Stefan Planitzer, BSc

On integers of the form $p + g^n$

MASTER THESIS

written to obtain the academic degree of a

Master of Science (MSc)

Master's Programme Mathematical Computer Science

submitted at

Graz University of Technology

supervisor:
Assoc.Prof. Dipl.-Math. Dr.rer.nat.habil. Christian Elsholtz

Institute of Analysis and Computational Number Theory (Math A)

Graz, March 2015

STATUTORY DECLARATION

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present masters thesis.

Date

Signature

Abstract

English version

The problem of determining the density (if it exists) of integers of the form $p + 2^n$ within the positive integers has quite a long history and many famous mathematicians including Euler, de Polignac and Erdős worked on it. Romanov proved that the proportion of integers representable as the sum of a prime and a power of two is positive. In his honour the density of those integers is called Romanov's constant. Today there is quite a gap between explicit upper and lower bounds on the upper and lower density of those integers respectively. Modern approaches to getting lower bounds on the lower density of integers of the form $p + 2^n$ use methods from sieve theory and computational results.

In this thesis we present a generalization of a new method of Elsholtz and Schlage-Puchta who hold the current record on lower bounds for the lower density of sums of primes and powers of 2. As Elsholtz and Schlage-Puchta we get bounds for the lower density of integers of the form $p + g^n$ for arbitrary $g \in \mathbb{N}$ by solving the problem in all residue classes modulo $g^m - 1$ for fixed m separately. Amongst the tools we need to do so are a variant of the Selberg sieve and computational methods.

keywords: Romanov's constant, de Polignac's conjecture, Selberg sieve, Cauchy-Schwarz inequality, Prime Number Theorem for arithmetic progressions, Möbius' μ function

Deutsche Version

Das Problem die Dichte von Zahlen der Form $p+2^n$ innerhalb der natürlichen Zahlen zu bestimmen (falls sie denn existiert) hat mittlerweile eine recht lange Geschichte. Zu den berühmtesten Mathematikern die daran gearbeitet haben zählen unter anderen Euler, de Polignac und Erdős. Romanov konnte zwar zeigen, dass sich ein positiver Anteil der natürlichen Zahlen als Summe einer Primzahl und einer Potenz von 2 darstellen lässt (ihm zu Ehren wird diese Dichte auch als Romanovs Konstante bezeichnet), die bekannten oberen und unteren Schranken für die obere bzw. untere Dichte dieser Zahlen liegen allerdings noch weit auseinander. Moderne Ansätze zur Lösung dieses Problems beinhalten unter anderem den Einsatz von Siebmethoden und Computern.

In dieser Arbeit wollen wir eine Verallgemeinerung einer neuen Methode von Elsholtz und Schlage-Puchta präsentieren, die den aktuellen Rekord bezüglich unterer Schranken für die untere Dichte von Zahlen der Form $p + 2^n$ halten. Wie Elsholtz und Schlage-Puchta werden wir Schranken für die untere Dichte von Zahlen der Form $p+g^n$ für beliebiges $g \in \mathbb{N}$ erhalten, indem wir das Problem getrennt auf Restklassen modulo $g^m - 1$ für festes m betrachten. Unter den Werkzeugen die wir dafür brauchen werden sind eine Variante des Selberg Siebs sowie der Einsatz eines Computeralgebrasystems.

Schlüsselwörter: Romanovs Konstante, de Polignacs Vermutung, Selberg Sieb, Cauchy-Schwarzsche Ungleichung, Primzahlsatz für arithmetische Progressionen, Möbiussche μ Funktion

List of Tables

Contents

1 Notation

Throughout this thesis $\mathbb{N}$ denotes the set of integers, p will always denote a prime number and $\mathbb{P}$ denotes the set of primes $p \in \mathbb{N}$. For any $n \in \mathbb{N}$ we define $P^+(n)$ to be the largest prime factor of n. As usual (a, b) stands for the greatest common divisor of a and b, $[a, b]$ denotes their least common multiple and by $a|b$ we mean that a divides b i.e.

$$a|b \Leftrightarrow \exists d \in \mathbb{N} : b = d \cdot a.$$

For $n \in \mathbb{N}$ $p^n||a$ denotes the highest power of p dividing a. Euler's totient function and Möbius' μ function are denoted by $\varphi(n)$ and $\mu(n)$ and with log we always mean the logarithm with base e. With $\mathrm{ord}_n(a)$ for $a, n \in \mathbb{N}$ we mean the order of a mod n defined by

$$\mathrm{ord}_n(a) := \begin{cases} \min\{k \in \mathbb{N} : a^k \equiv 1 \bmod n\}, & \text{if the minimum exists} \\ \infty, & \text{otherwise.} \end{cases}$$

The von Mangoldt function $\Lambda(n)$ is defined by

$$\Lambda(n) := \begin{cases} \log p, & \text{if } n = p^k, k \in \mathbb{N} \\ 0, & \text{otherwise} \end{cases}$$

and its summatory function is denoted by

$$\Psi(x) := \sum_{1 \leq n \leq x} \Lambda(n).$$

If we only want to sum over integers n satisfying $n \equiv a \bmod q$ we write

$$\Psi(x; q, a) := \sum_{\substack{1 \leq n \leq x \\ n \equiv a \bmod q}} \Lambda(n).$$

The logarithmic integral $\mathrm{li}(x)$ is defined by

$$\mathrm{li}(x) := \int_2^x \frac{\mathrm{d}t}{\log t}.$$

The number of primes below some number x will be denoted by

$$\pi(x) := \{p \in \mathbb{P} : p \leq x\}$$

and for positive integers q, a with $(q, a) = 1$ we define

$$\pi(x; q, a) := \{p \in \mathbb{P} : p \equiv a \bmod q\}.$$

The number of different representations of $n \in \mathbb{N}$ as $n = p + g^k$ for $g \in \mathbb{N}$ is denoted by $r_g(n)$, i.e.

$$r_g(n) = |\{(p, k) : n = p + g^k, p \in \mathbb{P}, k \in \mathbb{N}\}|.$$

For $g \in \mathbb{N}$, $g \geq 2$, we set

$$\underline{d_g} := \liminf_{x \to \infty} \frac{|\{n \leq x : r_g(n) > 0\}|}{x}$$

and

$$\overline{d_g} := \limsup_{x \to \infty} \frac{|\{n \leq x : r_g(n) > 0\}|}{x}$$

for the upper and lower density of positive integers representable as the sum of a prime and a power of g. For coprime $g, n \in \mathbb{N}$ we define

$$\epsilon_g(n) := \min\{k \in \mathbb{N} : g^k \equiv 1 \bmod n\}.$$

The multiplicative function $f(n)$ is defined by

$$f(n) := \begin{cases} \prod_{p|n} \frac{1}{p-2}, & \text{if } \mu^2(n) = 1 \text{ and } n \text{ is odd} \\ 0, & \text{otherwise} \end{cases}$$

and using this we furthermore define

$$S_g(t, m) := \sum_{\substack{(d, 2(g^m - 1)) = 1 \\ (\epsilon_g(d), m) | t}} \frac{f(d)(\epsilon_g(d), m)}{\epsilon_g(d)}.$$

2 Numbertheoretic basics

In this chapter we introduce basic results in number theory. We start by giving results about Euler's φ function and Möbius μ function and end by stating important theorems in number theory (like the Prime Number Theorem for arithmetic progressions) we want to use later.

