=x-y
Print["Faktoren: ",p,"*",q];
```

Der Algorithmus kann beschleunigt werden, indem man das wiederholte Quadrieren vermeidet und $(s + i)^2$ rekursiv aus dem Vorgänger-Term bestimmt: $(s + 1)^2 - n = s^2 + 2s + 1 - n$. Der Test, ob y^2 eine Quadratzahl ist läßt sich auch beschleunigen, indem die letzten beiden Ziffern der Zahl getestet werden (es gibt für beliebig große Quadratzahlen nur 22 von 100 verschiedenen Möglichkeiten für die letzten beiden Ziffern).

Die Laufzeit dieses Algorithmus ist gut ($\sim\sqrt{n}$), wenn beide Faktoren etwa gleich groß sind. Sie wird jedoch schlecht, wenn einer der Faktoren sehr klein wird (z.B. 3).

Die Iteration ist immer endlich, d.h. sie wird immer bei einer Quadratzahl abbrechen. Bei Primzahlen treten jedoch viele Iterationen auf, so dass diese Methode als Primzahltest ungeeignet ist. Die schlechte Laufzeit bei unterschiedlich großen Faktoren kann man umgehen, indem man einen geeigneten Faktor k findet, so dass der Algorithmus an $k \cdot n$ angewendet, zwei näher beieinander liegende Faktoren findet. Ein solcher Algorithmus

ist wesentlich effizienter als der Fermat-Algorithmus und existiert als die ‚Lehman-Methode‘⁸⁷.

20.9.3.2 ANDERE FAKTORISIERUNGSMETHODEN

Der Autor möchte hier ein paar unkonventionelle Methoden vorstellen, unabhängig von deren praktischer Anwendbarkeit.

Die Sigma-Phi-Methode:

Sei n das Produkt von genau zwei unterschiedlichen Primzahlen: $n = pq$.

Dann gilt:

$$\sigma(n) = (p + 1)(q + 1) = n + 1 + (p + q)$$

$$\varphi(n) = (p - 1)(q - 1) = n + 1 - (p + q)$$

p und q lassen sich berechnen:

$$p = \frac{(\sigma(n) - \varphi(n))}{4} - \sqrt{\left[\frac{(\sigma(n) - \varphi(n))}{4}\right]^2 - \left[\frac{(\sigma(n) + \varphi(n))}{2}\right] + 1} \quad (166)$$

$$q = \frac{(\sigma(n) - \varphi(n))}{4} + \sqrt{\left[\frac{(\sigma(n) - \varphi(n))}{4}\right]^2 - \left[\frac{(\sigma(n) + \varphi(n))}{2}\right] + 1} \quad (167)$$

Beispiel: $n = 1037$

$\sigma(n)$: 1116, $\varphi(n)$: 960, $\frac{(\sigma(n) - \varphi(n))}{4}$: 39 ergibt $1037 = 17 * 61$

Beispiel: $n = 519920418755535776857$

$\sigma(n)$: 519920418801139303860, $\varphi(n)$: 519920418709932249856,

$\frac{(\sigma(n) - \varphi(n))}{4}$: 22801763501 ergibt

$519920418755535776857 = 22801763489 * 22801763513$

Mathematica:

```
n=519920418755535776857;
sigmaN=DivisorSigma[1,n]; eulerP=EulerPhi[n];
sum=sigmaN+eulerP; dif=sigmaN-eulerP; sqTerm=(dif/4)^2-sum/2+1;
p=dif/4-Sqrt[sqTerm]
q=dif/4+Sqrt[sqTerm]
```

Damit hat man allerdings das Problem der Faktorisierung von n nur verlagert auf die Bestimmung von $\sigma(n)$ und $\varphi(n)$, was wiederum eine ähnliche Komplexität bedeutet.

⁸⁷ Richard Crandall, Carl Pomerance: *Prime Numbers. A Computational Perspective*, S. 191

Eine ‚ganz verrückte‘ Methode (analytisch):

Wir betrachten die Funktion zweier Variablen

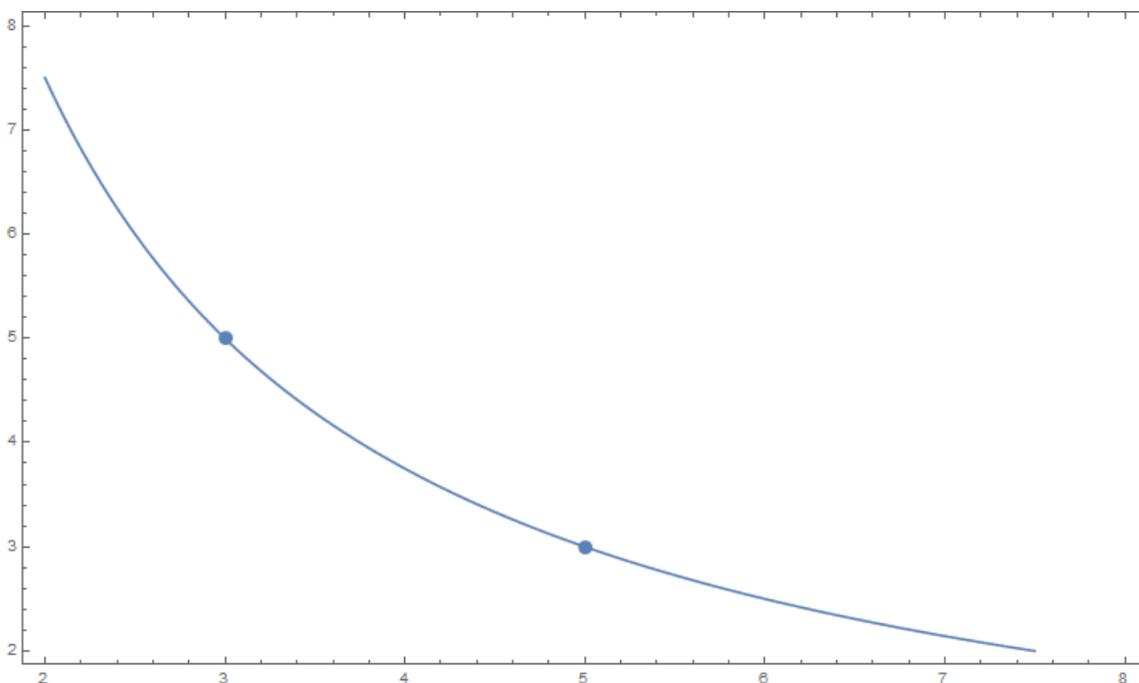
$$\text{productF}(x, y) = x * y - n$$

und untersuchen, für welche Werte x und y diese Funktion den Wert 0 annimmt. Diese Werte liegen alle auf einer Null-Linie und stellen sozusagen alle ‚reellen‘ Faktoren von n dar (in diesem Fall ist dies eine Hyperbel). Wenn wir uns aus dieser Null-Linie die ganzzahligen (x, y) –Werte herauspicken, dann haben wir n faktorisiert.

Beispiel:

$$\text{productF}(x, y) = x * y - 15$$

Die ‚Null-Linie‘ als Contour-Plot sieht wie folgt aus:



Die ganzzahligen Werte der Null-Linie liegen, wie man sieht bei den Punkten (3,5) und (5,3).

Mathematica:

```
testF[m_]:=If[val=Abs[Round[{m}]-{m}];val[[1]][[1]]<10^(-5)&&val[[1]][[2]]<10^(-5),True,False];
primeIndex=2; Prime[primeIndex]
Prime[primeIndex+1]
n=Prime[primeIndex]*Prime[primeIndex+1]
sqN=Round[Sqrt[n]+1];
productF[x_,y_]:=((x)*(y)-n);
(*Find Zero-Line:*)
ptsxy=ContourPlot[productF[x,y]==0,{x,2,8},{y,2,8},MaxRecursion->4];
Show[ptsxy,ListPlot[{{3,5},{5,3}},ImageSize->{708,425},AspectRatio->Full]
ptsxyl=Cases[Normal@ContourPlot[productF[x,y]==0,{x,2,8},{y,2,8},
MaxRecursion->4],Line[{x__}]>x,Infinity]
Round[Select[ptsxyl,testF]]
Sort[DeleteDuplicates[Round[Select[ptsxyl,testF]]]]
```

Ergibt:

$$\{\{3, 5\}, \{5, 3\}\}$$

Eine analytische Methode:

Wir starten wieder mit $x * y = n$, mit $x, y \in \mathbb{R}$ und $n \in \mathbb{N}$ und suchen nach ganzzahligen Lösungen von x, y .

Dies stellt eine Gleichung mit zwei Unbekannten dar, mit der Randbedingung, dass x, y ganzzahlig sein sollen. Zum Lösen von Gleichungen mit zwei Unbekannten brauchen wir zwei Gleichungen. Die Frage ist nun: Woher bekommen wir eine zweite Gleichung, die die Randbedingung der Ganzzahligkeit ausdrückt? Hierfür kommen mehr Möglichkeiten in Frage, z. B.:

$$\sin\left(\pi\left(2x - \frac{1}{2}\right)\right) + \sin\left(\pi\left(2y - \frac{1}{2}\right)\right) + 2 = 0 \quad (168)$$

$$x \cdot y = n \quad (169)$$

Gleichungen (168) und (169) beschreiben ein nichtlineares Gleichungssystem von zwei Gleichungen mit zwei Unbekannten. Die reellen Lösung(en) dieses Gleichungssystem ergeben die Primfaktoren unserer Zahl n . Die Lösung ist jedoch schwierig und mit einfachen Mitteln nicht möglich. Man kann (169) nach y auflösen und in (168) einfügen. Wenden wir dann noch eine Potenzfunktion $(x)^{\frac{1}{3}}$ an (um die ‚Fast-Lösungen‘ ein bisschen weiter von der X-Achse wegzurücken), dann erhalten wir folgende Funktion:

$$\text{fakFunc}(x, n) = \left(\sin\left(\pi\left(2x - \frac{1}{2}\right)\right) + \sin\left(\pi\left(2\frac{n}{x} - \frac{1}{2}\right)\right) + 2 \right)^{\frac{1}{3}} \quad (170)$$

Die reellen Nullstellen von $\text{fakFunc}(x)$ ergeben die komplette Liste aller möglichen Teiler von n .

Beispiel mit $n = 1037$:

Weitere ungelöste mathematische Probleme

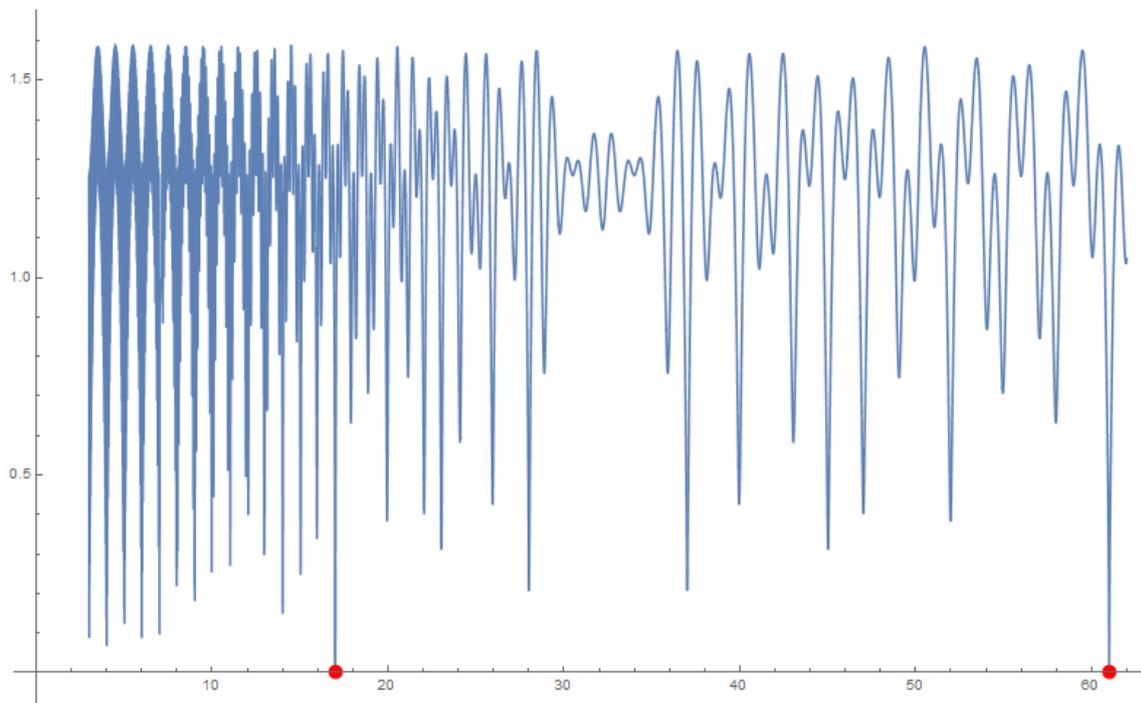


Abbildung 167: $\text{FakFunc}(x, 1037)$ mit Nullstellen bei Primfaktoren 17 und 61

Mathematica:

```
n=1037;  
intFunc[x_, y_] := (Sin[Pi*(2x-1/2)] + Sin[Pi*(2y-1/2)]) + 2;  
Show[Plot[(intFunc[x, n/x])^(1/3), {x, 3, 62}, MaxRecursion->15, AxesOrigin->  
>{0, 0}], ListPlot[{{17, 0}, {61, 0}}, PlotStyle->Red]]
```

Die Funktion $\text{intFunc}(x, y)$, Formel (168) sieht übrigens aus wie ein ‚Eierkarton‘:

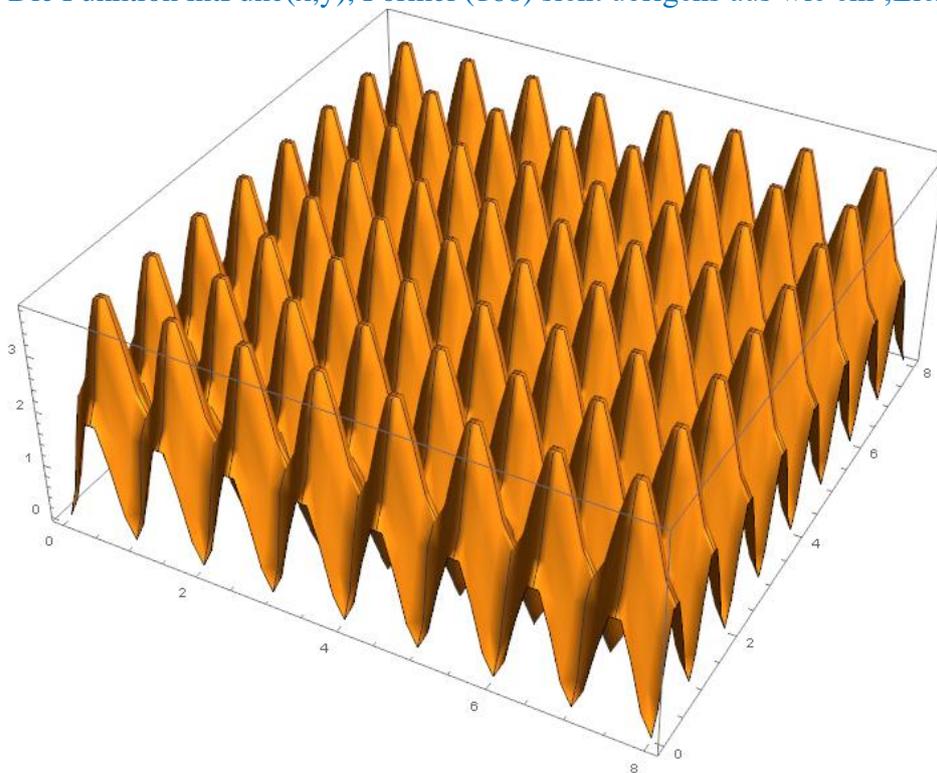


Abbildung 168: Funktion $f(x, y)$: Hat für jeden ganzzahligen (x, y) -Punkt eine Nullstelle

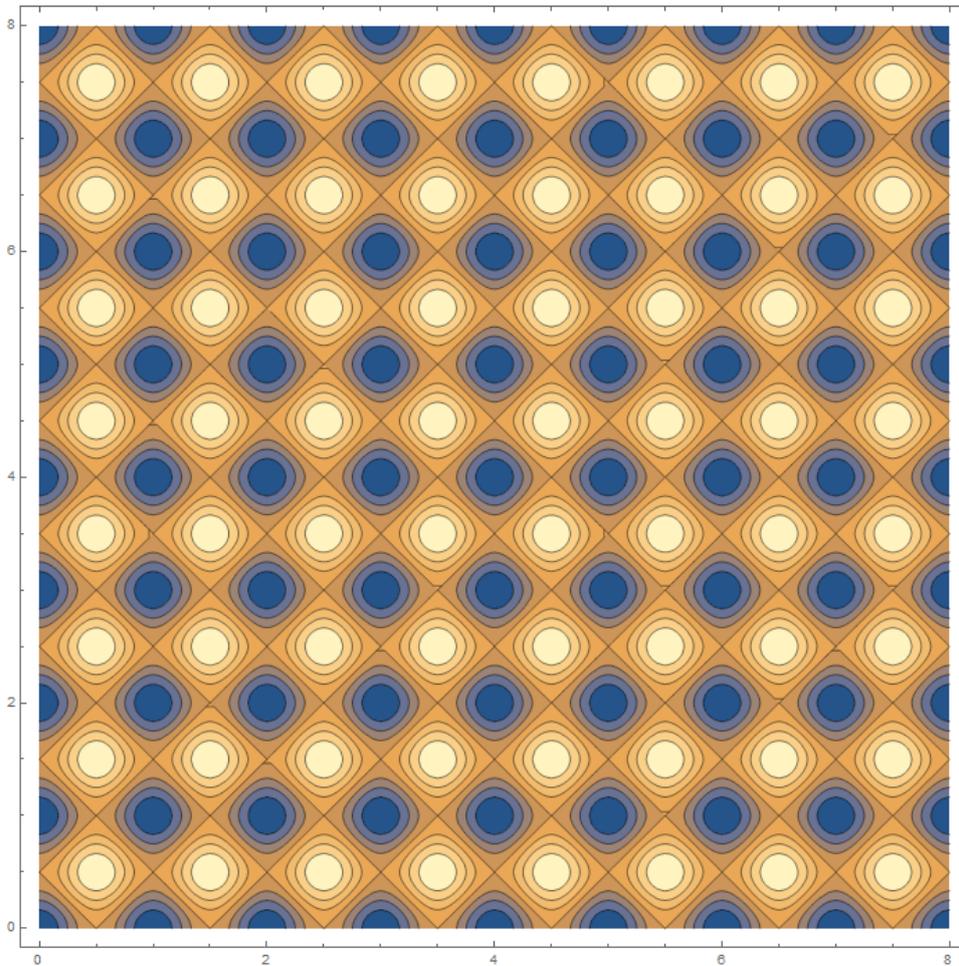


Abbildung 169: wie oben, jedoch Contour-Plot

Mathematica:

```
ContourPlot [intFunc[x,y], {x,0,8}, {y,0,8}, ImageSize->Large]
Plot3D[intFunc[x,y], {x,0,8}, {y,0,8}, ImageSize->Large]
```

20.10 TABELLEN

20.10.1 ANZAHL DER PRIMZAHLEN BIS ZU EINER GRENZE N: $\pi(N)$

Exakte Werte von $\pi(x)$ für x bis 10^{27} stehen in der „Online Encyclopedia of Integer Sequences“ (<http://oeis.org>) . z. B: [A006880](#):

Tabelle 29: Vergleich der exakten Pi-Funktion mit der Riemann-Funktion (gerundet)

n	$\pi(10^n)$	Riemann(10^n)	Riemann(10^n)- $\pi(10^n)$
0		0	0
1		4	5
2		25	26

Tabellen

3	168	168	0
4	1229	1227	2
5	9592	9587	-5
6	78498	78527	29
7	664579	664667	88
8	5761455	5761552	97
9	50847534	50847455	-79
10	455052511	455050683	-1828
11	4118054813	4118052495	-2318
12	37607912018	37607910542	-1476
13	346065536839	346065531066	-5773
14	3204941750802	3204941731602	-19200
15	29844570422669	29844570495887	73218
16	279238341033925	279238341360977	327052
17	2623557157654233	2623557157055978	-598255
18	24739954287740860	24739954284239494	-3501366
19	234057667276344607	234057667300228940	23884333
20	2220819602560918840	2220819602556027015	-4891825
21	21127269486018731928	21127269485932299724	-86432204
22	201467286689315906290	201467286689188773625	-127132665
23	1925320391606803968923	1925320391607837268776	1033299853
24	18435599767349200867866	18435599767347541878147	-1658989719
25	176846309399143769411680	176846309399141934626966	-1834784714
26	1699246750872437141327603	1699246750872419991992147	-17149335456
27	16352460426841680446427399	16352460426841662910939465	-17535487934
28	157589269275973410412739598	-	?
29	1520698109714272166094258063		

Tabelle 30: Vergleich der exakten Pi-Funktion mit der Riemanns exakter Formel $\pi^*(n)$ (siehe (132), Summe über 10000 Nullstellen, gerundet

n	$\pi(10^n)$	$\pi^*(10^n)$	$\pi^*(10^n) - \pi(10^n)$
0	0	-	-
1	4	4	0
2	25	25	0
3	168	168	0
4	1229	1229	0
5	9592	9592	0
6	78498	78498	0
7	664579	664579	0
8	5761455	5761462	7
9	50847534	50847519	-15
10	455052511	455052528	17
11	4118054813	4118054697	-116

12	37607912018	37607911016	-1002
13	346065536839	346065537034	195
14	3204941750802	3204941747414	-3388
15	29844570422669	29844570424541	1872
16	279238341033925	279238341008610	-25315
17	2623557157654233	2623557157681368	27135
18	24739954287740860	24739954288134940	394080
19	234057667276344607	234057667277476288	1131681
20	2220819602560918840	2220819602559672832	-1246008
21	21127269486018731928	21127269486003990528	-14741400
22	201467286689315906290	201467286689365917696	50011406
23	1925320391606803968923	1925320391606731276288	-72692635
24	18435599767349200867866	18435599767349571354624	370486758
25	176846309399143769411680	176846309399143087341568	-682070112
26	1699246750872437141327603	1699246750872436043939840	-1097387763
27	16352460426841680446427399	16352460426841662628560896	-17817866503
28	157589269275973410412739598	-	?
29	1520698109714272166094258063		

Tabelle 31: Vergleich der exakten Pi-Funktion mit der Riemanns exakter Formel $\pi^*(n)$ (siehe (132), Summe über 100000 Nullstellen, gerundet

n	$\pi(10^n)$	$\pi^*(10^n)$	$\pi^*(10^n) - \pi(10^n)$
0	0	-	-
1	4	4	0
2	25	25	0
3	168	168	0
4	1229	1229	0
5	9592	9592	0
6	78498	78498	0
7	664579	664579	0
8	5761455	5761457	2
9	50847534	50847536	2
10	455052511	455052532	21
11	4118054813	4118054886	73
12	37607912018	37607911595	-423
13	346065536839	346065537866	1027
14	3204941750802	3204941749206	-1596
15	29844570422669	29844570413033	-9636
16	279238341033925	279238341037530	3605
17	2623557157654233	2623557157660142	5909
18	24739954287740860	24739954287711076	-29784
19	234057667276344607	234057667276885600	540993
20	2220819602560918840	2220819602559328000	-1590840
21	21127269486018731928	21127269486015279104	-3452824
22	201467286689315906290	201467286689324924928	9018638
23	1925320391606803968923	1925320391606799433728	-4535195

Tabellen

24	18435599767349200867866	18435599767349154021376	-46846490
25	176846309399143769411680	176846309399143557103616	-212308064
26	1699246750872437141327603	1699246750872436312375296	-828952307
27	16352460426841680446427399	16352460426841660481077248	-19965350151
28	157589269275973410412739598	-	?
29	1520698109714272166094258063		

Tabelle 32: Vergleich der exakten Pi-Funktion mit der Riemanns exakter Formel $\pi^*(n)$ (siehe (132), Summe über 1 Million Nullstellen, gerundet

n	$\pi(10^n)$	$\pi^*(10^n)$	$\pi^*(10^n) - \pi(10^n)$
0	0	-	-
1	4	4	0
2	25	25	0
3	168	168	0
4	1229	1229	0
5	9592	9592	0
6	78498	78498	0
7	664579	664579	0
8	5761455	5761456	1
9	50847534	50847536	2
10	455052511		
11	4118054813		
12	37607912018		
13	346065536839		
14	3204941750802		
15	29844570422669		
16	279238341033925		
17	2623557157654233		
18	24739954287740860		
19	234057667276344607		
20	2220819602560918840		
21	21127269486018731928		
22	201467286689315906290		
23	1925320391606803968923		
24	18435599767349200867866	18435599767349269364736	68496870
25	176846309399143769411680	176846309399144194637824	425226144
26	1699246750872437141327603	1699246750872437117681664	23645939
27	16352460426841680446427399		
28	157589269275973410412739598	-	?
29	1520698109714272166094258063		

Die Übereinstimmung des analytisch berechneten Wertes mit dem exakten Wert $\pi(10^{26})$ ist beachtlich: Der Wert ist auf 17 Dezimalstellen genau! Trotzdem ist das Ergebnis im Vergleich mit der ‚normalen‘ Riemann-Funktion (14 Stellen Genauigkeit) nur um drei Dezimalstellen besser (obwohl die Summenterme der ersten 1.000.000 Nullstellen der Zetafunktion ausgewertet wurden).

 20.10.2 MERSENNE-PRIMZAHLEN

Diese Tabelle enthält alle derzeit (Stand: Dez. 2020) bekannten Primzahl-Exponenten.

Nr.	p (Exponent)	Ziffern in M_p	Jahr	Entdecker
1	2	1	----	----
2	3	1	----	----
3	5	2	----	----
4	7	3	----	----
5	13	4	1456	anonymous
6	17	6	1588	Cataldi
7	19	6	1588	Cataldi
8	31	10	1772	Euler
9	61	19	1883	Pervushin
10	89	27	1911	Powers
11	107	33	1914	Powers
12	127	39	1876	Lucas
13	521	157	1952	Robinson
14	607	183	1952	Robinson
15	1279	386	1952	Robinson
16	2203	664	1952	Robinson
17	2281	687	1952	Robinson
18	3217	969	1957	Riesel
19	4253	1281	1961	Hurwitz
20	4423	1332	1961	Hurwitz
21	9689	2917	1963	Gillies
22	9941	2993	1963	Gillies
23	11213	3376	1963	Gillies
24	19937	6002	1971	Tuckerman
25	21701	6533	1978	Noll & Nickel
26	23209	6987	1979	Noll
27	44497	13395	1979	Nelson & Slowinski
28	86243	25962	1982	Slowinski
29	110503	33265	1988	Colquitt & Welsh
30	132049	39751	1983	Slowinski

Tabellen

31	216091	65050	1985	Slowinski
32	756839	227832	1992	Slowinski & Gage et al.
33	859433	258716	1994	Slowinski & Gage
34	1257787	378632	1996	Slowinski & Gage
35	1398269	420921	1996	Armengaud, Woltman, et al. (GIMPS)
36	2976221	895932	1997	Spence, Woltman, et al. (GIMPS)
37	3021377	909526	1998	Clarkson, Woltman, Kurowski et al. (GIMPS, PrimeNet)
38	6972593	2098960	1999	Hajratwala, Woltman, Kurowski et al. (GIMPS, PrimeNet)
39	13466917	4053946	2001	Cameron, Woltman, Kurowski et al. (GIMPS, PrimeNet)
40	20996011	6320430	2003	Shafer, Woltman, Kurowski et al. (GIMPS, PrimeNet)
41	24036583	7235733	2004	Findley, Woltman, Kurowski et al. (GIMPS, PrimeNet)
42	25964951	7816230	2005	Nowak, Woltman, Kurowski et al. (GIMPS, PrimeNet)
43	30402457	9152052	2005	Cooper, Boone, Woltman, Kurowski et al. (GIMPS, PrimeNet)
44	32582657	9808358	2006	Cooper, Boone, Woltman, Kurowski et al. (GIMPS, PrimeNet)
45	37156667	11185272	2008	Elvenich, Woltman, Kurowski et al. (GIMPS, PrimeNet)
46	42643801	12837064	2009	Strindmo, Woltman, Kurowski et al. (GIMPS, PrimeNet)
47	43112609	12978189	2008	Smith, Woltman, Kurowski et al. (GIMPS, PrimeNet)
48	57885161	17425170	2013	Cooper, Woltman, Kurowski et al. (GIMPS, PrimeNet)
??	74207281	22338618	2016	Cooper, Woltman (Prime95), Kurowski & Blosser (PrimeNet), GIMPS et al.
??	77232917	23249425	2017	GIMPS / Jon Pace (Prime95)
??	82589933	24862048	2018	GIMPS / Patrick Laroche (Prime95)

20.10.3 FERMAT-PRIMZAHLEN

Die gegenwärtig fünf bekannten Fermat-Primzahlen lauten:

3, 5, 17, 257, 65537

20.10.4 ENTARTUNG VON TYP4-OCRONs UND -EOCRONS

Tabelle 33: Die ersten 23 Typ4 OCRONs, GOCRONs (GC) sowie ihrer Entartungen

n	GC("*"="0,"P"=1,"2"=2,"^"=3), OCRON	n	GC("*"="0,"P"=1,"2"=2,"^"=3), OCRON
2	2 2	13	609 2P2*P 657 22P*P
3	9 2P	14	2584 22*P2* 2692 222*P* 2740 222^P* 2776 22^P2*
4	40 22* 43 22^	15	2404 2PP2P* 2452 2P2PP*
5	37 2PP	16	651 22*2^ 675 222*^ 687 222^^ 699 22^2^ 2680 22P^2* 2716 222P^* 10376 22*2*2* 10400 22*22** 10412 22*22^* 10760 222**2* 10784 222*2** 10880 2222*** 10928 2222^** 10952 222^*2* 10976 222^2** 11144 22^2*2* 11168 22^22** 11180 22^22^*
6	152 2P2* 164 22P*	17	645 22*PP 693 22^PP
7	161 22*P 173 22^P	18	2488 2P2^2* 2668 22P2^* 9764 2P2*2P* 9800 2P2P*2* 9824 2P2P2** 9872 2P22P** 10532 22P*2P* 10640 22P2P**
8	167 22P^ 648 22*2* 672 222** 684 222^* 696 22^2*	19	669 22P^P 2593 22*2*P 2689 222**P 2737 222^*P 2785 22^2*P
9	155 2P2^ 612 2P2P*	20	9608 2PP2*2* 9632 2PP22** 9644 2PP22^* 10388 22*2PP* 10568 22PP*2* 10592 22PP2** 10832 222PP**

Tabellen

			11156 22^2PP*
10	600 2PP2* 660 22PP*	21	9860 2P22*P* 9908 2P22^P* 10340 22*P2P* 11108 22^P2P*
11	149 2PPP	22	2392 2PPP2* 2644 22PPP*
12	2440 2P2*2* 2464 2P22** 2476 2P22^* 2596 22*2P* 2632 22P*2* 2656 22P2** 2704 222P** 2788 22^2P*	23	621 2P2^P 2449 2P2P*P

Tabelle 34: Anzahl der Entartungen bei EOCRONs vom Typ 4

n	Entart.	n	Entart.	n	Entart.	n	Entart.
1	-	26	13	51	10	76	185
2	-	27	10	52	63	77	10
3	1	28	63	53	39	78	85
4	4	29	3	54	106	79	3
5	1	30	36	55	4	80	693
6	5	31	1	56	311	81	39
7	3	32	271	57	30	82	13
8	16	33	4	58	13	83	2
9	3	34	13	59	3	84	594
10	5	35	10	60	260	85	10
11	1	36	159	61	11	86	32
12	26	37	13	62	5	87	10
13	3	38	40	63	45	88	134
14	13	39	10	64	1139	89	61
15	4	40	134	65	10	90	231
16	68	41	3	66	36	91	24
17	3	42	85	67	10	92	52
18	23	43	8	68	36	93	4
19	10	44	26	69	8	94	10
20	26	45	19	70	85	95	30
21	10	46	10	71	13	96	3508
22	5	47	2	72	997	97	2
23	2	48	693	73	5	98	111
24	134	49	15	74	55	99	19
25	3	50	23	75	19	100	159

Die Entartung bei EOCRONs Typ 4 ist deutlich höher als bei ‚normalen‘ OCRONs Typ 4 (siehe Tabelle 20).

20.10.5 NULLSTELLEN DER RAMANUJANSCHEN TAU-L-FUNKTION

Tabelle 35: Die ersten 128 Nullstellen der Ramanujanschen Tau-L-Funktion entlang der kritischen Geraden $\text{Re}(s) = 6$

n	n-te Nullstelle
1	9.222379399921084797142611932940781116486
2	13.907549861392134005200205137953162193298
3	17.44277697823447326186396821867674589157
4	19.65651314195496013326192041859030723572
5	22.33610363720986669022749993018805980682
6	25.27463654811236537511831556912511587143
7	26.80439115835040198021488322410732507706
8	28.83168262418687532999683753587305545807
9	31.17820949836026045431935926899313926697
10	32.77487538223120822067357948981225490570
11	35.19699584121007518433543737046420574188
12	36.74146297671030936271563405171036720276
13	37.75391597562427392631434486247599124908
14	40.21903437422132299161603441461920738220
15	41.73049228930784693147870711982250213623
16	43.59174123557517077642842195928096771240
17	45.04007921377559853226557606831192970276
18	46.19731875314330693527153925970196723938
19	48.35905247802367057374794967472553253174
20	49.27605353655818021252343896776437759399
21	51.15656028143634870275491266511380672455
22	53.06671423542580612320307409390807151794
23	54.09995263156227451872837264090776443481
24	55.21778745348462535957878571934998035431
25	56.71529404472536839421081822365522384644
26	58.58016100791407154702028492465615272522
27	59.78593800331714191997889429330825805664
28	61.13672295792679989290263620205223560333
29	62.66499232630715710001823026686906814575
30	64.08664571892624906013224972411990165710
31	64.84864127982825721119297668337821960449
32	66.49476926718958225137612316757440567017
33	67.93860977475046070139796938747167587280
34	69.04339787488993351871613413095474243164
35	71.11465341424647590429231058806180953979
36	71.74750419616562169267126591876149177551
37	72.81406066758940198724303627386689186096
38	74.09582544001794701671315124258399009705
39	75.77216168976411836410989053547382354736
40	77.10183189348964560849708504974842071533
41	77.68461125026033187168650329113006591797
42	79.79293909123566663765814155340194702148
43	80.56019206809750698994321282953023910522
44	82.00757620451852858423080760985612869263
45	82.84252583957207605180883547291159629822
46	83.97564035576498042701132362708449363708
47	85.46221814858006382564781233668327331543
48	86.75597218825528500474320026114583015442
49	88.07513099425673885889409575611352920532

Tabellen

50	89.02289034074360074555443134158849716187
51	90.45103289616260155980853596702218055725
52	91.11271853147249544235819485038518905640
53	92.44292549472127973331225803121924400330
54	93.76912394743676770758611382916569709778
55	95.13807853977348827356763649731874465942
56	95.62492107704515831301250727847218513489
57	97.34104088984686597996187629178166389465
58	98.70980408818076057286816649138927459717
59	99.74664890030413744170800782740116119385
60	100.22461499968198950227815657854080200195
61	101.34359353371037570923363091424107551575
62	103.16663591563629154279624344781041145325
63	103.81733899744642712903441861271858215332
64	105.22181333799052538324758643284440040588
65	106.29382213420061020769935566931962966919
66	107.42670755392653347826126264408230781555
67	108.47543790163686594496539328247308731079
68	109.39169607602677558588766260072588920593
69	110.70966268400202636712492676451802253723
70	111.53473540163911081890546483919024467468
71	112.75715359897023404300853144377470016479
72	113.84343404772059216156776528805494308472
73	115.06276556053481385788472834974527359009
74	116.46348398369597987311863107606768608093
75	117.11654084727238966934237396344542503357
76	118.14687073684822848917974624782800674438
77	119.08216779664660123216890497133135795593
78	119.99454209523629799605259904637932777405
79	121.78633067852094029603904346004128456116
80	122.55731782502655846656125504523515701294
81	123.21241716312161429414118174463510513306
82	124.60624049116798062186717288568615913391
83	125.94289344930038510028680320829153060913
84	126.75939204586923381157248513773083686829
85	127.55580316015350206271250499412417411804
86	128.62383894451065202702011447399854660034
87	129.60342208412549780405242927372455596924
88	130.94859240739617689541773870587348937988
89	131.70819904811898481966636609286069869995
90	132.96854278614409849978983402252197265625
91	134.34729668877156427697627805173397064209
92	135.07869588873938937467755749821662902832
93	135.55289998752846258867066353559494018555
94	137.09033471100445922274957410991191864014
95	137.70022292031720212435175199061632156372
96	139.28400855168445104936836287379264831543
97	139.93658439005704963165044318884611129761
98	140.89653322681010649830568581819534301758
99	142.1411519890185388703685021027922630310
100	143.0835552634784448855498339980840682983
101	144.3547263694031244085635989904403686523
102	145.1653120064068502870213706046342849731
103	146.1487705718024301404511788859963417053

104	146.4097883646259958823065971955657005310
105	148.1177541226128084872470935806632041931
106	149.0412678815713718449842417612671852112
107	150.2750742969780901603371603414416313171
108	150.9064237539794532949599670246243476868
109	152.1344343784803641028702259063720703125
110	153.1151471940314081621181685477495193481
111	154.0518290966241181649820646271109580994
112	154.7953122295758987547742435708642005920
113	155.7320793911374607887410093098878860474
114	157.0957831922944762936822371557354927063
115	157.9127528865146530279162107035517692566
116	158.6608139225808713490550871938467025757
117	159.6686139103367452207749010995030403137
118	161.3063702811864743580372305586934089661
119	161.8503586051299976134032476693391799927
120	162.8714549225416021727141924202442169189
121	163.5474941087671822970150969922542572021
122	164.3389052284337310538830934092402458191
123	165.6101228957916760009538847953081130981
124	166.5807970056847295836632838472723960876
125	167.6436347091075731441378593444824218750
126	168.6591247847260888192977290600538253784
127	169.2457741065447009987110504880547523499
128	170.5979320487521135873976163566112518311

Mathematica-Programm: [Bitte den Autor kontaktieren.](#)

20.10.6 ABC-VERMUTUNG: FITPARAMETER UND C₃-WERTE VON EBENENGLEICHUNGEN VERSCHIEDENER GÖDELISIERUNGSMETHODEN

Tabelle 36: $c=30011$. Fitparameter und c_3 der Ebenengleichungen für M_{abc} (Typ M2GOCRON4) für verschiedene Sätze von Gödelsymbolen

C	c_3	Codetable: Symbole/Werte	Max. Wert	Standardfehler	t-Statistik
30011	3.50329	1: {*, P, 2, ^}, {0, 1, 2, 3}	31.5607	0.00147067	2382.1
30011	3.31794	2: {*, P, 2, ^}, {0, 1, 3, 2}	31.8085	0.00203153	1633.22
30011	3.4018	3: {*, P, 2, ^}, {0, 2, 1, 3}	31.1111	0.00109664	3102.03
30011	3.01535	4: {*, P, 2, ^}, {0, 2, 3, 1}	31.7433	0.000640746	4706.
30011	3.1857	5: {*, P, 2, ^}, {0, 3, 1, 2}	30.9755	0.00200754	1586.86
30011	2.97014	6: {*, P, 2, ^}, {0, 3, 2, 1}	31.3842	0.000996648	2980.14
30011	4.00661	9: {*, P, 2, ^}, {1, 2, 0, 3}	30.2574	0.00451077	886.529
30011	3.03373	10: {*, P, 2, ^}, {1, 2, 3, 0}	31.6698	0.000607757	4991.67
30011	3.78849	11: {*, P, 2, ^}, {1, 3, 0, 2}	30.2873	0.00555072	682.522
30011	2.99422	12: {*, P, 2, ^}, {1, 3, 2, 0}	31.2772	0.00115591	2590.36
30011	4.50418	15: {*, P, 2, ^}, {2, 1, 0, 3}	30.242	0.00378901	1188.75
30011	3.35767	16: {*, P, 2, ^}, {2, 1, 3, 0}	31.6661	0.00190649	1761.18
30011	3.93106	17: {*, P, 2, ^}, {2, 3, 0, 1}	30.2885	0.00646281	608.258
30011	3.2632	18: {*, P, 2, ^}, {2, 3, 1, 0}	30.611	0.0024814	1315.06
30011	4.61984	21: {*, P, 2, ^}, {3, 1, 0, 2}	29.8592	0.00406558	1136.33

Tabellen

30011	3.58239	22: {*, P, 2, ^}, {3, 1, 2, 0}	31.2661	0.0011851	3022.85
30011	4.25748	23: {*, P, 2, ^}, {3, 2, 0, 1}	29.886	0.00587147	725.113
30011	3.5185	24: {*, P, 2, ^}, {3, 2, 1, 0}	30.6002	0.0016776	2097.34

Tabelle 37: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für M_{abc} (Typ M2GOCRON4) für verschiedene Sätze von Gödelsymbolen

C	c_3	Codetable: Symbole/Werte	Max. Wert	Standardfehler	t-Statistik
10009	3.50252	1: {*, P, 2, ^}, {0, 1, 2, 3}	27.4048	0.00255229	1372.31
10009	3.31693	2: {*, P, 2, ^}, {0, 1, 3, 2}	27.6525	0.00352484	941.016
10009	3.40264	3: {*, P, 2, ^}, {0, 2, 1, 3}	26.9558	0.00190137	1789.57
10009	3.01507	4: {*, P, 2, ^}, {0, 2, 3, 1}	27.6497	0.00111371	2707.22
10009	3.18746	5: {*, P, 2, ^}, {0, 3, 1, 2}	26.9829	0.00348145	915.555
10009	2.97092	6: {*, P, 2, ^}, {0, 3, 2, 1}	27.4021	0.00172728	1719.99
10009	4.00661	9: {*, P, 2, ^}, {1, 2, 0, 3}	26.104	0.00790261	506.999
10009	3.0335	10: {*, P, 2, ^}, {1, 2, 3, 0}	27.6466	0.00105616	2872.2
10009	3.79832	11: {*, P, 2, ^}, {1, 3, 0, 2}	26.278	0.0097115	391.116
10009	2.9951	12: {*, P, 2, ^}, {1, 3, 2, 0}	27.398	0.00200372	1494.77
10009	4.50975	15: {*, P, 2, ^}, {2, 1, 0, 3}	26.0831	0.00668297	674.813
10009	3.35675	16: {*, P, 2, ^}, {2, 1, 3, 0}	27.5776	0.00330739	1014.92
10009	3.94228	17: {*, P, 2, ^}, {2, 3, 0, 1}	26.277	0.0113075	348.642
10009	3.26532	18: {*, P, 2, ^}, {2, 3, 1, 0}	26.9845	0.00430428	758.62
10009	4.62653	21: {*, P, 2, ^}, {3, 1, 0, 2}	25.7004	0.00719384	643.124
10009	3.58183	22: {*, P, 2, ^}, {3, 1, 2, 0}	27.2467	0.00205594	1742.19
10009	4.26777	23: {*, P, 2, ^}, {3, 2, 0, 1}	25.9801	0.0102978	414.436
10009	3.51983	24: {*, P, 2, ^}, {3, 2, 1, 0}	26.8782	0.00291095	1209.17

Tabelle 38: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für M_{abc} (Typ EGOCRON4) für verschiedene Sätze von Gödelsymbolen

C	c_3	Codetable: Symbole/Werte	Max. Wert	Standardfehler	t-Statistik
10009	2.1217	1: {*, P, 2, ^}, {0, 1, 2, 3}	23.2525	0.00256872	825.975
10009	1.93604	2: {*, P, 2, ^}, {0, 1, 3, 2}	23.5014	0.00353924	547.021
10009	2.02181	3: {*, P, 2, ^}, {0, 2, 1, 3}	23.1332	0.00187808	1076.53
10009	1.63421	4: {*, P, 2, ^}, {0, 2, 3, 1}	23.4314	0.00112142	1457.26
10009	1.80654	5: {*, P, 2, ^}, {0, 3, 1, 2}	23.526	0.00345596	522.731
10009	1.59005	6: {*, P, 2, ^}, {0, 3, 2, 1}	23.5452	0.00169719	936.872
10009	2.61236	9: {*, P, 2, ^}, {1, 2, 0, 3}	23.1025	0.00785712	332.484
10009	1.65067	10: {*, P, 2, ^}, {1, 2, 3, 0}	23.3575	0.00106003	1557.19
10009	2.40797	11: {*, P, 2, ^}, {1, 3, 0, 2}	23.5053	0.00967604	248.859
10009	1.61167	12: {*, P, 2, ^}, {1, 3, 2, 0}	23.5442	0.00198794	810.723
10009	3.08406	15: {*, P, 2, ^}, {2, 1, 0, 3}	22.414	0.00649123	475.111
10009	1.97171	16: {*, P, 2, ^}, {2, 1, 3, 0}	23.3565	0.00330899	595.863
10009	2.5431	17: {*, P, 2, ^}, {2, 3, 0, 1}	23.5043	0.0112604	225.844
10009	1.87643	18: {*, P, 2, ^}, {2, 3, 1, 0}	23.5239	0.00430664	435.707
10009	3.18294	21: {*, P, 2, ^}, {3, 1, 0, 2}	22.4109	0.0068767	462.858

10009	2.19211	22: {*, P, 2, ^}, {3, 1, 2, 0}	22.9524	0.00204121	1073.92
10009	2.8499	23: {*, P, 2, ^}, {3, 2, 0, 1}	23.0993	0.0101915	279.636
10009	2.12547	24: {*, P, 2, ^}, {3, 2, 1, 0}	23.1287	0.00293149	725.048

Tabelle 39: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für M_{abc} (Typ EGOCRON4) für verschiedene Sätze von Gödelsymbolen: Reihenfolge: Reversed

C	c_3	Codetable: Symbole/Werte	Max. Wert	Standardfehler	t-Statistik
10009	2.43885	1: {*, P, 2, ^}, {0, 1, 2, 3}	22.2941	0.00330491	737.948
10009	2.48199	2: {*, P, 2, ^}, {0, 1, 3, 2}	22.2949	0.0025768	963.203
10009	1.86469	3: {*, P, 2, ^}, {0, 2, 1, 3}	22.9589	0.00150375	1240.03
10009	1.94351	4: {*, P, 2, ^}, {0, 2, 3, 1}	22.9598	0.00109807	1769.94
10009	1.5486	5: {*, P, 2, ^}, {0, 3, 1, 2}	23.3565	0.00118561	1306.16
10009	1.58908	6: {*, P, 2, ^}, {0, 3, 2, 1}	23.3575	0.00199429	796.813
10009	1.81771	9: {*, P, 2, ^}, {1, 2, 0, 3}	23.0656	0.000933021	1948.2
10009	1.97304	10: {*, P, 2, ^}, {1, 2, 3, 0}	23.0668	0.00316066	624.249
10009	1.51979	11: {*, P, 2, ^}, {1, 3, 0, 2}	23.4296	0.00127364	1193.27
10009	1.63535	12: {*, P, 2, ^}, {1, 3, 2, 0}	23.4318	0.00446254	366.462
10009	2.27073	15: {*, P, 2, ^}, {2, 1, 0, 3}	22.657	0.00209251	1085.17
10009	2.46931	16: {*, P, 2, ^}, {2, 1, 3, 0}	22.6589	0.00254729	969.387
10009	1.54154	17: {*, P, 2, ^}, {2, 3, 0, 1}	23.499	0.00276984	556.546
10009	1.62921	18: {*, P, 2, ^}, {2, 3, 1, 0}	23.5001	0.00550417	295.996
10009	2.25579	21: {*, P, 2, ^}, {3, 1, 0, 2}	22.8032	0.00147073	1533.78
10009	2.42527	22: {*, P, 2, ^}, {3, 1, 2, 0}	22.8045	0.00365269	663.97
10009	1.83374	23: {*, P, 2, ^}, {3, 2, 0, 1}	23.2557	0.00223614	820.047
10009	1.93552	24: {*, P, 2, ^}, {3, 2, 1, 0}	23.2561	0.00520616	371.774

Tabelle 40: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für M_{abc} (Typ M2GOCRON4) für verschiedene Sätze von Gödelsymbolen; Reihenfolge: Reversed

C	c_3	Codetable: Symbole/Werte	Max. Wert	Standardfehler	t-Statistik
10009	4.85748	1: {*, P, 2, ^}, {0, 1, 2, 3}	25.5946	0.00684048	710.109
10009	4.90055	2: {*, P, 2, ^}, {0, 1, 3, 2}	25.4929	0.00652405	751.152
10009	4.28317	3: {*, P, 2, ^}, {0, 2, 1, 3}	26.0164	0.00620476	690.305
10009	4.36192	4: {*, P, 2, ^}, {0, 2, 3, 1}	25.937	0.00610748	714.192
10009	3.967	5: {*, P, 2, ^}, {0, 3, 1, 2}	26.3287	0.00614129	645.955
10009	4.00745	6: {*, P, 2, ^}, {0, 3, 2, 1}	26.3326	0.00633565	632.524
10009	3.54826	9: {*, P, 2, ^}, {1, 2, 0, 3}	26.8814	0.00192201	1846.12
10009	3.6121	10: {*, P, 2, ^}, {1, 2, 3, 0}	26.8521	0.00222321	1624.72
10009	3.36945	11: {*, P, 2, ^}, {1, 3, 0, 2}	27.0258	0.00278201	1211.16
10009	3.41428	12: {*, P, 2, ^}, {1, 3, 2, 0}	27.0297	0.00325271	1049.67
10009	3.37628	15: {*, P, 2, ^}, {2, 1, 0, 3}	27.2372	0.00192779	1751.37
10009	3.42781	16: {*, P, 2, ^}, {2, 1, 3, 0}	27.2175	0.00241658	1418.46
10009	3.0478	17: {*, P, 2, ^}, {2, 3, 0, 1}	27.4339	0.00142776	2134.67
10009	3.06892	18: {*, P, 2, ^}, {2, 3, 1, 0}	27.4352	0.0023285	1317.99
10009	3.09633	21: {*, P, 2, ^}, {3, 1, 0, 2}	27.5812	0.00291311	1062.9

Mathematica-Programme

10009	3.13022	22: {*, P, 2, ^}, {3, 1, 2, 0}	27.5819	0.00355751	879.89
10009	2.93734	23: {*, P, 2, ^}, {3, 2, 0, 1}	27.6553	0.00149217	1968.51
10009	2.95744	24: {*, P, 2, ^}, {3, 2, 1, 0}	27.6556	0.00252606	1170.77

20.10.7 REED JAMESON PSEUDO-PRIMZAHLEN

Bisher sind folgende Reed Jameson Pseudoprimzahlen bekannt:

```
4.647.272.200.763.653
13.145.972.926.201.741
37.550.172.530.083.333
91.475.036.245.333.333
138.059.041.752.628.921
1.017.051.023.982.373.381
1.198.917.598.782.691.327
2.193.915.384.965.973.241
3.451.615.699.229.107.381
3.512.610.370.112.161.753
4.595.180.567.858.094.061
6.048.451.215.682.221.781
6.338.484.791.054.344.501
7.928.915.800.561.771.753
8.145.180.508.453.751.953
8.791.425.219.802.647.241
9.298.405.698.887.024.981
9.538.676.189.678.282.653
10.465.926.737.075.038.153
10.672.259.013.245.100.833
10.832.491.549.192.774.861
10.877.405.928.733.495.009
10.956.794.257.273.312.801
11.422.820.349.626.091.841
11.555.150.568.592.132.153
13.383.002.224.373.603.221
14.127.279.039.356.766.601
17.487.206.393.334.007.501
```

Quelle: Peter Danzeglocke (berechnet mit einem optimierten C++-Programm).

Der Bereich $n < 10^{10}$ enthält keine Reed Jameson Pseudoprimzahlen (Stand: Dezember 2018).

20.11 MATHEMATICA-PROGRAMME

In diesem Abschnitt findet man eine Sammlung von Mathematica-Programmen, so z. B. auf Geschwindigkeit optimierte Versionen der Beispiel-Programme weiter oben.

Tschebyschew-Funktion Psi(x):

Die Funktion `myPsi[x]` lässt sich etwas schneller machen, indem wir Symmetrieeigenschaften ausnutzen und die Funktionen `Evaluate[]` und `Compile[]` benutzen:

```
#####
myPsi[x_]:=Evaluate[-2*Sum[ ((x)^ZetaZero[i])/ZetaZero[i], {i,1,15}]-
0.5*Log[1-1/x^2]+x-Log[2*Pi]];
myPsic=Compile[{{x,_Complex}},myPsi[x],CompilationOptions-
>{"ExpressionOptimization"->True},
CompilationOptions->{"InlineCompiledFunctions"->Auto}]
Timing[Plot[Re[myPsic[x]],{x,1,100}]]
```

20.11.1 VERGLEICH DER ANZAHL VON PRIMZAHL-ZWILLINGEN, -COUSINS UND SEXY-PRIMZAHLEN MIT DER FORMEL VON HARDY-LITTLEWOOD

```
#####
Mathematica-Programm: Bitte den Autor kontaktieren.
```

20.11.2 RG-FOLGEN

RG-Folgen mit ‚Prime-GOCRONs‘ (Typ6):

Mathematica-Programm: Bitte den Autor kontaktieren.

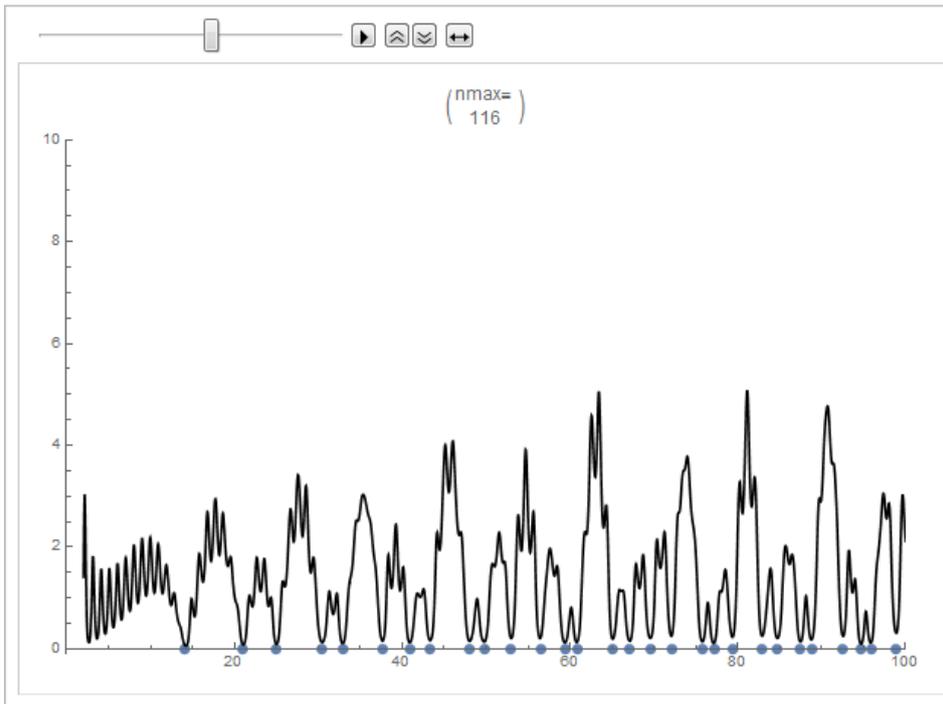
RG-Folgen mit ‚EGOCRONs‘ (Typ4):

(*Die folgenden Beispiele benötigen die OCRON-Bibliothek (s. unten*)

(*Beispiel:*)
Mathematica-Programm: Bitte den Autor kontaktieren.

20.11.3 RIEMANNSCHE ZETA-FUNKTION

```
#####
(*Animation des ‚Rauschens‘ der einfachen Produktdarstellung im
Komplexen:*)
Mathematica-Programm: Bitte den Autor kontaktieren.
(* (Snapshot:*)
```



```
#####
(*Iterative, approximative Methode zur Berechnung der
Produktdarstellung, ueber Primzahlen*)
(*entlang der kritischen Geraden *)
Mathematica-Programm: Bitte den Autor kontaktieren.
```

```
#####
(*Iterative, approximative Methode zur Berechnung der
Produktdarstellung, ueber Nullstellen der Zeta-Funktion*)
(*entlang der reellen X-Achse Geraden, Nullstellen bei Primzahlen *)
Mathematica-Programm: Bitte den Autor kontaktieren.
```

```
#####
(*Parametrischer 3D-Plot der Riemannsches Zeta-Funktion auf der krit.
Linie *)
Mathematica-Programm: Bitte den Autor kontaktieren.
```

20.11.4 REED JAMESON UND PERRIN-FOLGEN

```
(*#####*)
Mathematica-Programm: Bitte den Autor kontaktieren.
```

Mathematica-Programm: Bitte den Autor kontaktieren.

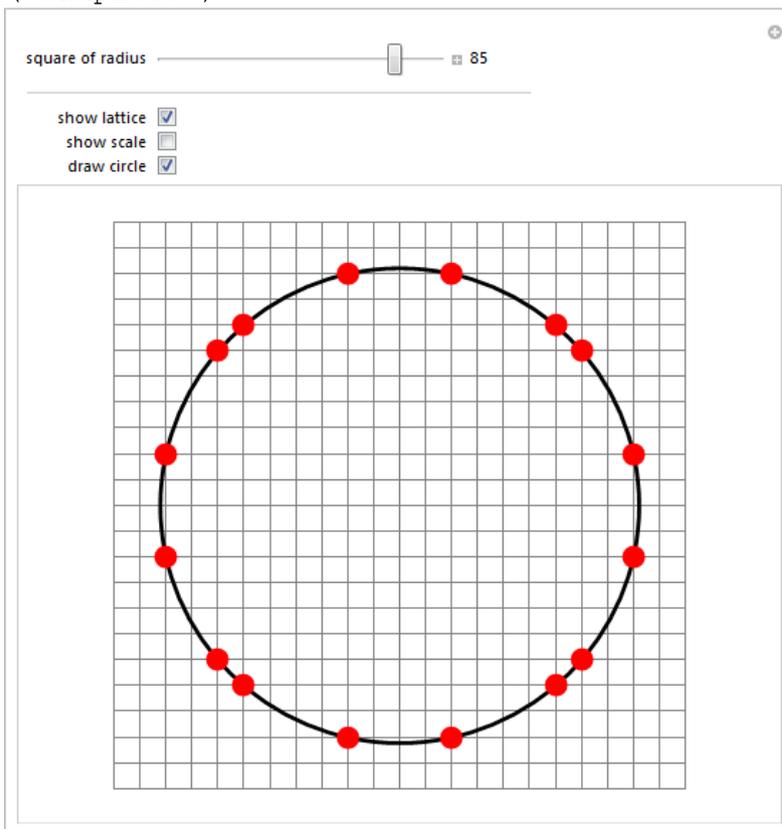
20.11.5 GITTERPUNKTE AUF N-SPHERES (N-DIMENSIONALEN KUGELN)

```
#####
(*Interaktive animation: Gitterpunkte auf 1-Sphere*)
m=Manipulate[Graphics[
```

```

dim=2;sqN=Sqrt[n];sqNInt=Round[sqN];
numberOfGridPoints=SquaresR[dim,n];
If[numberOfGridPoints>0,sol=FindInstance[a^2+b^2==n,{a,b},
Integers,numberOfGridPoints]];
Flatten[Table[{},{x,-sqNInt-2,sqNInt+2},
{y,-sqNInt-2,sqNInt+2}]],
Prolog->{If[ci,{{Black,Thickness[0.007],Circle[{0,0},sqN]}},
If[numberOfGridPoints>0,{Red,PointSize[0.04],
Point[{a,b]}/.sol]}],{}},
Frame->If[ft,Automatic,False],
PlotRange->{{-sqNInt-2,sqNInt+2},{-sqNInt-2,sqNInt+2}},
FrameTicks->If[ft,Automatic,None],
ImageSize->{480,400},ImageMargins->10,
GridLines->If[lattice,{Range[-sqNInt-2,sqNInt+2],
Range[-sqNInt-2,sqNInt+2]}],
{{n,10,"square of radius"},2,100,1,
Appearance->"Labeled"},Delimiter,{{lattice,True,"show
lattice"},{True,False}},
{{ft,False,"show scale"},{True,False}},{{ci,True,"draw
circle"},{True,False}},
AutorunSequencing->Automatic]
(*snapshot:*)

```



```

Export["C:\\animations\\latticePointsOnN-
spheresInNDimensions\\latticePointsOn1-spheresIn2Dimensions_RQ2-
100.mov",m]

```

```

#####
(*Integer Grid-Points, touching the surface of a sphere for a
given square of radius*)

```

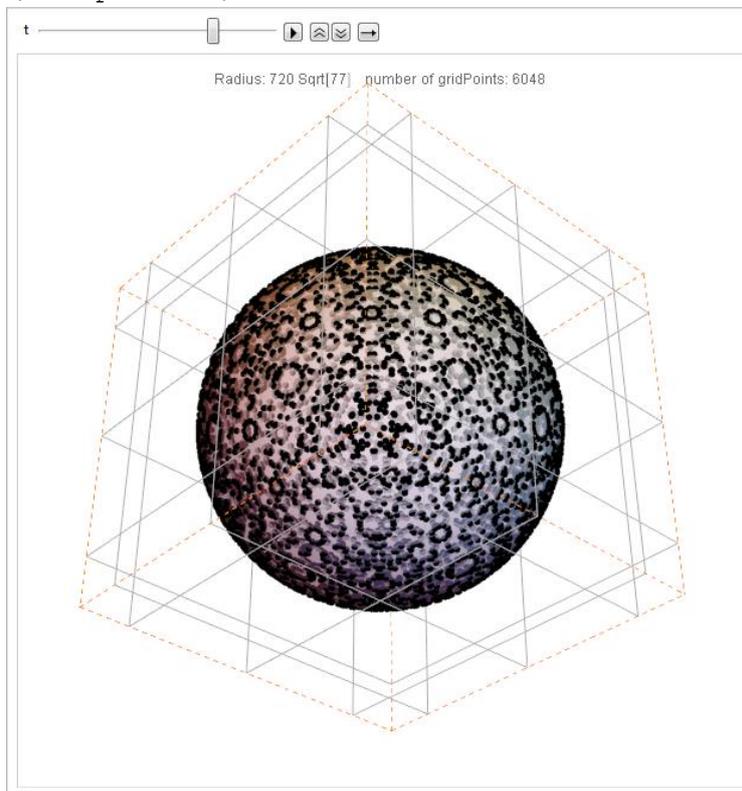
Mathematica-Programm: Bitte den Autor kontaktieren.

Mathematica-Programme

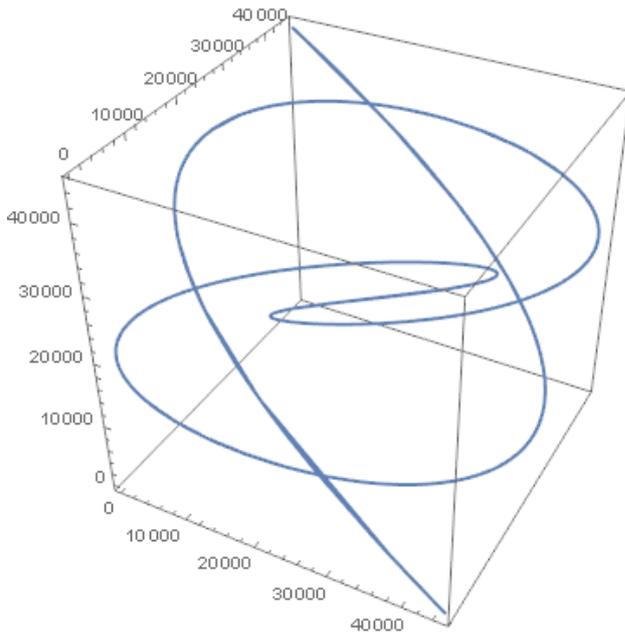
```
#####  
(* Gitterpunkte auf Oberfläche einer dreidimensionalen Kugel*)  
(*Die Winkel der Kugelkoordinaten interpretiert als 2 dimensionale  
kartesische Koordinaten*)  
Mathematica-Programm: Bitte den Autor kontaktieren.
```

```
#####  
(*Integer Grid-Points, touching the surface of a sphere for a  
given square of radius*)  
Mathematica-Programm: Bitte den Autor kontaktieren.
```

(*Snapshot:*)



(*used viewvector:*)



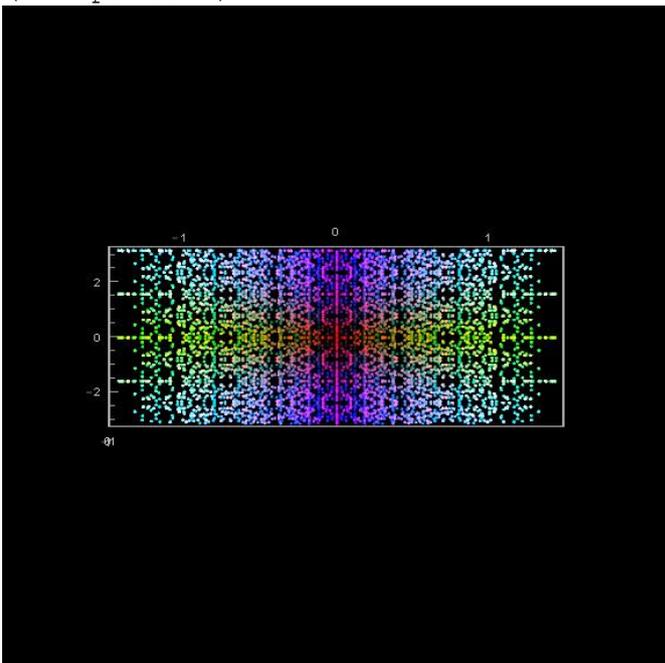
(*#####*)
 (*Abbildung 86*)
 (*3D Plots of Glomes, interpreting Phi, Theta and Psi as cartesian Coordinates *)

Mathematica-Programm: Bitte den Autor kontaktieren.

 (*Journey through the surface of a 4-dim Sphere *)
 (*Animation: 3D Plot of Glome, interpreting Phi, Theta and Psi as cartesian coordinates *)

Mathematica-Programm: Bitte den Autor kontaktieren.

(*Snapshot:*)



20.11.6 AUSWERTUNG UND STATISTIK FÜR PRIMZAHL-DIFFERENZEN

(*Statistics with primenumber differences of higher order*)

Mathematica-Programm: Bitte den Autor kontaktieren.

20.11.7 ABC-VERMUTUNG

```
(*abc-Vermutung: Berechnet Logarithm. Goedel-GOCRON4-Codes von abc-
Punkten. Und stellt diese *)
(*als 3DPlot zusammen mit einer 'gefitteten' Ebene in verschiedenen
Ansichten dar*)
(* Zur Ausführung wird die Ocron Mathematica-Bibliothek benötigt! *)
Mathematica-Programm: Bitte den Autor kontaktieren.
```

20.11.8 SONSTIGE MATHEMATICA-PROGRAMME

```
(*Polynom mit 26 Variablen vom Grad 25, dessen positive Werte mit
Primzahlen identisch sind*)
(*Dieses Programm sucht nach positiven Lösungen *)
c0=w z+h+j-q;
c1=(g k+2g+k+1) * (h+j) +h-z;
c2=2n+p+q+z-e;
c3=16(k+1)^3 * (k+2) * (n+1)^2+1-f^2;
c4=e^3*(e+2) * (a+1)^2+1-o^2;
c5=(a^2-1) *y^2+1-x^2;
c6=16r^2 y^4*(a^2-1)+1-u^2;
c7=((a+u^2*(u^2-a))^2-1) * (n+4d y)^2+1-(x+c u)^2;
c8=n+1+v-y;
c9=(a^2-1) *l^2+1-m^2;
c10=a i+k+1-l-i;
c11=p+1*(a-n-1)+b*(2a n+2a-n^2-2n-2)-m;
c12=q+y*(a-p-1)+s*(2a p+2a-p^2-2p-2)-x;
c13=z+p l*(a-p)+t*(2a p-p^2-1)-p m;
k=0;
FindInstance[Element[k+2, Primes]&& c0==0&& c1==0&& c2==0&& c3==0&& c4==0&& c
5==0&& c6==0&& c7==0&& c8==0&& c9==0&& c10==0&& c11==0&& c12==0&& c13==0&& a>=0
&& b>=0&& c>=0&& d>=0&& e>=0&& f>=0&& g>=0&& h>=0&& i>=0&& j>=0&& k>=0&& l>=0&& m>
=0&& n>=0&& o>=0&& p>=0&& q>=0&& r>=0&& s>=0&& t>=0&& u>=0&& v>=0&& w>=0&& x>=0&&
y>=0&& z>=0, {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}, Integ
ers]