2.1 Euler's φ function and Möbius' μ function

We start this chapter by defining the greatest common divisor and the least common multiple of two integers.

Definition 2.1.1. *Let $a, b \in \mathbb{N}$, $P = \{p \in \mathbb{P} : p|ab\}$ and $a = \prod_{p \in P} p^{k_p}$, $b = \prod_{p \in P} p^{l_p}$ where $k_p, l_p \in \mathbb{N}_0$. Then we define the greatest common divisor of a and b as*

$$(a, b) := \gcd(a, b) := \prod_{p \in P} p^{\min(k_p, l_p)}$$

and the least common multiple of a and b as

$$[a, b] := \operatorname{lcm}(a, b) := \prod_{p \in P} p^{\max(k_p, l_p)}.$$

We call the integers a, b relatively prime or coprime if $(a, b) = 1$.

We will give a quick application of the least common multiple in a Lemma concerning the order of an integer modulo two different moduli we want to use later.

Lemma 2.1.1. *Let $a, m, n \in \mathbb{N}$ with $(a, n) = (a, m) = (m, n) = 1$ and* $\operatorname{ord}_m(a) = k$, $\operatorname{ord}_n(a) = l$. *Then*

$$\operatorname{ord}_{mn}(a) = [k, l].$$

Proof. From the definition of the least common multiple it follows easily that $[k, l]$ is the least integer divisible by k and l. In particular

$$k|[k, l], \text{ and } l|[k, l]$$

hence $[k, l] = u \cdot k$ and $[k, l] = c \cdot l$. Since k and l are the orders of a modulo m and n respectively we get $m|a^{[k,l]} - 1$ and $n|a^{[k,l]} - 1$. With $(m, n) = 1$ this implies

$$mn|a^{[k,l]} - 1$$

and thus

$$\operatorname{ord}_{mn}(a)|[k, l].$$

Now let $\operatorname{ord}_{mn}(a) = q$. Then $mn|a^q - 1$ hence $m|a^q - 1$ and $n|a^q - 1$. This implies that $k|q$ and $l|q$ and since $[k, l]$ is the least integer with this property we get $[k, l]|q = \operatorname{ord}_{mn}(a)$. □

Before defining Euler's φ and Möbius' μ function we state the Chinese remainder theorem. It will be useful in proving properties of the φ function and we will need it later when we deal with sieve theory. The version of this theorem below is taken from [Dic57, Theorem 15] and given without proof.

Theorem 2.1.1 (Chinese Remainder Theorem)**.** *If $m_1, \ldots, m_t$ are relatively prime in pairs, there exist integers x for which simultaneously*

$$x \equiv a_1 \bmod m_1, \ldots, x \equiv a_t \bmod m_t.$$

All such integers x are congruent modulo $m = m_1 m_2 \cdots m_t$.

The following is an easy consequence of the Chinese Remainder Theorem.

Corollary 2.1.1. *Let $(a + kb)_{k\in\mathbb{N}}$ and $(c + ld)_{l\in\mathbb{N}}$, $a, b, c, d \in \mathbb{N}$ be two arithmetic progressions where $(b, d) = 1$. Then these progressions intersect in a unique arithmetic progression modulo bd.*

Proof. Since $(b, d) = 1$ by Theorem 2.1.1 there exists x with

$$x \equiv a \bmod b \text{ and } x \equiv c \bmod d$$

and this x is uniquely determined $\bmod bd$. Hence $(x+rbd)_{r\in\mathbb{N}}$ is the unique intersection of the two given progressions modulo bd. □

Definition 2.1.2 (Euler's φ function)**.** *Let* $\varphi : \mathbb{N} \to \mathbb{N}$ *with*

$$\varphi(n) = |\{a \leq n : (a, n) = 1\}|.$$

Definition 2.1.3 (Möbius' μ function)**.** *Let* $\mu : \mathbb{N} \to \{0, \pm 1\}$ *with*

$$\mu(n) = \begin{cases} (-1)^{|\{p \in \mathbb{P} : p | n\}|}, & \textit{if } p \textit{ is squarefree} \\ 0, & \textit{otherwise.} \end{cases}$$

An important class of frequently studied functions in number theory is the class of multiplicative functions.

Definition 2.1.4. *Let* $f : \mathbb{N} \to \mathbb{C}$ *be a function. We call* f *a multiplicative function if* $f(1) = 1$ *and for any* $a, b \in \mathbb{N}$ *with* $(a, b) = 1$ *we have* $f(ab) = f(a)f(b)$*.*

Lemma 2.1.2. *Euler's* φ *function and Möbius'* μ *function are multiplicative.*

Proof. We start with Möbius' μ function. We have $\mu(1) = (-1)^{|\{p \in \mathbb{P} : p | 1\}|} = (-1)^0 = 1$. If either a or b is not square free - say a is not square free - then ab is not square free and we have $0 = \mu(ab) = \mu(a)\mu(b)$. So we can assume that a, b are square free and coprime. In this case we have

$$\begin{aligned} \mu(ab) = (-1)^{|\{p \in \mathbb{P} : p | ab\}|} &= (-1)^{|\{p \in \mathbb{P} : p | a\}| + |\{p \in \mathbb{P} : p | b\}|} = \\ &= (-1)^{|\{p \in \mathbb{P} : p | a\}|}(-1)^{|\{p \in \mathbb{P} : p | b\}|} = \mu(a)\mu(b). \end{aligned}$$

For Euler's φ function we also have $\varphi(1) = 1$. Furthermore for any $a_1 \leq q$, $b_1 \leq b$ with $(a_1, a) = 1$ and $(b_1, b) = 1$ by Theorem 2.1.1 there is exactly one $x \in \mathbb{N}$, $x \leq ab$ with

$$x \equiv a_1 \bmod a \text{ and } x \equiv b_1 \bmod b. \tag{2.1.1}$$

Now suppose by contradiction that $(x, ab) = g > 1$. Then there is some $p \in \mathbb{P}$ with $p | g$ hence $p | x$, $p | ab$ and thus $p | a$ or $p | b$ - w.l.o.g. $p | a$. Because of $x \equiv a_1 \bmod a$ we get that $a_1 = x + la$ for some $l \in \mathbb{N}$ and since $p | (x + la)$ we also have $p | a_1$. Therefore $p | (a_1, a)$ which is a contradiction to $(a_1, a) = 1$.

On the other hand, if we have some $c_1 \in \mathbb{N}$ with $(c_1, ab) = 1$ then also $(c_1, a) = 1$ and $(c_1, b) = 1$ and there are $a_1 \leq a$ and $b_1 \leq b$ with $a_1 \equiv c_1 \bmod a$ and $b_1 \equiv c_1 \bmod b$ satisfying $(a_1, a) = 1$ and $(b_1, b) = 1$.

These properties ensure that the map

$$\Psi : \{a_1 \leq a : (a_1, a) = 1\} \times \{b_1 \leq b : (b_1, b) = 1\} \to \{c_1 \leq ab : (c_1, ab) = 1\}$$

with $\Psi((a_1, b_1)) = x$ where x is the unique solution of equation (2.1.1) satisfying $x \leq ab$ is bijective in the case of $(a, b) = 1$, hence φ is multiplicative. $\square$

Another useful property of the μ function is given in the following Lemma.

Lemma 2.1.3.

$$\sum_{d|n} \mu(d) = \begin{cases} 1, & \textit{if } n = 1 \\ 0, & \textit{if } n > 1. \end{cases}$$

Proof. We prove the Lemma by induction on the number k of different prime divisors of n. For $k = 0$ we have $n = 1$ hence

$$\sum_{d|1} \mu(d) = \mu(1) = 1.$$

Let n be a number with exactly $k > 0$ prime divisors and let those prime divisors be $\{p_1, \ldots, p_k\}$. Since for any non-squarefree d we have $\mu(d) = 0$ we may write

$$\sum_{d|n} \mu(d) = \sum_{I \subset \{p_1,\ldots,p_k\}} (-1)^{|I|}.$$

We can separate the squarefree divisors of n into two sets: divisors not being divisible by p_k and those that are divisible by p_k. Doing so we write

$$\begin{aligned} \sum_{d|n} \mu(d) &= \sum_{I \subset \{p_1,\ldots,p_{k-1}\}} (-1)^{|I|} + \sum_{I \subset \{p_1,\ldots,p_{k-1}\}} (-1)^{|I \cup \{p_k\}|} = \\ &= \sum_{I \subset \{p_1,\ldots,p_{k-1}\}} (-1)^{|I|} + \sum_{I \subset \{p_1,\ldots,p_{k-1}\}} (-1)^{|I|+1} = \\ &= \sum_{I \subset \{p_1,\ldots,p_{k-1}\}} (-1)^{|I|} - \sum_{I \subset \{p_1,\ldots,p_{k-1}\}} (-1)^{|I|} = 0. \end{aligned}$$

□

The following theorem is taken from [HW08, Theorem 266] and called the Möbius inversion formula.

Theorem 2.1.2. *Lef $f : \mathbb{N} \to \mathbb{C}$ be a multiplicative function. If*

$$g(n) = \sum_{d|n} f(n),$$

then

$$f(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) g(d) = \sum_{d|n} \mu(d) g\left(\frac{n}{d}\right).$$

2.2 The distribution of primes

The following two classical results from analytic number theory about the distribution of primes together with their proofs can be found in [Brü95]. For the Prime Number Theorem cf. [Brü95, Satz 2.8.2] and for the Prime Number Theorem for arithmetic progressions cf. [Brü95, Satz 3.3.3]

Theorem 2.2.1 (Prime Number Theorem)**.** *There exits a constant $c > 0$ such that*

$$\Psi(x) = x + \mathcal{O}(xe^{-c\sqrt{\log x}})$$

and hence

$$\pi(x) = \mathrm{li}(x) + \mathcal{O}(xe^{-c\sqrt{\log x}}).$$

Theorem 2.2.2 (Prime Number Theorem for arithmetic progressions, Siegel-Walfisz)**.** *Fix $A > 0$. The there exists a constant $C = C(A)$ such that for $q \leq (\log x)^A$ and all a with $(a, q) = 1$*

$$\Psi(x; q, a) = \frac{x}{\varphi(q)} + \mathcal{O}(xe^{-C\sqrt{\log x}})$$

and hence

$$\pi(x; q, a) = \frac{1}{\varphi(q)}\,\mathrm{li}(x) + \mathcal{O}(xe^{-c\sqrt{\log x}}).$$

From the above two theorems we get in particular that $\pi(x) \sim \frac{x}{\log x}$ and $\pi(x; q, a) \sim \frac{1}{\varphi(a)} \frac{x}{\log x}$.

3 Romanov's Constant - a historical overview

3.1 De Polignac's conjecture and Romanov's constant

In 1849 Alphonse de Polignac [dP49, p. 400] stated

> "Tout nombre impair est égal à une puissance de 2, plus un nombre premier. (Vérifié jusqu'à 3 millions.)"

claiming that any odd number lower than 3000000 is the sum of a prime and a power of 2. The conjecture that this holds for any odd number since then bears his name. Also in [dP49] a rectification of de Polignac concerning this statement is published. He says that when writing up his article for submission he was very short of time and couldn't do all the calculations himself and it seems that the verification of the above conjecture was not done with greatest care. De Polignac mentions a letter of Leonhard Euler to Christian Goldbach in which a counterexample can be found and indeed in 1752 Euler [Eul, p. 596 f.] wrote the following to Goldbach:

> "Es fiel mir dabey ein ziemlich ähnliches theorema ein, nehmlich: dass a numero impari non primo $2n-1$ allzeit eine potestas binarii abgezogen werden könne, dass der Rest ein numerus primus sey. Nach angestellter Probe hat sich dieses auch noch bis auf sehr grosse Zahlen wahr befunden; als ich aber auf $959 = 7 \cdot 137$ kam, so fand sich eine Ausnahme, indem $959 - 2^a$ nullo modo primus werden kann."

So de Polignac's conjecture does fail for the composite number 959, hence not even a restricted version of the conjecture, claiming that de Polignac's conjecture is true for any odd integer that is not a prime number (127 is an exeption of de Polignac's conjecture as well, but it is a prime number) is true.

Knowing that we cannot represent any odd integer as the sum of a prime and a power of 2 we shift our attention to the following question: Is the lower density of numbers representable as a sum of a prime and a power of 2 within the integers positive, i.e. is there some $\alpha > 0$ with

$$\liminf_{x\to\infty} \frac{\{n \leq x : r_2(n) > 0\}}{x} = \alpha?$$

Romanov [Rom34] proved that such an α exists not only for sums of primes and powers of two but for sums of primes and powers of an arbitrary positive integer $g \geq 2$. In the following we give a short overview over the main ideas of his proof.

We consider two sequences the first of which is the sequence of primes

$$p_1, p_2, p_3, \ldots$$

i.e. p_i denotes the i-th prime number and the second one is the sequence of consecutive powers of g i.e.

$$1, g, g^2, \ldots.$$

There are $\pi(x)$ primes less than or equal to x and we know that there are $N(x) := \lfloor \log_g x \rfloor$ powers of g not exceeding x. We now denote by $A_1^{(x)}(n)$, $A_2^{(x)}(n)$ and $r_g^{(x)}(n)$ the number of solutions of the equations

$$p_i - p_j = n,\ g^i - g^j = n \text{ and } p_i + g^j = n$$

respectively where $p_i \leq x$,$p_j \leq x$, $g^i \leq x$ and $g^j \leq x$. We define $\nu(2x)$ to be the number of integers $n \leq 2x$ which are of the form $p + g^i$ where $p \leq x$ and $g^i \leq x$.