(*Berechnung der Sigma1-Funktion*)
myDivisorSigma[k_, n_] :=
Sum[m^(k-1) Sum[Cos[(2 Pi j n)/m], {j, 1, m}], {m, 1, n}]
(*Beispiel: n= 31*)
myDivisorSigma[1, 31]
```

Man sieht es diesem Ausdruck nicht an, dass er identisch mit 32 ist:

$$\begin{aligned}
& -2(-19 + 2\sin\left(\frac{\pi}{14}\right) - 2\sin\left(\frac{3\pi}{14}\right) - 2\sin\left(\frac{\pi}{18}\right) + \sin\left(\frac{\pi}{22}\right) - \sin\left(\frac{3\pi}{22}\right) + \sin\left(\frac{5\pi}{22}\right) - \sin\left(\frac{\pi}{26}\right) + \sin\left(\frac{3\pi}{26}\right) \\
& \quad - \sin\left(\frac{5\pi}{26}\right) + \sin\left(\frac{\pi}{30}\right) - \sin\left(\frac{7\pi}{30}\right) - \sin\left(\frac{\pi}{34}\right) + \sin\left(\frac{3\pi}{34}\right) - \sin\left(\frac{5\pi}{34}\right) + \sin\left(\frac{7\pi}{34}\right) \\
& \quad + \sin\left(\frac{\pi}{38}\right) - \sin\left(\frac{3\pi}{38}\right) + \sin\left(\frac{5\pi}{38}\right) - \sin\left(\frac{7\pi}{38}\right) + \sin\left(\frac{9\pi}{38}\right) - \sin\left(\frac{\pi}{42}\right) - \sin\left(\frac{5\pi}{42}\right) \\
& \quad + \sin\left(\frac{\pi}{46}\right) - \sin\left(\frac{3\pi}{46}\right) + \sin\left(\frac{5\pi}{46}\right) - \sin\left(\frac{7\pi}{46}\right) + \sin\left(\frac{9\pi}{46}\right) - \sin\left(\frac{11\pi}{46}\right) - \sin\left(\frac{\pi}{50}\right) \\
& \quad + \sin\left(\frac{3\pi}{50}\right) + \sin\left(\frac{7\pi}{50}\right) - \sin\left(\frac{9\pi}{50}\right) + \sin\left(\frac{11\pi}{50}\right) + \sin\left(\frac{\pi}{54}\right) + \sin\left(\frac{5\pi}{54}\right) - \sin\left(\frac{7\pi}{54}\right) \\
& \quad - \sin\left(\frac{11\pi}{54}\right) + \sin\left(\frac{13\pi}{54}\right) - \sin\left(\frac{\pi}{58}\right) + \sin\left(\frac{3\pi}{58}\right) - \sin\left(\frac{5\pi}{58}\right) + \sin\left(\frac{7\pi}{58}\right) - \sin\left(\frac{9\pi}{58}\right) \\
& \quad + \sin\left(\frac{11\pi}{58}\right) - \sin\left(\frac{13\pi}{58}\right) + 2\cos\left(\frac{\pi}{7}\right) + 2\cos\left(\frac{\pi}{9}\right) - 2\cos\left(\frac{2\pi}{9}\right) + \cos\left(\frac{\pi}{11}\right) - \cos\left(\frac{2\pi}{11}\right) \\
& \quad + \cos\left(\frac{\pi}{13}\right) - \cos\left(\frac{2\pi}{13}\right) + \cos\left(\frac{3\pi}{13}\right) + \cos\left(\frac{\pi}{15}\right) - \cos\left(\frac{2\pi}{15}\right) + \cos\left(\frac{\pi}{17}\right) - \cos\left(\frac{2\pi}{17}\right) \\
& \quad + \cos\left(\frac{3\pi}{17}\right) - \cos\left(\frac{4\pi}{17}\right) + \cos\left(\frac{\pi}{19}\right) - \cos\left(\frac{2\pi}{19}\right) + \cos\left(\frac{3\pi}{19}\right) - \cos\left(\frac{4\pi}{19}\right) + \cos\left(\frac{\pi}{21}\right) \\
& \quad - \cos\left(\frac{2\pi}{21}\right) - \cos\left(\frac{4\pi}{21}\right) + \cos\left(\frac{5\pi}{21}\right) + \cos\left(\frac{\pi}{23}\right) - \cos\left(\frac{2\pi}{23}\right) + \cos\left(\frac{3\pi}{23}\right) - \cos\left(\frac{4\pi}{23}\right) \\
& \quad + \cos\left(\frac{5\pi}{23}\right) + \cos\left(\frac{\pi}{25}\right) - \cos\left(\frac{2\pi}{25}\right) + \cos\left(\frac{3\pi}{25}\right) - \cos\left(\frac{4\pi}{25}\right) - \cos\left(\frac{6\pi}{25}\right) + \cos\left(\frac{\pi}{27}\right) \\
& \quad - \cos\left(\frac{2\pi}{27}\right) - \cos\left(\frac{4\pi}{27}\right) + \cos\left(\frac{5\pi}{27}\right) + \cos\left(\frac{\pi}{29}\right) - \cos\left(\frac{2\pi}{29}\right) + \cos\left(\frac{3\pi}{29}\right) - \cos\left(\frac{4\pi}{29}\right) \\
& \quad + \cos\left(\frac{5\pi}{29}\right) - \cos\left(\frac{6\pi}{29}\right) + \cos\left(\frac{7\pi}{29}\right))
\end{aligned}$$

Deshalb:

```
FullSimplify[myDivisorSigma[1, 31]]
32
```

Oder noch besser:

```
N[myDivisorSigma[1, 31]]
32.
```

20.11.9 OCRONS- UND ABC-VERMUTUNG: PROGRAMM-BIBLIOTHEK

20.11.9.1 OCRON-ROUTINEN

```
(* #####*)
(* Library: GOCRON-Routines actual version Sept.2016*)
(* #####*)
Mathematica-Programm: Bitte den Autor kontaktieren.
(* OCRON-Functionlist:
nToGoedelSymbolList[n_],goedelSymbolListToN[symbolList_]
nToGoedelSymbolListForPrimeOCRONS[n_],goedelSymbolListToNForPrimeOCRONS[symbolList_]
nToGoedelSymbolListForVirtualOCRONS[n_],goedelSymbolListToNForVirtualOCRONS[symbolList_]
##### OCRON4, GOCRON4 #####
nToOCRON4[n_], oCRON4ToN[symbolList_], oCRON4ToNMaxVal[symbolList_,maxVal_]
nToGOCRON4[n_], gOCRON4ToN[n_], gOCRON4ToNMaxVal[symbolList_,maxVal_]
checkOCRON4[n_]
##### M2OCRON4, M2GOCRON4 without leading 2 #####
nToM2OCRON4[n_], m2OCRON4ToN[symbolList_]
nToM2GOCRON4[n_], msGOCRON4ToN[n_]
##### EOCRON4, EGOCRON4 #####
nToEOCRON4[n_],eOCRON4ToN[symbolList_], eOCRON4ToNMaxVal[symbolList_,maxVal_]
nToEGOCRON4[n_],eGOCRON4ToN[n_], eGOCRON4ToNMaxVal[symbolList_,maxVal_]
##### PrimeOCRON, PrimeGOCRON (Typ 6)#####
nToPrimeOCRON[n_], primeOCRONTON[n_]
nToPrimeGOCRON[n_],primeGOCRONTON[n_]
##### Sonstige #####
createAscendingEOcron4List[n_]
createAscendingEVirtualOcron4List[n_]
createAscendingVirtualOcron4List[n_]
createAllValuesListFromAscendingVirtualOcron4s[n_]
createIntValuesListFromAscendingVirtualOcron4s[n_]
createAscendingIntList[n_]
createAscendingOCRONListFromNaturalNumbers[n_]
createAscendingGOCRONListFromNaturalNumbers[n_]
createAscendingEOCRONListFromNaturalNumbers[n_]
createAscendingEGOCRONListFromNaturalNumbers[n_]
createAscendingM2OCRONListFromNaturalNumbers[n_]
createAscendingM2GOCRONListFromNaturalNumbers[n_]
resetGloc4Codes[]
setGloc4CodeSymbols[symbols_]
setGloc4CodeValues[values_]
##### Virtual OCRONS #####
checkVirtualOCRON4[n_], virtualOCRON4ToOCRON4[symbolList_]
virtualOCRON4ToN[symbolList_]
#####
Evaluating OCRONS by converting the polish RPN-representation used in OCRONS to 'normal'
Mathematica expressions before numerical evaluation #####
#####
oCRON4ToExpression[symbolList_]
logOCRON4ToExpression[symbolList_]
logOCRON4ToExpressionSimplify[symbolList_]
oCRON4ToExpressionPowerExpand[symbolList_]
logOCRON4ToExpressionPowerExpand[symbolList_]
logLogOCRON4ToExpressionPowerExpand[symbolList_]
convertOcronToTraditionalForm[symbolList_]
convertLogOcronToTraditionalForm[symbolList_]
convertLogLogOcronToTraditionalForm[symbolList_]
*)
```

20.11.9.2 ABC-VERMUTUNG

```
(*#####*)
(*#####*)
```

(*radicals, quality, isPossibleABC, radABC() computes radABC from c, but does not always get the smallest one!!*)

(*minRadABC takes always the smallest*)

20.11.9.3 ENTARTUNG VON OCRONS

```
(*#####*)
(*Degeneration-values of OCRONS Typ 4 *)
(*Needs GOCRON4-Library *)
(*#####*)
gloc4Codes={{{"*", "P", "2", "^"}, {0,1,2,3}}; (*actual code-Table, *)
(*Note: maxEGocrons should be at least 13 symbols long (e.g.  $22 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^7 = 128$ ) *)
(* Because of
goedelSymbolListToN[{"^", "^", "^", "^", "^", "^", "^", "^", "^", "^", "^", "^", "^"}]=671
08863 *)
(* maxEOcrons should be at least 67.108.863 to get all degeneration values up
to maxValue = 128 (=2^7 *)
(* maxEOcrons should be at least 1073741823 to get all degeneration values up
to maxValue = 256 (=2^8 *)
maxValue=128; maxEOcrons=67200000;
ocron4DegenList=Table[{}, {i,1,maxValue}];
For[i=1,i<maxEOcrons,i++,
eOcron=nToGoedelSymbolList[i];
iValue=oCRON4ToNMaxVal[eOcron,maxValue];
If[iValue>0&& iValue<= maxValue,AppendTo[ocron4DegenList[[iValue]],eOcron];
];
If[Mod[i,100000]==0,Print[N[i/67200000]]];
]
For[i=1,i<maxValue,i++,
Print[i,": ",Length[ocron4DegenList[[i]]],"->",ocron4DegenList[[i]]];
]
```

20.11.10 SOUND-ROUTINEN

```
(*#####Prime-Sound-Library#####*)
(*Generate a sortet list of the combined (sievenumber, primenumber)-
pairs by: *)
Mathematica-Programm: Bitte den Autor kontaktieren.
```

20.11.11 RSA-VERSCHLÜSSELUNG UND ENTSCHLÜSSELUNG

```
(*Example1: Encode/Decode a number (1115) *)
(*very simple example for the RSA encryption*)
(*Without Encoding/Decoding Functions from Mathematica...*)
(*##### implement coding mechanism #####*)
(*choose two different prime numbers:*)
Mathematica-Programm: Bitte den Autor kontaktieren.

(*Example2: same as Example 1: Encode a number (1115) *)
(*using Mathematica built-in functions*)
(*publicKey[], privateKey[], Encrypt[], Decrypt[]*)
(*used padding-mode: ,none` *)
(*IMPORTANT: In this Mathematica-version (10.3) Encoding with
PublicKey-Objects and padding: "None" only works for up to 16 Bit
Modules *)
Mathematica-Programm: Bitte den Autor kontaktieren.
(*##### Encode and decode messages:#####*)
(*this is our message to be encoded:*)
Mathematica-Programm: Bitte den Autor kontaktieren.
```

Anhang

PrivateKey [ cipher: RSA
private exponent length: 8 bits
public modulus length: 12 bits
padding: None
public exponent: 17
private exponent: 157
public modulus: 2773]

PublicKey [ cipher: RSA
public modulus length: 12 bits
padding: None
public exponent: 17
public modulus: 2773]

(*Example3: RSA-Encoding/Decoding a small String ("OK") using PKCS1 padding*)

(*let Mathematica choose p, q and the modulus using Mathematica built-in functions*)

Mathematica-Programm: Bitte den Autor kontaktieren.

<|"PrivateKey" → PrivateKey [ cipher: RSA
public modulus length: 97 bits
private exponent length: 93 bits
padding: PKCS1
public exponent: 65537
public modulus: 12202433704 <<7>> 277596949541
private exponent: 76245427803 <<5>> 502629985493] , "

PublicKey" → PublicKey [ cipher: RSA
public modulus length: 97 bits
padding: PKCS1
public exponent: 65537
public modulus: 12202433704 <<7>> 277596949541]]>

(*Example4: Hacking a private Key from a public key using PKCS1 padding with key length 192*)

(*We use Mathematica built-in functions PrivateKey[], Decrypt[] and FactorInteger[]*)

(*#####*)

Mathematica-Programm: Bitte den Autor kontaktieren.

(*##### decoding can be done different ways:#####*)

(* Decrypted data by using Decrypt[] will not contain padded data...*)

```

bCryptArray=ByteArray[IntegerDigits[mCrypt,256]];
decryptedByteArray=Normal[Decrypt[privKey,bCryptArray]]
decryptedString=FromCharCode[decryptedByteArray];
Print["Decryption-result (original String: ",decryptedString];
(**** program - output:****)
prime p from RSA:module: 68357071940820194611682396513
prime q from RSA:module: 78553627484042565312533006567
private Exponent:
484499185966049249555967871982611572207133532958607342401
Private Modulus:
5369695965139088101081485235420567443013865529391511497792
Hacked private key:
cipher: RSA
private exponent length: 192 bits
public modulus length: 192 bits
padding: PKCS1
public exponent: 65537

```

PrivateKey [ cipher: RSA
private exponent length: 192 bits
public modulus length: 192 bits
padding: PKCS1
public exponent: 65537
private exponent: 484499185966 <<34>> 958607342401
public modulus: 536969596513 <<34>> 315726900871]

```
Original Text as Bytearray including Bytes padded by PKCS1 algorithm:
{2,11,165,77,224,174,48,231,225,235,0,69,108,118,105,115,32,108,105,118,101,115,33}
\.02\.0bMà@0çáè\.00Elvis lives!
Decryption-result (original Byte array:
{69,108,118,105,115,32,108,105,118,101,115,33}
Decryption-result (original String):
Elvis lives!
```

```
(*Example5: Hacking a private Key from a public key using PKCS1
padding with key length 2048*)
(*We use Mathematica built-in functions PrivateKey[], Decrypt[] and
FactorInteger[]*)88
(*#####*)
Mathematica-Programm: Bitte den Autor kontaktieren.
(* Decrypted data by using Decrypt[] will not contain padded data...*)
bCryptArray=ByteArray[IntegerDigits[mCrypt,256]];
decryptedByteArray=Normal[Decrypt[privKey,bCryptArray]]
decryptedString=FromCharCode[decryptedByteArray];
Print["Decryption-result (original String: ",decryptedString];

(**** program - output:****)
prime p from RSA-module:
5042275217484184784387456407481025964634418009557323862771843210015347
0233709220326765100448150841802101002465172565326870447505988642493935
6768330261183984688981022399271959231632244880124488202703458535772508
3261691330915873078509567583024603043325764678776810906881522663421805
761981426998378611173580023640569
prime q from RSA-module:
6230506107037277994848859713460167565407435407248343949280547643268996
2794782739027185058685887929516793028417110464017369917383411955467390
4850630650913159085619714144991810427714356257581581356417704599361604
875399387751304617083538258312857625045844275691053230285222491700224
633245295180992749493823
private Exponent:
1030819360403968961043390501763010666063077815038257405347287921525051
2535022756352365421194768891684069779277718177933402369048702835912585
4131450497268501685537802072878657793084753037172733458691935491519525
1853460378496829887538632390637136150965931733628074554699429623111223
3269880545420815346132763817866426056363791352182879224677368503022618
4983798138489051261011671669601896311386511911972803990381100552366494
4073403141189139015889364169952790178286921216796288440782997538376575
```

⁸⁸ Das Programm läuft ab Mathematica Version 15.5 (2026)

Anhang

7085206627441700960078133155365855116355242551141828940174105853199096
690164133765434208900272472536995205015922393069952712705

Private Modulus:

3141592653589793238462643383279502884197169399375105820974944592307816
4062862089986280348253421170679821480865132823066470938446095505822317
2535940812848111745028410270193852110555964462294895493038196442881097
5665933446128475648233786783165271201909145648566923460348610454326648
2133936072602491412737302005743673942332300681176030308206877770767919
8534374004936614234231860407863629025266826226514213872656537709283991
0702130843755236406451881492103609092001021009355392277882966409625448
3914303698969808213385445154539250686410599473315757219688912541904259
662541240447603317926765114985912144304711024675664570896

Hacked private key:

cipher: RSA

private exponent length: 2047 bits

public modulus length: 2048 bits

padding: PKCS1

public exponent: 65537

PrivateKey [ cipher: RSA
private exponent length: 2047 bits
public modulus length: 2048 bits
padding: PKCS1
public exponent: 65537
private exponent: 10308193604 <<594>> 069952712705
public modulus: 31415926535 <<594>> 248437705287]

Original Text as Bytearray including Bytes padded by PKCS1 algorithm:

{2,169,246,29,163,145,193,96,236,157,15,189,194,238,0,73,102,32,68,111
,110,97,108,100,32,84,114,117,109,112,32,115,104,111,117,108,100,32,11
9,105,110,32,116,104,101,32,112,114,101,115,105,100,101,110,116,105,97
,108,32,101,108,101,99,116,105,111,110,115,44,32,116,104,105,115,32,11
9,111,117,108,100,32,98,101,32,97,32,100,105,115,97,115,116,101,114,32
,102,111,114,32,116,104,101,32,85,110,105,116,101,100,32,83,116,97,116
,101,115,32,111,102,32,65,109,101,114,105,99,97,46,10,39,116,119,97,11
5,32,98,114,105,108,108,105,103,44,32,97,110,100,32,116,104,101,32,115
,108,105,116,104,121,32,116,111,118,101,115,10,100,105,100,32,103,121,
114,101,32,97,110,100,32,103,105,109,98,108,101,32,105,110,32,116,104,
101,32,119,97,98,101,58,10,65,108,108,32,109,105,109,115,121,32,119,10
1,114,101,32,116,104,101,32,98,111,114,111,103,111,118,101,115,44,10,9
7,110,100,32,116,104,101,32,109,111,109,101,32,114,97,116,104,115,32,1
11,117,116,103,114,97,98,101}

\.02@ø\.\1dÉ`À`ì□\.\0f¼Ái\.\00If Donald Trump should win the presidential
elections, this would be a disaster for the United States of America.

'twas brillig, and the slithy toves

did gyre and gimble in the wabe:

All mimsy were the borogoves,

and the mome raths outgrabe

Decryption-result (original String: If Donald Trump should win the
presidential elections, this would be a disaster for the United States
of America.

'twas brillig, and the slithy toves

did gyre and gimble in the wabe:

All mimsy were the borogoves,

and the mome raths outgrabe

(* Berechnet Aliquot-Sequenzen für ein paar interessante Startwerte*)

Mathematica-Programm: Bitte den Autor kontaktieren.

Output:

```
{1, Terminating, {1, 0}}
{2, Terminating, {2, 1, 0}}
{3, Terminating, {3, 1, 0}}
{4, Terminating, {4, 3, 1, 0}}
{5, Terminating, {5, 1, 0}}
{6, Perfect, {{6}}}}
{7, Terminating, {7, 1, 0}}
{8, Terminating, {8, 7, 1, 0}}
{9, Terminating, {9, 4, 3, 1, 0}}
{10, Terminating, {10, 8, 7, 1, 0}}
{11, Terminating, {11, 1, 0}}
{12, Terminating, {12, 16, 15, 9, 4, 3, 1, 0}}
{28, Perfect, {{28}}}}
{496, Perfect, {{496}}}}
{220, Amicable, {{220, 284}}}}
{1184, Amicable, {{1184, 1210}}}}
{12496, Sociable, {{12496, 14288, 15472, 14536, 14264}}}}
{1264460, Sociable, {{1264460, 1547860, 1727636, 1305184}}}}
{790, Aspiring, {790, 650, 652, {496}}}}
{909, Aspiring, {909, 417, 143, 25, {6}}}}
{562, Cyclic, {562, {284, 220}}}}
{1064, Cyclic, {1064, 1336, {1184, 1210}}}}
{1488, Non-
terminating, {1488, 2480, 3472, 4464, 8432, 9424, 10416, 21328, 22320, 55056, 957
28, 96720, 236592, 459792, 881392, 882384, 1474608}}
```

(*Aliquot 276 (306, 396, 696)OE:*)

Mathematica-Programm: Bitte den Autor kontaktieren.

(*#####*)

(*Plot Differences of Log of Aliquot-Sequences, using ListPlot*)

n=921232;noIterate=1000; diffOrder=1;

Mathematica-Programm: Bitte den Autor kontaktieren.

20.11.13 DIE ARECIBO-NACHRICHT

(*Arecibo-Message*)

```
n=23;
t=Table[BitShiftRight[BitAnd[27886402056107263551714831669687744330301
5886191896083753494207226153602508928851994608485761550978983329982259
3335259720410959738432212343758921014182008038517667278025253709464080
0567916516636264434941344165234644984933485655114374616110243082450500
4833981684141550381731028954290673308020242293291528914499592811145845
8595397126461136347103419178098716188118662826517986311913829406689871
7096729057657705911386899499333419586667745206851413286366090402386362
1169622066629371322105035882727404788841080, 2^i], i], {i, 1679, 0, -
1}]; ArrayPlot[Partition[t, n], Mesh->All,
ColorRules->{1->RGBColor[.0, .1, .9], 0->RGBColor[.9, .5, .2]},
ImageSize->Medium, PlotLabel->{"Arecibo-Message"},
PlotLegends->Automatic]
```

20.11.14 KORRELATIONEN IN DEN LETZTEN ZIFFERN DER PRIMZAHLFOLGE

(*statistical properties and correlations, concerning the last digits in the prime sequence*)

(* one predecessor:*)

Mathematica-Programm: Bitte den Autor kontaktieren.

20.11.15 PRIMZAHL N-LINGE UND MAXIMALE PRIMZAHLDICHTE

```
(* Construction of a maximal prime number density *)
(* the generated Sequence of possible prime positions is identical with OEIS A020498 *)
(* 1,3,7,9,13,19,21,27,31,33,37,43,49,51,57,63,69,73,, , *)
(*The patterns resulting from the differences of the p-positions (generated by sieving)
repeat after the following cycles: *)
(*sieving up to 2: {2 } length of period: 1, primorial(1)=2 *)
(*RotateLeft[Differences[Select[Range[2+1],GCD[#,2]\[Equal]1&]],0]; *)
(*sieving up to 3: {2,4} length of period: 2, primorial(2)=6*)
(*RotateLeft[Differences[Select[Range[2*3+1],GCD[#,2*3]\[Equal]1&]],1]; *)
(*sieving up to 5: {2,4,2,4,6,2,6,4} length of period: 8, primorial(3)=30 *)
(*RotateLeft[Differences[Select[Range[2*3*5+1],GCD[#,2*3*5]\[Equal]1&]],2]; *)
(*sieving up to 7: {2,4,2,4,6,2,6,4,2,4,6,6,2,6,4,2,6,4,6,8,...2,10,2,10 } length of
period: 48, primorial(4)=210 *)
(*RotateLeft[Differences[Select[Range[2*3*5*7+1],GCD[#,2*3*5*7]\[Equal]1&]],1];*)
(*sieving up to 11:
{2,4,2,4,6,2,6,4,2,4,6,6,2,6,6,4,6,8,4,2,4,2,4,8,6,4,8,4,6,2,6,6,4,2,4,6,8,4,2,4,2,10,
...2,10,2,10 } length of period: 480, primorial(5)=2310*)
(*RotateLeft[Differences[Select[Range[2*3*5*7*11+1],GCD[#,2*3*5*7*11]\[Equal]1&]],262]*)
(*sieving up to 13:
{2,4,2,4,6,2,6,4,2,4,6,6,2,6,6,4,6,8,4,6,2,4,8,6,4,8,4,6,2,...2,10,2,10 } length of
period: 5760, primorial(6)=30030 *)
(*RotateLeft[Differences[Select[Range[2*3*5*7*11*13+1],GCD[#,2*3*5*7*11*13]\[Equal]1&]],
2899]*)
(*sieving up to 17: {... } length of period: 92160 , primorial(7)=510510 *)
(*RotateLeft[Differences[Select[Range[primorial[7]+1],GCD[#,primorial[7]]\[Equal]1&]],89
465]*)
(*sieving up to 19: {... } length of period: 1658880, primorial(8)=9699690 *)
(*RotateLeft[Differences[Select[Range[2*3*5*7*11*13*17*19+1],GCD[#,2*3*5*7*11*13*17*19]\[
Equal]1&]],??]*)
(* The period length can be easily calculated by the formula: a(0)=1;for
n>0,a(n)=(prime(n)-1)*a(n-1) *)
(* Mathematica: RecurrenceTable[{a[0]\[Equal]1,a[n]\[Equal](Prime[n]-1)a[n-1]},a,{n,10}]
*)
(* Arguments of Range[] and GCD[]: primorial[n]:
2,6,30,210,2310,30030,510510,9699690,... Mathematica:
primorial[n_]:=Product[Prime[i],{i,n}];*)
(* The Sequence {0,1,2,1,262,2899,89465...} of the arguments for the RotateLeft function
is unknown *)
(* The values for the corresponding RotateRight operators read:
{0,1,6,47,218,2861,2695,... *)
(* functions: *)
```

LITERATURVERZEICHNIS

Basieux, P. (2004). *Die Top Seven der mathematischen Vermutungen*. Gamburg: Rowohlt-Verlag.

Borwein. (2000). Computational strategies for the Riemann Zeta function. *J. Comp. App. Math.*

- Edwards, H. M. (1974). *Riemann's Zeta Function*. San Diego, CA: Academic Press Limited.
- Hardy, G. H., & Ramanujan, A. (1940 (First Edition) 1978 (Last Edition, corrected)).
Ramanujan: Twelve Lectures on subjects suggested by his life and work. Cambridge,
 New York: American Mathematical Society.
- Hofstadter, D. R. (1991 / 1985). *Gödel Escher Bach*. München: Ernst Klett Verlag.
- Johnson F.Yan, A. K. (Jan. 1991). Prime Numbers and the Amino Acid Code: Analogy in
 Coding Properties. *Journal of Theor. Biology*, S. 333-341.
- Koch, H., & Pieper, E. (1976). *Zahlentheorie: Ausgewählte Methoden und Ergebnisse*. Berlin:
 VEB Deutscher Verlag der Wissenschaften.
- Ribenboim, P. (1989). *The Book of Prime Number Records*. New York: Springer-Verlag.
- Richard Crandall, C. P. (2001). *Prime Numbers: A Computational Perspective*. New York:
 Springer-Verlag.
- Sauty, M. d. (2004). *Die Musik der Primzahlen*. München: Verlag C.H. Beck.
- Singh, S. (1998). *Fermats letzter Satz*. München, Wien: Carl Hanser Verlag.
- Tammet, D. (2014). *Die Poesie der Primzahlen*. München: Carl Hanser Verlag.
- Taschner, R. (2013). *Die Zahl, die aus der Kälte kam*. München: Carl-Hanser Verlag.
- Tegmark, M. (2015). *Unser Mathematisches Universum*. Berlin: Ullstein Buchverlage GmbH.

ABBILDUNGSVERZEICHNIS

Abbildung 1: Anzahl der Primzahl-Zwillinge von 2 bis 3500.....	22
Abbildung 2: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (ohne Vorgänger)	32
Abbildung 3: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger: ,1') 33	
Abbildung 4: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (alle mögl. Vorgänger).....	33
Abbildung 5: n-te Mersenne Primzahl (doppelt logarithmische Darstellung)	36
Abbildung 6: n-te Mersenne Primzahl (doppelt logarithmische Darstellung)	36
Abbildung 7: n-te Mersenne-Primzahl (doppelt logarithmische Phasenraumdarstellung)	37
Abbildung 8:Lissajous-ähnliche Darstellung mit Hilfe von Mersenne-Primzahl-Exponenten	39
Abbildung 9: GIMPS-Statistik des Autors.....	40
Abbildung 10: Seite aus , <i>Liber Abaci</i> von Leonardo Fibonacci.....	52
Abbildung 11: Reed Jameson Folge: Plot der 0-Positionen der Summe der Mod-Werte.....	60
Abbildung 12: Reed Jameson Folge: Plot der Summe der Mod-Werte.....	60

Abbildungsverzeichnis

Abbildung 13: Perrin-Folge: Plot der Mod-Werte. Die Nullstellen sind (fast immer) bei Primzahlen.....	61
Abbildung 14: Hofstadters Q-Folge: Eine Meta-Fibonacci-Folge.....	63
Abbildung 15: Carmichael-Zahlen bis 10000.....	64
Abbildung 16: Wilson-Quotienten der ersten 100 Primzahlen(logarithmische Darstellung).....	70
Abbildung 17: RG-Folge ,15' (logarithm. Plot).....	73
Abbildung 18: RG-Folge ,21' (logarithm. Plot).....	74
Abbildung 19: RG-Folge ,25' (logarithm. Plot).....	74
Abbildung 20: RG-Folge ,27' (logarithm. Plot).....	75
Abbildung 21: RG-Folge ,33' (logarithm. Plot).....	75
Abbildung 22: RG-Folge ,9' (logarithm. Plot).....	76
Abbildung 23: RG-Folgen in negativer Richtung (Typ EGOCRON4).....	77
Abbildung 24: RG-Folgen in positiver Richtung (Typ EGOCRON4): Sie enden immer bei 6.....	78
Abbildung 25: Parametrischer 3D-Plot (Re- und Im-Teil) der Zetafunktion mit Nullstellen.....	82
Abbildung 26: Parametrischer 3D-Plot (Abs- und Arg-Teil) der Zetafunktion mit Nullstellen.....	83
Abbildung 27: Betrag und Phase der Zetafunktion entlang der krit. Geraden (0-70).....	84
Abbildung 28: Vergleich der Phase der Zetafunktion mit der (negativen) Riemann-Siegel Funktion.....	84
Abbildung 29: Zeta-Funktion (auf der krit. Linie, $t=0-70$, Produktformel mit 100 Primzahlen).....	88
Abbildung 30: Zeta-Funktion (Betrag, auf krit. Linie, $t=0-70$, Produktformel mit 100 Primzahlen).....	89
Abbildung 31: Zeta-Funktion (Real- und Imaginärteil, krit. Linie, $t=0-70$, exakte Formel).....	89
Abbildung 32: Zeta-Funktion (Betrag, krit. Linie, $t=0-70$, exakte Formel).....	90
Abbildung 33: Graph nach Formel (62) mit $x_0 = 1, y_0 = 0$, Nullstellen: blaue Kreise.....	92
Abbildung 34: Zeta(s) nach (62) unter Verwendung der ersten 5 Primzahlen.....	93
Abbildung 35: Formel (63) (Betrag, $x=10-100$, Produkt mit 100 Nullstellen der Zetafunktion).....	94
Abbildung 36 Graph mit absoluten Minima bei Primzahlen mit $x_0 = 1, y_0 = 0$	95
Abbildung 37: Anzahl der Nullstellen der Zetafunktion von 0-200, (exakt und Näherung).....	96
Abbildung 38: Anzahl der Nullstellen der Zetafunktion von 0-200, (Formel (66) und Näherung).....	97
Abbildung 39: Nullstellen-Zählfunktion der Zetafunktion mit Primzahl-Summenterm.....	98
Abbildung 40: $R(x)-\pi(x)$, für Werte von $x=1$ bis 1000.....	103
Abbildung 41: Funktion $\omega(n)$, Anzahl verschiedener Primfaktoren (rot: asymptotisch).....	105
Abbildung 42: Funktion $\Omega(n)$, totale Anzahl Primfaktoren (rot: asymptotisch).....	105
Abbildung 43: Integer-Logarithmus: Summe aller Primzahlen der Zerlegung für n : $\text{sopfr}(n)$	106
Abbildung 44: Liouville Lambda-Funktion, von 1 bis 1000.....	107
Abbildung 45: Liouville-Funktion, von 1 bis 100: Exakt und analytisch berechnet.....	108
Abbildung 46: Tschebyschew Psi Funktion, dargestellt von 0 bis 100.....	109
Abbildung 47: Tschebyschew Psi-Funktion analytisch berechnet, dargestellt von 0 bis 100.....	110

Abbildung 48 Tschebyschew Psi Funktion, Vergleich analytische und zahlentheoretische Berechnungsmethode.....	110
Abbildung 49: Eulersche Phi-Funktion, dargestellt von 1 bis 100.....	111
Abbildung 50: Summatorische Funktion $\Phi(n)$ der Phi-Funktion, dargestellt von 1 bis 100 (Vergleich zahlentheoretische und analytische Berechnungsmethode).....	112
Abbildung 51: Euler Phi-Funktion (Vergleich analytische und zahlentheoretische Berechnung)	113
Abbildung 52: Anzahl-Teiler-Funktion $\sigma_0(n)$, dargestellt von 0 bis 50	116
Abbildung 53: Teilersummen-Funktion $\sigma_1(n)$, dargestellt von 0 bis 100.....	117
Abbildung 54 $ \sigma_1x $: Vergleich der analytisch berechneten Werte von Sigma mit den exakten Werten. Es wurden die Ramanujan-Summen $cq(n)$ bis $q=1000$ ausgewertet, n läuft von 0 bis 12.....	119
Abbildung 55: $\text{Arg}(\sigma_1(x))$: Phase der erweiterten Sigmafunktion. Es wurden die Ramanujan-Summen $cq(n)$ bis $q=100$ ausgewertet, n läuft von 0 bis 12. Die blaue Kurve wurde skaliert.	119
Abbildung 56 $ \sigma_1(x) $: Analytisch berechnete Werte von Sigma. Es wurden die Ramanujan-Summen $cq(n)$ bis $q=50$ ausgewertet, n läuft von 0 bis 5000	120
Abbildung 57 $ \sigma_1(x) $: Vergleich der analytisch berechneten Werte von Sigma mit den exakten Werten. Es wurden die Ramanujan-Summen $cq(n)$ bis $q=1000$ ausgewertet, n läuft von 100 bis 150.....	121
Abbildung 58: $ \sigma_1(x) -x-1$: Vergleich der analytisch berechneten Werte von Sigma mit den exakten Werten. Es wurden die Ramanujan-Summen $cq(n)$ bis $q=3000$ ausgewertet, n läuft von $1E9$ bis $1E9+10$. Die Nullstellen fallen mit Primzahlen zusammen.....	121
Abbildung 59: Sigma-Funktion analytisch mit Cos()-Termen berechnet.....	122
Abbildung 60: Ramanujan $\tau(n)$ (grau), in Rot: Nur n für $\text{Mod}(n, 112)=0$	123
Abbildung 61: Ramanujan-Tau-L-Funktion (Dirichlet-L-Reihe) 0-70, mit 34 Nullstellen, entlang der kritischen Geraden	125
Abbildung 62: Möbius-Funktion $\mu(n)$, von 1 bis 100	127
Abbildung 63: Mertens-Funktion $M(n)$ von 1 bis 400	127
Abbildung 64: Radikal(n) ($n=1,100$).....	129
Abbildung 65: Ramanujan-Summen $cq(n)$ von $q = 1$ bis 12 und n von 0 bis 17.....	133
Abbildung 66: Ramanujan-Summen $cq(n)$ von $q = 1$ bis 24 und n von 0 bis 24.....	134
Abbildung 67: Ramanujan-Summen $cq(x)$, analytisch fortgesetzt ($q = 1$ bis 6 und $x = 0$ bis 30)	136
Abbildung 68: Ramanujan-Summen $cq(x)$, analytisch fortgesetzt ($q = 7$ bis 12 und $x = 0$ bis 30)	137
Abbildung 69: Variante der $\mathfrak{Z}(x)$ nach (122) von 10 bis 100, mit Nullstellen (bzw. Minima) bei Primzahlen	142
Abbildung 70: Reed-Jameson-Funktion von 0 bis 100.....	143
Abbildung 71: Vergleich $\pi(n)$ mit Gausscher Näherung.....	145
Abbildung 72: Vergleich $\pi(n)$ mit Integrallogarithmus, dargestellt von 1 bis 1000	146
Abbildung 73: Vergleich $\pi(n)$ mit Riemann-Funktion $R(x)$, im Bereich von 0 bis 100.....	147
Abbildung 74: Vergleich $\pi(n)$ mit Riemann-Funktion $R(x)$, Bereich von 0 bis 1000.....	147
Abbildung 75: Riemanns exakte Formel für $\pi(x)$ (von $x = 1$ bis 25, mit Summe über die ersten 10 Nullstellenpaare der Zetafunktion).....	149
Abbildung 76: Riemanns exakte Formel für $\pi(x)$ (von $x = 25$ bis 50, mit Summe über die ersten 100 Nullstellenpaare der Zetafunktion)	149
Abbildung 77: n -te zusammengesetzte Zahl („Nicht-Primzahl“)	151
Abbildung 78: Gitterpunkte auf 1-Sphere, quadratischer Radius von 8 bis 13	154

Abbildungsverzeichnis

Abbildung 79: $r_2(n)$: Anzahl/4 der möglichen Darstellungen von n als Summe von 2 Quadraten. Mit Primzahlen (rote Punkte) ist kein Zusammenhang erkennbar.....	156
Abbildung 80: $r_2(n)$: Anzahl/4 der Darstellungen von n als Summe von 2 Quadraten (bis 100000)	156
Abbildung 81: $r_2 * (bn)$: Anzahl unterschiedlicher Darstellungen von bn als Summe von zwei Quadraten.....	157
Abbildung 82: Ganzzahlige Gitterpunkte einer Kugel mit (quadrierten) Radius 11!	159
Abbildung 83: Ganzzahlige Gitterpunkte von Kugeln (quad. Radius 999-102).....	160
Abbildung 84: $r_3(n)$: Anzahl/6 der möglichen Darstellungen von n als Summe von 3 Quadraten. Mit Primzahlen (rote Punkte) ist kein Zusammenhang erkennbar.....	162
Abbildung 85: $r_3(n)$: Anzahl/6 der Darstellungen von n als Summe von 3 Quadraten (bis 100000)	162
Abbildung 86: $r_3 * (n)$: Anzahl unterschiedlicher Darstellungen von n als Summe von drei Quadraten, ($n=1$ bis 500).....	163
Abbildung 87: $r_3 * (n)$: Anzahl unterschiedlicher Darstellungen von n als Summe von drei Quadraten, ($n=1$ bis 100000).....	163
Abbildung 88, Gitterpunkte auf Oberfläche einer Kugel mit $r_2 = 1001$, Winkel der Kugelkoordinaten als 2-dimensionale kartesische Koordinaten interpretiert (gleiche Farbe bedeutet gleiche Punkte hinsichtl. Spiegel-Operationen	164
Abbildung 89: wie oben, jedoch $r^2 = 11!$ (Mathematica-Programme im Anhang).....	164
Abbildung 90: $r_4(n)$: Anzahl/8 der möglichen Darstellungen von n als Summe von 4 Quadraten. Werte, die bei Primzahlen liegen, sind rot markiert.	167
Abbildung 91: $r_4(n)$: Anzahl/8 der Darstellungen von n als Summe von 4 Quadraten (bis 100000)	168
Abbildung 92: $r_4 * (n)$: Anzahl unterschiedlicher Darstellungen von n als Summe von vier Quadraten, ($n=1$ bis 500).....	169
Abbildung 93: Beispiel1: Gitterpunkte auf der 3-Sphere einer 4-dimension. Kugel mit $r^2=1001$	170
Abbildung 94: 6 Ansichten von Abbildung 93: Rechts/Links, vorne/hinten, Oben/unten	171
Abbildung 95: Beispiel2: Gitterpunkte auf 3-Sphere einer 4-dimensionalen Kugel mit $r^2=10007$	172
Abbildung 96: 2 Ansichten von Abbildung 95: Beispiel2: Gitterpunkte auf 3-Sphere einer 4-dimensionalen Kugel mit $r^2=10007$ Rechts/Links, vorne/hinten, Oben/unten.....	173
Abbildung 97: Entartung von wohlgeformten OCRON3s bis $n=768$ (logarithmisch dargestellt)	182
Abbildung 98: Längen der Typ4-OCRONS von Mersenne-Zahlen bis $M83$	185
Abbildung 99: Längen der Typ4-OCRONS von Wagstaff-Zahlen bis $W73$	186
Abbildung 100: Entartung von wohlgeformten OCRON4s bis $n=256$ (logarithmisch dargestellt)	192
Abbildung 101: Logarithmische Darstellung der Gödelnummern für die ersten 5000 natürlichen Zahlen bei Verwendung der GödelCodes “*”=0, “P”=1, “2”=2, sowie “^”=3	199
Abbildung 102: Logarithmische Darstellung der Gödelnummern für die ersten 200 natürlichen Zahlen bei Verwendung der GödelCodes “*”=0, “P”=1, “2”=2, sowie “^”=3	199
Abbildung 103: Entartung von wohlgeformten OCRON5s bis $n=128$	200
Abbildung 104: Prime-GOCRONS Typ6 ($n \rightarrow \text{GOCRON}[n]$) von 1 bis 10000	203
Abbildung 105: Inverse Prime-GOCRONS Typ6 ($\text{GOCRON} \rightarrow n$) von 1 bis 10000.....	205
Abbildung 106: Matrix: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 100000 Primzahlen	213

Abbildung 107: Diagramm: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 100000 Primzahlen.....	214
Abbildung 108: Matrix: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 1000000 Primzahlen.....	215
Abbildung 109: Diagramm: Häufigkeit der Regellängen bei 1. Ordnung Differenzsequenzen der ersten 1000000 Primzahlen.....	215
Abbildung 110: Anzahl der Regeln erstellt aus der Differenzen-Primzahlfolge 1. Ordnung in Abhängigkeit des Zahlenbereichs $10n$	216
Abbildung 111: Wie verhalten sich Regellängen bei Primzahl Differenz-Sequenzen n-ter Ordnung?.....	217
Abbildung 112: Matrix: Häufigkeit der Regellängen bei 14. Ordnung Differenzsequenzen der ersten 100000 Primzahlen.....	217
Abbildung 113: Max. Regellänge in Primzahl-Differenz-Sequenzen der Ordnung n für die ersten 106 Primzahlen.....	218
Abbildung 114: Sortierte Primzahl-Differenz-Werte der Ordnung 10 der ersten 100000 Primzahlen.....	219
Abbildung 115: : Sortierte Primzahl-Differenz-Werte der Ordnung 10 im mittleren Bereich (2000 Werte) der ersten 100000 Primzahlen.....	220
Abbildung 116: abc-Treffer: die ersten 91 möglichen c-Werte (9-10000).....	223
Abbildung 117: abc-Treffer: die ersten 868 möglichen c-Werte (9-1000000).....	224
Abbildung 118: max. 'Qualität' von abc-Tripeln in Abhängigkeit von c (mit 138 abc-Treffern) im Bereich bis 20000.....	224
Abbildung 119: <i>Mabc</i> : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10007$ (Primzahl), a und b sind teilerfremd (verschiedene Ansichten). Die 10002 Punkte liegen etwa auf einer Ebene.....	227
Abbildung 120: <i>Mabc</i> : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10008$, a und b sind teilerfremd (verschiedene Ansichten). Die 3310 Punkte liegen etwa in einer Ebene.....	228
Abbildung 121: <i>Mabc</i> : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10008$, (verschiedene Ansichten). Die 10004 Punkte sind räumlich verteilt.....	228
Abbildung 122: <i>Mabc</i> : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=100002$, (verschiedene Ansichten). Die 99998 Punkte liegen räumlich verteilt.....	229
Abbildung 123: <i>Mabc</i> : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=100002$, a und b teilerfremd (verschiedene Ansichten). Die 28558 Punkte liegen etwa auf einer Ebene.....	229
Abbildung 124: <i>Mabc</i> : Logarithm. Gödel-GOCRON4-Codes der abc-Punkte. $C=10007$ (Primzahl), a, b sind teilerfremd (verschiedene Ansichten). Die 100003 Punkte liegen etwa auf einer Ebene.....	230
Abbildung 125: Ebene von <i>Mabc</i> -Punkten für $c=100003$ (Primzahl).....	232
Abbildung 126: Primzahlsignal x_i mit Intervalllänge $M=216$. Rot: $M \ln M i$	238
Abbildung 127: Spektrale Leistungsdichte eines Primzahl-Signales (rot: $1k\alpha$ mit $\alpha=1.55$).....	239
Abbildung 128: Anfang des 4. Kapitels aus Leonhard Eulers Arbeit „Tentamen Novae Theoriae musicae“.....	251
Abbildung 129: 4. Kapitel, §14 aus Leonhard Eulers Arbeit „Tentamen Novae Theoriae musicae“.....	253
Abbildung 130: Leonhard Eulers Gradus Suavitatis im Bereich 1 bis 500.....	255
Abbildung 131: Primzahl-Rhythmen, mit den ersten 50 Primzahlen und A3 als tiefstem Ton.....	257
Abbildung 132: Mathematica-Soundobjekt (Primzahl-Song).....	258

Abbildungsverzeichnis

Abbildung 133: Melodyne erzeugt Primzahlrhythmen mit dem Sieb des Eratosthenes ...	258
Abbildung 134: Sternbild Herkules, mit Kugelsternhaufen M13 (Ziel der Arecibo-Botschaft).....	269
Abbildung 135: Arecibo-Message	270
Abbildung 136: Gauss'sche Primzahlen, mit Hilfe von Fouriertransformationen 'gefiltert'	273
Abbildung 137: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:1,x)	275
Abbildung 138: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:3,x)	276
Abbildung 139: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:7,x)	276
Abbildung 140: Wahrscheinlichkeiten der Endziffern in der Primzahlfolge (Vorgänger:9,x)	277
Abbildung 141: RG-Folgen von Typ4-EGOCRONs in Richtung positiver Indices (Werte 30-44).....	283
Abbildung 142: RG-Folgen von Typ4-EGOCRONs in Richtung positiver Indices (Werte 60-74).....	284
Abbildung 143: RG-Folgen von Typ4-EGOCRONs in Richtung positiver Indices (Werte 90-107)	285
Abbildung 144: Die ersten 12 Werte der Aliquot Sequenzen 276,306,396,696. Ab dem 3. Wert sind die Sequenzen identisch	293
Abbildung 145: Aliquot-Sequenz 276 (OE, die ersten 600 Werte).....	293
Abbildung 146: Aliquot-Sequenz 276 (OE, die ersten 1981 Werte)	294
Abbildung 147: Aliquot-Sequenz 552 (OE, die ersten 1126 Werte)	294
Abbildung 148: Aliquot-Sequenz 564 (OE, die ersten 3463 Werte)	295
Abbildung 149: Aliquot-Sequenz 660 (OE, die ersten 971 Werte).....	295
Abbildung 150: Aliquot-Sequenz 966 (OE, die ersten 948 Werte).....	296
Abbildung 151: Aliquot-Sequenz 840 (Ende bei 601, 746 Werte)	296
Abbildung 152 : Aliquot-Sequenz 1578 (OE, die ersten 7555 Werte)	297
Abbildung 153: Aliquot-Sequenz 921232 (Ende bei 11, 6358 Werte)	297
Abbildung 154: Aliquot-Sequenz 2856 (Ende bei Zyklus mit 28 Werten).....	298
Abbildung 155: Familie von Aliquot-Sequenzen (Sequenzende bei Primzahl 3)	299
Abbildung 156: Familie von Aliquot-Sequenzen (Sequenzende bei Primzahl 7)	300
Abbildung 157: Familie von Aliquot-Sequenzen (Sequenzende bei Primzahl 31)	301
Abbildung 158: Familie von Aliquot-Sequenzen (Sequenzende bei Primzahl 47)	301
Abbildung 159: Aliquot Sequenzlängen bis n=2500, OE-Sequenzen sind mit einer Länge von 10000 dargestellt.....	302
Abbildung 160: Endwerte von Aliquot-Sequenzen für Startwerte bis 2500	304
Abbildung 161: Startwerte von Aliquot-Sequenzen, die ein zyklisches Ende ergeben	304
Abbildung 162: Aliquot-Sequenz: logarith. Differenzen für Startwert n= 840, mit Vorzeichen	305
Abbildung 163; Aliquot: logarith. Differenzen für Startwert n= 840, ohne Vorzeichen.....	306
Abbildung 164: Aliquot: Differenzen der logarith. Werte für Startwert n= 840	307
Abbildung 165: Aliquot: Differenzen der logarith. Werte für Startwert n= 921232	307
Abbildung 166: Aliquot: Differenzen der logarith. Werte für Startwert n= 564 (OE)	308
Abbildung 167: FakFunc(x, 1037) mit Nullstellen bei Primfaktoren 17 und 61	314
Abbildung 168: Funktion f(x,y): Hat für jeden ganzzahligen (x-y)Punkt eine Nullstelle..	314
Abbildung 169: wie oben, jedoch Contour-Plot	315

TABELLENVERZEICHNIS

Tabelle 1: Anzahl der Primzahlzwillinge und Werte der Hardy-Littlewood-Funktion	21
Tabelle 2: Anzahl der Primzahl-Drillinge und Werte der Hardy-Littlewood-Funktion	24
Tabelle 3: Anzahl der Primzahl-Quadrupel und Werte der Hardy-Littlewood-Funktion : ..	25
Tabelle 4: Die Hardy-Littlewood-Konstanten	30
Tabelle 5: Anzahl der Primzahl-Quintupel und Werte der Hardy-Littlewood-Funktion	30
Tabelle 6: Anzahl der Primzahl-n-Tupel in Abhängigkeit von n	31
Tabelle 7: Glückliche Zahlen bis 1E15.....	43
Tabelle 8: Die ersten 10 vollkommene Zahlen	45
Tabelle 9: Anzahl der Sophie-Germain-Primzahlen bis 1E12	49
Tabelle 10: Einige linear rekursiv definierten Folgen	53
Tabelle 11: Wagstaff-Primzahlen: Exponent q	66
Tabelle 12: Verallgemeinerte Wieferich-Primzahlen	69
Tabelle 13: Die Zahlen 2 bis 50 in OCRON-Typ3-Darstellung.....	181
Tabelle 14: Die Zahlen 2 bis 49 in OCRON Typ4-Darstellung	183
Tabelle 15: Mersenne-Zahlen, sowie die Exponenten in OCRON Typ4-Darstellung	183
Tabelle 16: Mersenne-Zahlen in OCRON-Typ4-Darstellung	184
Tabelle 17: Wagstaff-Prim-Exponenten in OCRON-Typ4-Darstellung (Primzahlen in rot)	184
Tabelle 18: Wagstaff-Zahlen in OCRON-Typ4-Darstellung (Primzahlen in rot).....	185
Tabelle 19: Die ersten 100 Typ4-EOCRONs (in aufsteigender Reihenfolge).....	190
Tabelle 20: Entartungswerte der ersten 100 Typ4 OCRONs.....	191
Tabelle 21: Typ4-EOCRONs (Standarddarstellung) und Gödelnummern von 2 bis 100. GödelCodes:("*" =0, "P" =1, "2" =2, "^" =3)	196
Tabelle 22: Typ4-EOCRONs (Inverse Gödelisierung von $g=0$ bis 99.....	197
Tabelle 23: Prime-OCRONS (P und *-Operator) mit Gödelnummern (GCodes P=1, *=0)	201
Tabelle 24: Prime GOCRONS, OCRONS und dazu inverse Zahlen von 0 bis 99.....	204
Tabelle 25: $c=100003$. Fitparameter und c_3 der Ebenengleichungen für $Mabc$ (Typ GOCRON4) für verschiedene Sätze von Gödelsymbolen	231
Tabelle 26: Verschiedene c_3 -Werte für verschiedene GOCRONS und Gödelsymbole.....	232
Tabelle 27: Primzahl-Codierung der kanonischen Aminosäuren nach Yan et. al	237
Tabelle 28: Entartete virtuelle OCRONS. Primzahlen und -Potenzen in rot (Ordnung: exp)	287
Tabelle 29: Vergleich der exakten Pi-Funktion mit der Riemann-Funktion (gerundet) ...	315
Tabelle 30: Vergleich der exakten Pi-Funktion mit der Riemanns exakter Formel $\pi * (n)$ (siehe (132), Summe über 10000 Nullstellen, gerundet.....	316
Tabelle 31: Vergleich der exakten Pi-Funktion mit der Riemanns exakter Formel $\pi * (n)$ (siehe (132), Summe über 100000 Nullstellen, gerundet.....	317
Tabelle 32: Vergleich der exakten Pi-Funktion mit der Riemanns exakter Formel $\pi * (n)$ (siehe (132), Summe über 1 Million Nullstellen, gerundet.....	318
Tabelle 33: Die ersten 23 Typ4 OCRONS, GOCRONS (GC) sowie ihrer Entartungen	321
Tabelle 34: Anzahl der Entartungen bei EOCRONS vom Typ 4.....	322
Tabelle 35: Die ersten 128 Nullstellen der Ramanujanschen Tau-L-Funktion entlang der kritischen Geraden $Re(s) = 6$	323
Tabelle 36: $c=30011$. Fitparameter und c_3 der Ebenengleichungen für $Mabc$ (Typ M2GOCRON4) für verschiedene Sätze von Gödelsymbolen	325
Tabelle 37: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für $Mabc$ (Typ M2GOCRON4) für verschiedene Sätze von Gödelsymbolen	326

Tabelle 38: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für $Mabc$ (Typ EGOCRON4) für verschiedene Sätze von Gödelsymbolen	326
Tabelle 39: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für $Mabc$ (Typ EGOCRON4) für verschiedene Sätze von Gödelsymbolen: Reihenfolge: Reversed.....	327
Tabelle 40: $c=10009$. Fitparameter und c_3 der Ebenengleichungen für $Mabc$ (Typ M2GOCRON4) für verschiedene Sätze von Gödelsymbolen; Reihenfolge: Reversed	327

SACHVERZEICHNIS

12-Pence Münze	271	ECM-Methode	309
abc-Treffer	221	Edelsteine	34
abc-Vermutung	129, 221, 325	Einstein	206
Adenin	234	Elektromagnetische Wellen	267
Aliquot	76	Elliptische Jacobi-Funktion	158
Aliquot-Familie	298	Elliptische Kurven	18
Aliquot-Sequenzen	291	Elvis Presley	249
Aliquot-Summe	44	Energieniveaus	100
Allgemeine Relativitätstheorie	206	Energiezustände	99
Aminosäuren	235	Entartung	26, 180, 191, 192, 200
Andrew Odlyzko	100	Ente	272
Apostel	271	Entschlüsselungsexponent	243
Arecibo	269	Erhabene Zahl	271
ASCII Codes	240	Erschaffungsprozess	266
asymptotische Primzahl-Formel	147	Euklid	14
Atomkerne	99	Euklid-Mullin Sequenz	290
Außerirdische Lebensformen	267	Euler, Leonhard	250
Baumstruktur	298	Euler-Exponent	252
Bedeutungsebene	175, 193	Eulersche Phi-Funktion	111
befreundete Zahlen	76	Extraterrestrisch	267
berechenbare Probleme	290	F. Dyson	99
Bernoulli-Zahlen	85	Fahrstuhl	206
Blues-Schema	271	Faktorisierung	18
Brasilianische Zahlen	41	Faktorisierungsalgorithmus	245
Bruns Konstante	20	Faktorisierungsmethoden	311
C. F. Gauß	145	Farey-Sequenzen	126
Carmichael-Zahlen	64	Fermat	16, 40, 41, 113, 320
Catalansche Vermutung	69, 275	Fermat-Quotient	68
Chaos	8	Fermatsche Gleichung	47
Codon	235	Fibonacci-Primzahlen	18, 61
Cytosin	234	Fibonacci-Zahlen	49
Daniel, Arnaut	261	Kuriositäten	49
Dichtung	266	Fields-Medaille	209
diskrete Fouriertransformation	238	Forbes	29
DNA-Code	234	Thomas	25
DNA-Sequenzen	234	Foster, Jodie	268
Donald Duck	272	Freiheitsgrade	195
Donald Trump	341	Gandhi	138

ganzzahliger Logarithmus 106
 Gausschen Zahlenebene 273
[Gedankenexperiment](#) 206, 211, 237
 Gedichtform 261
 Gefühle 260
 geheimnisvoll 265
[gekrümmter Raum](#) 206
 Ghazel 264
 GIMPS 39
 Gitterpunkte 154
 Gleichnis 266
 Glomes *Siehe* Hyperkugeln
[GOCRONS](#) 175
 Ebene 226
 Gödel 79
[Gödel, Kurt](#) 175, 193
 Gödel-Codes 194
[Gödel-Escher-Bach](#) 193
[Gödelisierung](#) 175, 193
[Gödelnummer](#) 201
[Gödelnummerierung](#) 226
[Gödelnummern](#) 196
[Gödel-Transformation](#) 196
 Goldbach-Vermutung 279
 Goldener Schnitt 50
 Goodstein-Folge 177
 Gradus Suavitatis 106, 250, 252
 Gram-Funktion 103
[Gravitationsfeld](#) 206
 Gravitationswellen 267
 Guanin 234
 GUE-Hypothese 99
 GUE-Operator 101
 H. Montgomery 99
 Hacker 248
 Haikus 259
 Hardy-Littlewood
 Konstanten 30
 Näherung für Zwillingsprimzahlen 21
 Hauptzweig des komplexen Logarithmus
 148
 Heim, Burkhardt 271
 Hermitsche Operatoren 99
 Hilbert-Pólya-Vermutung 99
 Hofstadter 193
 Q-Folge 63
[Hyper-Kugelkoordinaten](#) 170
 Hyperkugeln 166
 Hyperkugeloberfläche 174
 Inhaltskette 291
 Inselbegabt 261
 Interstellare Botschaften 267
 J. S. Bach 9, 256
 Jambische Versmaß 261
 Jazz-Harmonik 255
 Kanonische Aminosäuren 237
 Klassenzahl 165
 kleiner Fermatsche Satz 113
 Knödel-Zahlen 64
 Koans 266
 Kommunikation 267
 Konsonanztheorie 250
[Kosmologen](#) 206
 Kryptographie 240
 Kugelkoordinaten 164
 Kugeln und Gitterpunkte 159
 Kugelpunkte 152
 Kugelsternhaufen 269
 Kunst 273
 Kunstwerk 265
 Kusszahl 271
[KVEC](#) 36
 Lauteinheiten 260
 Lehmer-Five 292
 Leonardo Fibonacci 52
 Liber Abaci 52
 Liouville-Funktion 106
 Lucas-Folge 53
 Lucas-Lehmer Test 34, 241
 Lücken 16
 Mächtige Zahlen 67
 Mangoldt-Funktion 108
[Markow-Kette](#) 212
 Mathematica 9
[MATHOrakel](#) 208, 209
[Matrix](#) 212
 ex nihilo 212
 Intelligenz der 212
 Regeln 212
 Übergangswahrscheinlichkeiten 212
[Matrix-Software](#) 212
 Matsuo Bashô 259
[Max Tegmark](#) 206
 Melodyne 258
 Menschheit 269
 Mersenne 16, 34, 44
 Mersenne-Primzahlen 319
 Mertens-Funktion 126
 Meta-Fibonacci Folgen 63
 MIDI 258
 Mihailescu 275
 Mills-Konstante 139
 Mills-Primzahlen 139
 Möbius-Funktion 103, 126
 Mochizuki, Shinichi 222
 Moren 260

- Multiversum 206
- Musik 250
- Musikalische Arithmetik 250
- Naturkonstanten 207
- N-Eck, regelmäßiges 42
- Neubäcker, Peter 258
- Neue Mersennesche Vermutung 35
- Neutronensterne 267
- Nicomachus 44
- Nucleotide 234
- Nucleotid-Zahlen 236
- numerische Evidenz 146
- OCRON Wesen 207
- OCRONianer 206
- OCRONS 175, 200
- OCRON-Wesen 206
- Öffentlicher Schlüssel 240
- Olympische Götter 271
- Omega-Funktionen 104
- Online Encyclopedia of Integer Sequences 315
- Online-Banking 239
- Operator-Sequenz 175, 180
- Orakellogorithmus 208
- P1-Test 34
- Paar-Korrelation 100
- Padovan-Folge 53
- Paralleluniversen 206
- Pell-Folge 53
- Perrin-Folge 53
- Perron's Formel 112
- PFLOP 40
- Philosophie 266
- Poesie 259
- Pollard ‚p-1‘ Methode 309
- Polnische Notation 175
- Polynom 25-ten Grades 139
- Prime-OCRONS 72
- Prim-Operator 179
- Primzahlen 8
 - arithmetische Folge 16
 - Cousin- 23
 - Drillinge 23
 - Endziffern von 32
 - Glückliche 42
 - Grundlegendes 14
 - höchstmögliche Dichte 26
 - Kehrwerte 16
 - Konstellationen von Primzahlen 280
 - Liste mit 14
 - Mersenne 34
 - Mirp-Primzahlen 65
 - N-linge 25
 - n-Tupel 26
 - Primzahl-Gödelisierung 194
 - Primzahl-Signal 237
 - Quintupel 30
 - Rhythmische Muster 256
 - Sexy 23
 - Sophie-Germain 47
 - Vierling 24
 - Wagstaff-Primzahlen 65
 - Wieferich-Primzahlen 67
 - Zwillinge 17, 20
 - Zwillingskonstante 20
- Primzahl-Experiment 238
- Primzahlsatz 144
- Primzahlzählfunktion 144
- Prim-Zeta-Funktion 80
- private key 240
- Privater Schlüssel 241
- Probabilistische Primzahltests 241
- PRP-Zahlen 241
- public key 240
- Pythagoreische Stimmung 251
- Quadranacci 56
- Quantenchaos 99
- Quanten-Computer 245
- Quantenmechanik 99, 207
- Quintinen 264
- Radikal 128
- Ramanujan 114, 117, 119, 122, 132
 - Tau-Dirichlet-L-Funktion 123
 - Tau-Funktion 123
 - Tau-L-Funktion 323
 - Tau-Theta-Funktion 123
 - Tau-Z-Funktion 123
- Ramanujan Summen-Funktion 117
- Ramanujan-Summen 129
- Reed Jameson Folge 53
- Reed-Jameson-Funktion 142
- Regelwerk 212
- Reim 260
- Reine Stimmung 251
- Rekursiv gödelisierte Zahlen 72
- RG-Folgen 73, 283
- Riemann-Funktion 103
- Riemanns Formel 148
- Riemannsche Vermutung 79
- Riemann-Siegel Funktionen 81
- Rilke, Rainer Maria 265
- RNA 235
- RSA 240
 - Modul 242
 - Padding 244
 - Verschlüsselung 240

Verschlüsselungsverfahren 240
 Sakrileg 52
 Schmetterlingstraum 265
 Schrödinger-Gleichung 101
 Science-Fiction 52
 Selbstbezüglichkeit 266
 Seele 256
 Selbstorganisation 239
 Sestine 261
 SETI-Projekt 267
 Sieb des Erathosthenes 256
 Sigma-Funktion *Siehe*
 Teilersummenfunktion
 Skiponacci 277
 Sonett 265
 Spektrale Eigenschaften 237
 Spektrale Leistungsdichte 239
 Stellenwertsysteme 196
 Sternbild 269
 Sterne 269
 Summendarstellung 176
 Symmetrische Verschlüsselung 240
 Tammet, Daniel 260
 Tankas 259
 Tau-Funktion 122
 Teilersummenfunktion 114
 Tetrinen 263
 Tierkreiszeichen 271
 Top Seven der Mathematischen
 Vermutungen 17
 Trial-Factoring 35
 Tribonacci-Folge 53
 Tschebyschew-Funktion 108
 Typographische Manipulation 211
 Übergangswahrscheinlichkeiten 212
 Übertragungsweg 240
 Universum 207
 Unkraut 8
 Unvollständigkeitstheorem 175
 Uracil 234
 Urknall 207
 verborgene Strukturen 171
 Vermutung von Catalan 292
 Verschlüsselungsexponenten 242
 Verschlüsselungsmethoden 9
 Viele-Welten-Hypothese 207
 vierdimensionale Kugeln 152
 vierdimensionale Objekte 153
 Vollkommene Zahl 16
 Vollkommene Zahlen 44
 Wagstaff-Primzahlen 18
 Wasserstoff 268
 Weierstraßsche P-Funktion 90
 Wellenfunktion 101
 Weltall 269
 Wieferich-Primzahl 67
 Wilson-Primzahl 70
 Wilson-Quotient 70
 Wirtschaftsleben 274
 Wittgenstein, Ludwig 266
 wohlgeformt 182, 187
 Wohlklang 250
 Wolstenholme-Primzahl 71
 Woronin 86
 Zahlenkörpersieb 18, 309
 Zahlenmystik 45
 Zahlensystem 176
 Zen-Buddhismus 266
 Zetafunktion 17
 Nullstellen der 17
 Zeta-Funktion 79, 329
 Argument 84
 Dualität zu Primzahlen 98
 Hurwitzsche 81
 Lerchsche 81
 Nullstellen 79
 Phase 84
 Produktformel 86
 unerwartete Produktdarstellung 93
 Z-Funktion 93
 Zuckerkandl 265
 Zufalls-Matrizen 99
 Zustandsfolge 212
 Zyklische Zahlen 262

COMPUTERPROGRAMME – CD

Die gedruckte Version dieses Buches enthält in der Beilage eine CD.

Diese CD enthält Animationen (MOV- oder Flash-Format), Mathematica-Notebooks, Grafiken (PDF- oder PGN-Format) und Sound-Dateien (MP3-, MIDI- oder Melodyne-Format), sowie die freie, gekürzte Online-Version dieses Buches im pdf-Format (in Deutscher und Englischer Sprache).

ANIMATIONEN

Verzeichnis: Lattice-Points_On_N-spheres_In_N_Dimensions:

```
latticePointsOn1-spheresIn2Dimensions_RQ2-100.mov
latticePointsOn2-spheresIn3Dimensions_RQ11Fakultät_rotateWithViewVector_1Minute.mov
latticePointsOn2-spheresIn3Dimensions_RQ1001_rotateWithViewVector.mov
latticePointsOn2-spheresIn3Dimensions_RQ1001_rotateWithViewVector_1Minute.mov
```

Verzeichnis: Zeta_Function:

```
ZetaProductOverPrimeTerms_n2_200_1_x_2_100.mov
ZetaProductOverPrimeTerms_n10_100_1_x_0_71.mov
ZetaProductOverPrimeTerms_n10_100_1_x_0_71.swf
```

MATHEMATICA-NOTEBOOKS

Folgende Verzeichnisse enthalten die über 120 Mathematica-Notebooks, mit denen die zahlreichen Tabellen, grafischen Darstellungen und Animationen generiert wurden:

```
1_F_Noise_PrimePi-Signals
ABC-Conjecture
Aliquot-Sequences
DNA-Sequences
Factorization
Fibonacci
Functions_Generating_Primes
Functions_Having_Zeros_Or_Minima_At_Primes
Gradus_Suavitatis_Music_General
Last_Digits_In_Prime_Sequence
Lattice-Points_In_4_Dimensions
Matrix
MersennePrimes
Moebius_Mertens
OCRONS
Plots_Of_Zetafunction_Using_Product-Representation
Prime_NTuples_MaxPrimeDensity
Prime-Polynom_With_26_Variables
Primes_And_Star_Constellations
Ramanujan-Sums
Ramanujan-Tau
RG_Numbers
Riemann_Exact_Explicit_Formula
RSA
Sigma_Function_Tests
Special_Types_Of_Primes_And_Other_Numbers
Tests_With_Recursive_Sequences_(Perrin_Reed_Jameson)
```

Twin_Triple_Sexy_Primes
Using_Zeta_Zeros_To_Compute_Numbertheoretic_Functions
Wieferich_And_Similar_Primes
Zeta-Function

SOUNDS

Verzeichnis: Sounds

primeNumberSong46Sec.mid
primeNumberSong46Sek.mp3
Eratosthenes.mpd (Melodyne-Datei)

GRAFIKEN

Das Verzeichnis Images enthält zahlreiche Grafiken in Vektor- und Raster-Formaten. Hinweis: Diese Grafiken dürfen ohne Erlaubnis des Autors nicht weiterverbreitet, vervielfältigt oder im Internet dargestellt werden.

DANKSAGUNGEN

Ich danke meiner Frau Michèle, die neben Berufstätigkeit und Klavierspielen noch Zeit fand, dieses Buch Korrektur zu lesen.

Vielen Dank an meinen Freund und Verleger Eckhard Bodner für die hervorragende und problemlose Zusammenarbeit.

Mein Freund und Gründer der Firma ‚Celemony‘ Peter Neubäcker hat mir freundlicherweise das Melodyne Arrangement für den Primzahl-Song erstellt.

Mein alter Schulfreund Reinhard Schiller hat mich auf einen Fehler im Kapitel ‚Wieferich-Primzahlen‘ hingewiesen.

Vielen Dank an Peter Danzeglocke, der wichtige Erkenntnisse zur Reed Jameson Folge beisteuerte.

Vielen Dank an meinen Übersetzer, Lektor und Korrektor Ewan Whyte, der auch für die deutsche Version dieses Buches wertvolle Hinweise gab.

Christof Born korrigierte ein paar Bildbeschriftungen.

Dank gebührt auch den Betreibern folgender Internetseiten, von denen viele Formeln und Ergebnisse übernommen wurden:

<http://www.mathworld.wolfram.com> (Eric Weisstein)

<https://www.primes.utm.edu> (Chris Caldwell)

<https://www.oeis.org> (Neil J. A. Sloane)

<http://www.trnicely.net> (Dr. Thomas R. Nicely)

<http://www.mersenne.org> (George Woltman und Scott Kurowski)

<http://www.primzahlen.de> (Hans-Michael Elvenich)

<http://www.seti.org> (Bill Diamond)

<http://www.aliquot.de> (Dr.rer.nat Wolfgang Creyaufmüller)

<http://www.mathpages.com> (Kevin Brown)

<http://www.wikipedia.org>

und natürlich: <http://www.wolfram.com/mathematica>

Schließlich gilt mein Dank den unzähligen Programmierern der Mathematica-Software, ohne die dieses Buch nicht möglich gewesen wäre.