We now set $\eta_n = 1$ if $r_g^{(x)}(n) > 0$ and $\eta_n = 0$ else. An application of the Cauchy-Schwarz inequality yields

$$\sum_{n=0}^{2x} \eta_n^2 \sum_{n=0}^{2x} r_g^{(x)}(n)^2 \geq \left(\sum_{n=0}^{2x} \eta_n r_g^{(x)}(n) \right)^2 = \left(\sum_{n=0}^{2x} r_g^{(x)}(n) \right)^2$$

hence we arrive at

$$\nu(2x) = \sum_{n=1}^{2x} \eta_n^2 \geq \frac{\left(\sum_{n=0}^{2x} r_g^{(x)}(n)\right)^2}{\sum_{n=0}^{2x} r_g^{(x)}(n)^2}.$$

By counting the solutions of the equations

$$g^i - g^j - p_k + p_l = 0$$

where $g^i, g^j, p_k, p_l \leq x$ in two different ways Romanov shows that

$$\sum_{n=0}^{x} r_g^{(x)}(n)^2 = \pi(x)N(x) + 2\sum_{n=1}^{x} A_1^{(x)}(n)A_2^{(x)}(n).$$

The first way of counting these solutions is by having a look at the number of solutions of

$$g^i - g^j = n \text{ and } p_k - p_l = n$$

for $n \in \{-x, -(x-1), \ldots x-1, x\}$. In this case we arrive at $A_1^{(x)}(0)A_2^{(x)}(0) + 2\sum_{n=1}^{x} A_1^{(x)}(n)A_2^{(x)}(n)$. A second way of counting solutions is by looking at the following system of equations

$$p_l + g^i = n \text{ and } p_k + g^j = n$$

for $n \in \{0, 1, \ldots, 2x\}$ which leads directly to $\sum_{n=0}^{2x} r_g^{(x)}(n)^2$. Because of $\sum_{n=0}^{2x} r_g^{(x)}(n) = \pi(x)N(x)$ it remains to find an upper bound for

$$\sum_{n=1}^{x} A_1^{(x)}(n)A_2^{(x)}(n).$$

Romanov observes that $A_2^{(x)}(n)$ is either 0 or 1 i.e. if an integer can be written as the difference of two powers of g then this representation is unique. $A_1^{(x)}(n)$ counts pairs of primes lower than or equal to x with fixed difference n and Romanov uses an upper bound sieve to get the following estimate

$$A_1^{(x)}(n) < c_1 \frac{x}{\log^2 x} \prod_{\substack{p|n \\ p \in \mathbb{P}}} \left(1 + \frac{1}{p}\right).$$

Putting this together and evaluating the sum Romanov shows that

$$\sum_{n=1}^{x} A_1^{(x)}(n)A_2^{(x)}(n) < c_2 x \sum_{\substack{k=1 \\ (k,a)=1}}^{\infty} \frac{\mu(k)^2}{k\epsilon_g(k)}$$

where he proves the sum on the right side to be convergent and therefore

$$\sum_{n=1}^{x} A_1^{(x)}(n)A_2^{(x)}(n) < c_3 x.$$

Now we can put everything together and have

$$\begin{aligned}\nu(2x) &> \frac{(\pi(x)N(x))^2}{\pi(x)N(x) + 2\sum_{n=1}^{x} A_1^{(x)}(n)A_2^{(x)}(n)} > \\ &> \frac{c_4^2 \frac{x^2}{\log^2 x}(\lfloor \frac{\log x}{\log g} \rfloor + 1)^2}{c_4 \frac{x}{\log x}(\lfloor \frac{\log x}{\log g} \rfloor + 1) + 2c_3 x} \geq \frac{\frac{c_4^2 x^2}{\log^2 g}}{\frac{c_4 x}{\log g} + \frac{c_4 x}{\log x} + 2c_3 x} \\ &= x \frac{\frac{c_4^2}{\log^2 g}}{\frac{c_4}{\log g} + \frac{c_4}{\log x} + 2c_3} > \beta x.\end{aligned}$$

3.2 Explicit values for Romanov's constant

Knowing that a positive proportion of the odd positive integers is of the form $p+2^n$ we ask for explicit lower bounds. While Romani [Rom83] gives heuristic arguments suggesting that the density of positive integers of the form $p+2^n$ is about 0.434..., Chen and Sun [CS04] proved that $\underline{d_2} > 0.0868$, Habsieger and Roblot [HR06] were able to improve on $\underline{d_2} > 0.0933$. Pintz [Pin06] proved that $\underline{d_2} \geq 0.09368$ but according to [HSF10] his calculation is based on a value of a constant in an upper bound sieve for which Dong Wu gave an incomplete proof and with a corrected version would arrive at $\underline{d_2} > 0.093626$. Habsieger and Sivak-Fischler [HSF10] themselves improved on $\underline{d_2} > 0.0936275$. The best known result is due to Elsholtz and Schlage Puchta [ESP12] which is $\underline{d_2} > 0.107648$. In the following we want to have a look at the ideas of Habsieger and Roblot [HR06] and Elsholtz and Schlage-Puchta [ESP12].

3.2.1 The lower bound by Habsieger and Roblot

Habsieger and Roblot [HR06] use a result of Pintz and Ruzsa to give a lower bound of 0.0933 for Romanov's constant. Pintz and Ruzsa [PR03] proved that

$$s(x) = \sum_{n=1}^{x} r_2^2(n) < 5.3636 \cdot x \frac{2}{\log^2 2}.$$

Instead of directly using the Cauchy-Schwarz inequality which would imply

$$(\pi(x)L)^2 \leq d(x)s(x)$$

where $L = \left\lfloor \frac{\log x}{\log 2} \right\rfloor$ and $d(x) = \sum_{\substack{n \leq x \\ r_2(n) > 0}} 1$, Habsieger and Roblot are able to improve the bound by the following approach.

First they define

$$\epsilon_x = \frac{\sum_{1 \le n \le x} r_2(n)}{\sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} 1}$$

to be the average number of representations of an integer as the sum of a prime and a power of two. Furthermore they set

$$\epsilon = \liminf \epsilon_x = \frac{1}{\underline{d_2} \log 2}$$

so that some subsequence of $(\epsilon_n)_{n \in \mathbb{N}}$ converges to ϵ. Instead of considering the sum $s(x)$, Habsieger and Roblot have a look at the sum of squared deviation of the numbers $r_2(n)$ from their average value ϵ_x:

$$\Delta_x = \sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} (r_2(n) - \epsilon_x)^2 = \sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} r_2(n)^2 - 2\epsilon_x \sum_{1 \le n \le x} r_2(n) + \epsilon_x^2 \sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} 1 =$$

$$= \sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} r_2(n)^2 - \frac{\left(\sum_{1 \le n \le x} r_2(n)\right)^2}{\sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} 1} = \sum_{1 \le n \le x} r_2(n)^2 - \frac{\left(\pi(n) \frac{\log x}{\log 2}\right)^2}{\sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} 1}.$$

Using the Prime Number Theorem 2.2.1 and the result of Pintz and Rusza we arrive at

$$\Delta_x \le \frac{2x}{\log^2 2} \left(5.3636 - \frac{1}{2\underline{d}} + \mathrm{o}(1)\right).$$

Next we need a lower bound for Δ_x. Habsieger and Roblot get it by just observing that if $\epsilon < 15$ would hold, they would, by definition of ϵ, arrive at a better result than $\underline{d} = 0.0933$ and for $\epsilon > 15.5$ they would get a wrong bound for $\underline{d}$. So we can choose $\epsilon \in (15, 15.5)$ and get

$$\Delta_x \ge \sum_{\substack{1 \le n \le x \\ r_2(n) > 0}} (15 - \epsilon_x)^2 \ge x \left(\underline{d} \left(15 - \frac{1}{\underline{d} \log 2}\right)^2 + \mathrm{o}(1)\right).$$

Putting the two inequalities together we arrive at

$$\underline{d} \left(15 - \frac{1}{\underline{d} \log 2}\right)^2 \le \frac{2}{\log^2 2} \left(5.3636 - \frac{1}{2\underline{d}}\right)$$

from where we get to

$$\underline{d}^2 225 \log^2 2 - \underline{d}(30 \log 2 + 10.7272) + 2 \le 0$$

and finally to $\underline{d} \ge 0.0933$.

3.2.2 A new idea by Elsholtz and Schlage-Puchta

Elsholtz and Schlage-Puchta hold the current record regarding Romanov's constant. Since we want to use a similar method as they used in [ESP12] the basic idea of their approach is pointed out in this section. As many of the other proofs the proof of Elsholtz and Schlage-Puchta of a lower bound for Romanov's constant relies on the use of the Cauchy-Schwarz inequality and they work with the observation that the inequality in the form

$$\langle v, w\rangle \leq ||v|| \cdot ||w||$$

is an equality if and only if v and w are linearly dependent. Linear dependency in our case would mean that all integers have the same number of representations as a sum of a prime and a power of two. Partitioning the integers in residue classes modulo $2^{24} - 1$ Elsholtz and Schlage-Puchta get some improvement by using the inequality on the residue classes separately making use of the fact that within those classes the distribution of the number of representations is more homogenous than by considering the positive integers as a whole.

As an example they give the situation modulo 3. By Theorem 2.2.2 the primes are equally distributed among the residue classes 1 and 2 mod 3 and the same holds for powers of 2. As a consequence a positive integer n with $n \equiv 0 \bmod 3$ is the sum of a prime and a power of 2 with probability $\frac{1}{2}$ whereas if $n \equiv 1 \bmod 3$ or $n \equiv 2 \bmod 3$ this probability is $\frac{1}{4}$. This is true since $1 + 2 \equiv 0 \bmod 3$ and $2 + 1 \equiv 0 \bmod 3$ but only $2 + 2 \equiv 1 \bmod 3$ and $1 + 1 \equiv 2 \bmod 3$.

This idea together with the use of computers to get explicit numerical values for constants needed for their lower bound yields the improvement.

3.3 A note on upper bounds

In 1950 Erdős [Erd50] and van der Corput [vdC50] independently proved that a positive proportion of the odd integers is not of the form $p+2^n$. We are going to have a look at Erdős' proof who invented covering congruences and used them to explicitly construct a subsequence of the odd positive integers that cannot be represented as a sum of a prime and a power of two. For his argument Erdős uses the following Lemma (the proof is taken from [Nat96, Lemma 7.11]).

Lemma 3.3.1. *Every integer satisfies at least one of the congruences*

$$\begin{aligned} &0 \bmod 2\\ &0 \bmod 3\\ &1 \bmod 4\\ &3 \bmod 8\\ &7 \bmod 12\\ &23 \bmod 24. \end{aligned}$$

Proof. It is easy to check that the integers $0, \ldots, 23$ satisfy at least one of the congruences. Since any integer k is congruent to one of the numbers r between 0 and 23 modulo 24 and r satisfies one of the congruences we arrive at

$$k \equiv r \bmod 24$$

$$r \equiv a_i \bmod m_i$$

for one of the congruences $a_i \bmod m_i$ in the Lemma. Now all the moduli of the system of congruences in the lemma are divisors of 24 hence

$$k \equiv r \equiv a_i \bmod m_i.$$

□

This Lemma can be used to prove the following Theorem (the proof is again taken from [Nat96, Theorem 7.12]).

Theorem 3.3.1. *A positive proportion of the odd integers is not of the form $p + 2^n$.*

Proof. For each of the moduli m_i of the congruences in Lemma 3.3.1 we choose a prime p_i such that the order of 2 mod p_i divides m_i.

$$\begin{aligned} 2^2 &\equiv 1 \bmod 3\\ 2^3 &\equiv 1 \bmod 7\\ 2^4 &\equiv 1 \bmod 5\\ 2^8 &\equiv 1 \bmod 17\\ 2^{12} &\equiv 1 \bmod 13\\ 2^{24} &\equiv 1 \bmod 241. \end{aligned}$$

With $l = 241$ and $m = 2^l \cdot 3 \cdot 7 \cdot 5 \cdot 17 \cdot 13 \cdot 241$ by Theorem 2.1.1 there is a unique congruence class r mod m with

$$\begin{aligned} r &\equiv 1 \bmod 2^l \\ r &\equiv 2^0 \bmod 3 \\ r &\equiv 2^0 \bmod 7 \\ r &\equiv 2^1 \bmod 5 \\ r &\equiv 2^3 \bmod 17 \\ r &\equiv 2^7 \bmod 13 \\ r &\equiv 2^{23} \bmod 241. \end{aligned}$$

Since any integer in the residue class r mod m is of the form $r + a \cdot m$, $a \in \mathbb{Z}$, and m is even and r is odd, all those integers are odd. Now let $N \equiv r$ mod m with $N > 2^l + l$ and choose $n \in \mathbb{N}$ with $2^n < N$. Using the notation of the proof of Lemma 3.3.1 there is some a_i with

$$n \equiv a_i \bmod m_i$$

hence

$$2^n = 2^{a_i + m_i u_i} \equiv 2^{a_i} \bmod p_i$$

for some integer u_i (note that the exponents of 2 in the defining congruences of r correspond to the moduli m_i of Lemma 3.3.1). We arrive at

$$N \equiv 2^n \bmod p_i$$

and therefore $N = 2^n + vp_i$. If $n \leq l$ we get

$$vp_i = N - 2^n \geq N - 2^l > l \geq p_i$$

by the choice of $N > 2^l + l$ and l as $\max\{p_i\}$ hence $v > 1$. If $n > l$ we have $N - 2^n \equiv N \equiv 1 \bmod 2^l$ and for some positive integer w

$$vp_i = N - 2^n = 1 + w2^l > 2^l > l \geq p_i$$

and again $v > 1$. So in any case $N - 2^n$ is composite. Therefore no positive integer in the congruence class r mod m can be represented as the sum of a prime and a power of 2 and the density of positive odd integers with this property is at least $\frac{1}{m} > 0$. □

4

Counting primes in residue classes

4.1 The basic idea of sieve methods

One of the central ingredients of the proof of a new lower bound for Romanov's constant by Elsholtz and Schlage-Puchta was the use of a sieve giving, for fixed $k, d \in \mathbb{N}$, an upper bound for the number of pairs of primes (p_1, p_2) where p_1 is in a fixed residue class modulo k satisfying $p_1 + b = p_2$. Since the proof of a lower bound for integers of the form $p + g^n$ for arbitrary $g \in \mathbb{N}$ will be based on the same sieve, this chapter is devoted to introduce it after giving the basic idea behind sieve methods in general. The rest of this chapter chapter is based on the treatise of Halberstam and Richert [HR11] on sieves.

4.1.1 The sifting function $\mathcal{S}$

Given some finite set $\mathcal{A} \subset \mathbb{N}$ and some set $\mathcal{P} \subset \mathbb{P}$ of primes the goal of a sieve method is to give an upper or lower bound for the number of elements in $\mathcal{A}$ that are not divisible by any prime in $\mathcal{P}$. For this purpose we define $P(z)$ to be

$$P(z) := \prod_{\substack{p<z \\ p\in\mathcal{P}}} p$$

and the sifting function $\mathcal{S}$

$$\mathcal{S}(\mathcal{A}, \mathcal{P}, z) := |\{a \in \mathcal{A} : (a, P(z)) = 1\}|.$$

As an example suppose that $\mathcal{P} = \mathbb{P}$ and we have a list $\mathcal{P}_z = \{p \in \mathbb{P} : p < z\}$ of primes smaller than $z \in \mathbb{N}$. We can extend the list $\mathcal{P}_z$ to a list $\mathcal{P}_{z^2}$ of primes not exceeding z^2 by additionally including those integers of the interval $[z, z^2)$ that are not divisible by any of the primes in $\mathcal{P}_z$. This process is referred to as Eratosthenes' Sieve and can be used to construct lists of primes. Consider for example the list $\mathcal{P}_{10}$ of primes less than 10

$$\mathcal{P}_{10} = \{2, 3, 5, 7\}.$$

By crossing out all multiples of any of those numbers we can construct a list of primes between 10 and 100:

~~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~~

If we choose $\mathcal{A}$ to be

$$\mathcal{A} = \{n \in \mathbb{N} : z \leq n < z^2\}$$

and we have a list of all primes smaller than z at hand, the Prime Number Theorem 2.2.1 states, that the number $\mathcal{S}(\mathcal{A}, \mathcal{P}, z)$ of integers surviving the sifting process satisfies $\mathcal{S}(\mathcal{A}, \mathcal{P}, z) \sim \frac{z^2}{2 \log z}$.

While Eratosthenes' Sieve provides a means of constructing lists of primes what we are looking for is a more theoretical approach. We want to get an upper bound for the set of primes described above by generalizing the sifting process and we are going to get it, by using a method called Selberg's upper bound sieve. Before we introduce it we will need to make some more definitions and get a formal sieve theoretic description of our problem.

4.1.2 The functions ω and g

Given a set $\mathcal{A}$ as above. Later we want to deal with subsets of $\mathcal{A}$ consisting of all $a \in \mathcal{A}$ being divisible by some fixed $d \in \mathbb{N}$ hence we define

$$\mathcal{A}_d := \{a \in \mathcal{A} : a \equiv 0 \bmod d\}.$$

Furthermore suppose that we have an approximation X for $|\mathcal{A}|$ with

$$r_1 := |\mathcal{A}| - X.$$

We then choose the function ω_0 in a way such that $\frac{\omega_0(p)}{p}X$ is an approximation for $|\mathcal{A}_p|$ and define the corresponding remainder term as

$$r_p := |\mathcal{A}_p| - \frac{\omega_0(p)}{p}X.$$

If we require $\omega_0(1) = 1$ and $\omega_0(d) = \prod_{\substack{p|d \\ p\in\mathbb{P}}} \omega_0(p)$ for squarefree d, ω_0 is always a multiplicative function on the positive squarefree integers. We additionally define

$$r_d := |\mathcal{A}_d| - \frac{\omega_0(d)}{d}X.$$

Note that we can always expect ω_0 to be non-negative, since if $\omega_0(p)$ is negative for some $p \in \mathbb{P}$, 0 is certainly a better approximation to $|\mathcal{A}_p|$ than $\frac{\omega_0(p)}{p}X < 0$ and we could in this case set $\omega_0(p) := 0$.

Definition 4.1.1. *Given a set $\mathcal{A}$ to be sifted by a set of primes $\mathcal{P}$ and a function ω_0 satisfying the requirements above. For $p \in \mathbb{P}$ define the function ω as follows:*

$$\omega(p) := \begin{cases} \omega_0(p), & \text{if } p \in \mathcal{P} \\ 0, & \text{if } p \notin \mathcal{P}. \end{cases}$$

The function ω is again extended to the set of positive squarefree integers by setting $\omega(1) = 1$ and

$$\omega(d) = \prod_{\substack{p|d \\ p\in\mathbb{P}}} \omega(p).$$

We furthermore define

$$R_d := |\mathcal{A}_d| - \frac{\omega(d)}{d}X.$$

For the rest of this chapter we want the following condition to be satisfied for ω: for any prime p and some fixed constant $\mathcal{C} \geq 1$

$$1 \leq \frac{1}{1 - \frac{\omega(p)}{p}} \leq \mathcal{C}.$$

If ω satisfies this condition the function

$$g(d) := \frac{\omega(d)}{d\prod_{\substack{p|d \\ p\in\mathbb{P}}} \left(1 - \frac{\omega(p)}{p}\right)}$$

is well defined for any squarefree $d \in \mathbb{N}$.

4.2 The setting in our case

Using the notation of sifting functions introduced in the previous section, we are now ready to apply it to our problem of counting pairs of primes with fixed distance $p_1 + b = p_2$ where p_1 is in a given residue class l modulo some number k. We will do this by considering the set

$$\mathcal{A} : \{p + b : p \in \mathbb{P},\, p \leq x,\, p \equiv l \bmod k\}$$

where $b, k, l \in \mathbb{N}$ and $x \in \mathbb{R}$ such that $(l, k) = 1$, $k < x$ and b is even. For squarefree $d \in \mathbb{N}$ the set $\mathcal{A}_d$ consists of all the elements $p + b$ in $\mathcal{A}$ additionally satisfying $p + b \equiv 0 \bmod d$. To get an estimate for the cardinality of those sets we consider the general congruence $n + b \equiv 0 \bmod d$. Here we will restrict ourselves to choices of b,k and d satisfying $(d, kb) = 1$.

We can choose $n \equiv -b \bmod d$ and since $(d, k) = 1$ by Corollary 2.1.1 the arithmetic progressions $-b \bmod d$ and $l \bmod k$ intersect in a unique progression $l' \bmod dk$. Now by the Prime Number Theorem for arithmetic progressions 2.2.2 we get

$$|\mathcal{A}_d| = |\{p \in \mathbb{P} : p \equiv l' \bmod kd\}| = \pi(x; dk, l') \sim \frac{\operatorname{li} x}{\varphi(dk)} = \frac{1}{d}\frac{d}{\varphi(d)}\frac{\operatorname{li} x}{\varphi(k)}$$

where we note that since $(b, d) = 1$ and $(l, k) = 1$ and $l' \bmod dk$ is the intersection of the progressions $-b \bmod d$ and $l \bmod k$ we have $(l', dk) = 1$. This suggests the definition of X and ω_0 in the following way:

$$X := \frac{\operatorname{li} X}{\varphi(k)}$$

and

$$\omega_0(d) := \frac{d}{\varphi(d)}.$$

Note that the identity function $j : \mathbb{N} \to \mathbb{N}$ as well as Euler's φ function (see Lemma 2.1.2) are multiplicative. A Lemma below will state that with this choices the remainder term

$$r_d = \pi(x; dk, l') - \frac{\operatorname{li} x}{\varphi(dk)}$$

is small enough for our purpose.

4.3 The Selberg upper bound method

In this section the basic idea of the Selberg upper bound sieve shall be explained and applied to the setting described in the previous section.

4.3.1 The basic idea

For a given set $\mathcal{A}$ and a subset $\mathcal{P}$ of the primes we start out with a sum of the form

$$\sum_{a\in\mathcal{A}}\left(\sum_{\substack{d|a\\ d|P(z)}}\lambda_d\right)^2$$

where $(\lambda_d)_{d\in\mathbb{N}}$ is any sequence of real numbers satisfying $\lambda_1 = 1$. If $a \in \mathcal{A}$ is not divisible by any prime in $\mathcal{P}$ the only divisor occurring in the inner sum is $d = 1$ and because of the condition $\lambda_1 = 1$ those elements contribute 1 to the sum. For all the other elements of the set $\mathcal{A}$ we add something non-negative hence the following inequality is true

$$\mathcal{S}(\mathcal{A},\mathcal{P},z) \leq \sum_{a\in\mathcal{A}}\left(\sum_{\substack{d|a\\ d|P(z)}}\lambda_d\right)^2 .$$

To get an upper bound for $\mathcal{S}(\mathcal{A},\mathcal{P},z)$ we have a look at the right hand side of this inequality. By multiplying out and interchanging summation we get

$$\sum_{a\in\mathcal{A}}\left(\sum_{\substack{d|a\\ d|P(z)}}\lambda_d\right)^2 = \sum_{\substack{d_1|P(z)\\ d_2|P(z)}}\lambda_{d_1}\lambda_{d_2}\sum_{\substack{a\in\mathcal{A}\\ [d_1,d_2]|a}}1 = \sum_{\substack{d_1|P(z)\\ d_2|P(z)}}\lambda_{d_1}\lambda_{d_2}|\mathcal{A}_{[d_1,d_2]}|.$$

If we define $D := [d_1, d_2]$ and use the notation we introduced for sieves earlier we get

$$R_D = |\mathcal{A}_D| - \frac{\omega(D)}{D}X$$

thus

$$\sum_{a\in\mathcal{A}}\left(\sum_{\substack{d|a\\ d|P(z)}}\lambda_d\right)^2 \leq X\sum_{\substack{d_1|P(z)\\ d_2|P(z)}}\lambda_{d_1}\lambda_{d_2}\frac{\omega(D)}{D} + \sum_{\substack{d_1|P(z)\\ d_2|P(z)}}|\lambda_{d_1}\lambda_{d_2}R_D| = X\Sigma_1 + \Sigma_2.$$

The quality of the upper bound depends essentially on the choice of the sequence $(\lambda_d)_{d\in\mathbb{N}}$. To choose these values in an optimal way is a hard problem and we impose a further constraint on the λ_d in order to make it easier: we require $\lambda_d = 0$ for $d > z$. The main idea behind this condition is, that by imposing it we can expect Σ_2 to be reasonably small because it consists of just $\mathcal{O}(z^2)$ non zero terms and we can concentrate on finding a minimum for Σ_1.

Since the proof for the optimal choice of the sequence $(\lambda_d)_{d\in\mathbb{N}}$ is not given in [HR11] the proof of the following Lemma is based on the proof of Satz 3.1 in [Pra57].

Lemma 4.3.1. *Let*

$$G(z) = G_1(z) := \sum_{d<z} \mu^2(d) g(d)$$

$$G_k(z) := \sum_{\substack{d<z \\ (d,k)=1}} \mu^2(d) g(d), \text{ for } k > 1$$

where $z \in \mathbb{R}^+$. *Then the choice of the sequence* $(\lambda_d)_{d\in\mathbb{N}}$ *with*

$$\lambda_d := \frac{\mu(d)}{\prod_{p|d}\left(1 - \frac{\omega(p)}{p}\right)} \frac{G_d(\frac{z}{d})}{G(z)}$$

satisfies the conditions $\lambda_1 = 1$ *and* $\lambda_d = 0$ *for* $d > z$ *and minimizes* Σ_1 *under this conditions if a minimum exists.*

Proof. We want to minimize

$$\Sigma_1 = \sum_{\substack{d_1|P(z) \\ d_2|P(z) \\ D=[p_1,p_2]}} \lambda_{d_1}\lambda_{d_2}\frac{\omega(D)}{D} = \sum_{D\leq z^2}\left(\sum_{\substack{d_1,d_1\leq z \\ [d_1,d_2]=D}} \lambda_{d_1}\lambda_{d_2}\right)\frac{\omega(D)}{D}.$$

Next we note that for any multiplicative function f and $a, b \in \mathbb{N}$ it is true that

$$f([a,b])f((a,b)) = f(a)f(b).$$

This holds since if we set $q_1, \ldots, q_2$ to be all the different prime numbers occurring in a and b, i.e.

$$a = \prod_{i=1}^{s} q_i^{a_i} \text{ and } b = \prod_{j=1}^{s} q_i^{b_i}$$

where $a_i, b_i \geq 0$ for $i \in \{1, \dots, s\}$. By definition of the greatest common divisor and the least common multiple we then want to show that

$$\prod_{i=1}^{s} f(q_i^{\max(a_i,b_i)}) \prod_{j=1}^{s} f(q_i^{\min(a_j,b_j)}) = \prod_{i=1}^{s} f(q_i^{a_i}) \prod_{j=1}^{s} f(q_i^{b_j})$$

but this is obviously true since for any pair (a_i, b_i) exactly the factors $f(q_i^{a_i})$ and $f(q_i^{b_i})$ occur on the right hand side and also on the left hand side because the tuple $(\min(a_i, b_i), \max(a_i, b_i))$ is either (a_i, b_i) or (b_i, a_i).

Using this and the fact that $\frac{\omega(D)}{D}$ is a multiplicative function we may write

$$\Sigma_1 = \sum_{d_1,d_2 \leq z} \frac{\lambda_{d_1}\omega(d_1)}{d_1} \frac{\lambda_{d_2}\omega(d_2)}{d_2} \frac{(d_1, d_2)}{\omega((d_1, d_2))}.$$

Now we have a look at

$$\sum_{d|r} \frac{1}{g(d)} = \sum_{d|r} \frac{d}{\omega(d)} \prod_{p|d} \left(1 - \frac{\omega(p)}{p}\right) = \sum_{d|r} \frac{d}{\omega(d)} \sum_{k|d} \mu(k) \frac{\omega(k)}{k} =$$

$$= \sum_{d|r} \sum_{k|d} \mu(k) \frac{\frac{d}{k}}{\omega(\frac{d}{k})} = \sum_{s|r} \frac{s}{\omega(s)} \sum_{t|\frac{r}{s}} \mu(t) = \frac{r}{\omega(r)}$$

where for the last equality we used Lemma 2.1.3. Putting things together we arrive at

$$\Sigma_1 = \sum_{d_1,d_2 \leq z} \frac{\lambda_{d_1}\omega(d_1)}{d_1} \frac{\lambda_{d_2}\omega(d_2)}{d_2} \sum_{\substack{r|d_1 \\ r|d_2}} \frac{1}{g(r)} = \sum_{r \leq z} \frac{1}{g(r)} \left(\sum_{\substack{d \leq z \\ r|d}} \frac{\lambda_d \omega(d)}{d} \right)^2 =$$

$$= \sum_{r \leq z} \frac{1}{g(r)} y_r^2$$

where $y_r = \sum_{\substack{d \leq z \\ r|d}} \frac{\lambda_d \omega(d)}{d} = \sum_{m \leq \frac{z}{r}} \frac{\lambda_{rm}\omega(rm)}{rm}$.

What we want to do now is minimizing Σ_1 with respect to the y_r because given the optimal values for the y_r using

$$\frac{\lambda_d \omega(d)}{d} = \sum_{r \leq \frac{z}{d}} \mu(r) y_{rd}$$

we get the optimal values for the λ_d. The last equation holds because

$$\sum_{m \leq \frac{z}{r}} \frac{\lambda_{rm}\omega(rm)}{rm} = \sum_{m \leq \frac{z}{r}} \sum_{k \leq \frac{z}{rm}} \mu(k) y_{krm} = \sum_{\nu \leq \frac{z}{r}} y_{r\nu} \sum_{k|\nu} \mu(k) = y_r.$$

The constraint $\lambda_1 = 1$ now is of the form

$$F = \sum_{r \le z} \mu(r) y_r = 1.$$

To find a minimum using a Lagrange multiplier we have to solve the following system of equations

$$\frac{\partial \Sigma_1}{\partial y_r} + \eta \frac{\partial F}{\partial y_r} = 0,\ r \in \{1, 2, \ldots, \lfloor z \rfloor\}$$

and substitute the solutions $y_1(\eta), y_2(\eta), \ldots, y_r(\eta)$ in F to find η. Computing the partial derivatives yields

$$\frac{2y_r}{g(r)} + \eta\mu(r) = 0$$

hence

$$y_r(\eta) = -\frac{\eta\mu(r)g(r)}{2}.$$

Substituting these values for y_r in F we get

$$-\frac{\eta}{2} \sum_{r \le z} \mu^2(r) g(r) = 1$$

hence

$$\eta = -2 \left(\sum_{r \le z} \mu^2(r) g(r) \right)^{-1}$$

and

$$y_r = \mu(r)g(r) \left(\sum_{r \le z} \mu^2(r) g(r) \right)^{-1} = \frac{\mu(r)g(r)}{G(z)}.$$

If Σ_1 has a minimum under constraint F those are the values for y_r realizing it. Now given the y_r we can compute the λ_d as

$$\frac{\lambda_d \omega(d)}{d} = \sum_{r \le \frac{z}{d}} \frac{\mu(r)\mu(rd)g(rd)}{G(z)} = \frac{\mu(d)g(d)}{G(z)} \sum_{\substack{r \le \frac{z}{d} \\ (r,d)=1}} \mu^2(r) g(r) =$$

$$= \frac{\mu(d)\omega(d)}{G(z) d \prod_{p|d} \left(1 - \frac{\omega(p)}{p}\right)} G_d(\frac{z}{d}).$$

We divide both sides of the last equation by $\frac{\omega(d)}{d}$ and get

$$\lambda_d = \frac{\mu(d)}{\prod_{p|d}\left(1-\frac{\omega(p)}{p}\right)}\frac{G_d(\frac{z}{d})}{G(z)}$$

hence the Lemma. $\square$

Using the optimal choice for the λ_d from the Lemma above evaluating Σ_1 now yields

$$\Sigma_1 = \frac{1}{G(z)}.$$

It remains to get a bound for Σ_2. With

$$G(z) = \sum_{l|d} \sum_{\substack{m<z \\ (m,d)=l}} \mu^2(m)g(m) = \sum_{l|d} \sum_{\substack{h<\frac{z}{l} \\ (h,\frac{d}{l})=1 \\ (h,l)=1}} \mu^2(lh)g(lh) =$$

$$= \sum_{l|d} \mu^2(l)g(l)G_d\left(\frac{z}{l}\right) \geq G_d\left(\frac{z}{d}\right) \sum_{l|d} \mu^2(l)g(l)$$

and the equality

$$\sum_{l|d} \mu^2(l)g(l) = \frac{1}{\prod_{p|d}\left(1-\frac{\omega(p)}{p}\right)}$$

we get that

$$G_d\left(\frac{z}{d}\right) \leq G(z)\prod_{p|d}\left(1-\frac{\omega(p)}{p}\right)$$

hence for any squarefree $d \in \mathbb{N}$

$$|\lambda_d| \leq 1$$

and

$$\Sigma_2 \leq \sum_{\substack{d_1<z,\ d_2<z \\ d_1|P(z) \\ d_2|P(z)}} |R_{[d_1,d_2]}|.$$

The numbers $[d_1, d_2]$ occurring on the right hand side of the last inequality are squarefree, they divide $P(z)$ and they are less than z^2. Since they are squarefree the number of pairs (d_1, d_2) with $[d_1, d_2] = D$ for fixed D is exactly $3^{\nu(D)}$, where $\nu(D)$ counts the number of different prime factors in D. We therefore get

$$\Sigma_2 \leq \sum_{\substack{d_1<z,\, d_2<z \\ d_1|P(z) \\ d_2|P(z)}} |R_{[d_1,d_2]}| \leq \sum_{\substack{D<z^2 \\ D|P(z)}} 3^{\nu(D)}|R_D| \leq \sum_{\substack{D<z^2 \\ \forall p\in\mathbb{P}\setminus\mathcal{P}:(p,D)=1}} \mu^2(D)3^{\nu(D)}|R_D|.$$

Putting all this together we have (cf. [HR11, Theorem 3.2.]):

Theorem 4.3.1. *The following inequality holds:*

$$S(\mathcal{A},\mathcal{P},z) \leq \frac{X}{G(z)} + \sum_{\substack{D<z^2 \\ \forall p\in\mathbb{P}\setminus\mathcal{P}:(p,D)=1}} \mu^2(D)3^{\nu(D)}|R_D|.$$

4.3.2 Intermediate results

From now on we are going to restrict ourselves to the case $\omega(p) = \frac{p}{p-1}$. For any integer K we furthermore define

$$\mathcal{P}_K := \{p \in \mathbb{P} : p \nmid K\}.$$

In order to prove Theorem 4.3.2 we are going to need the following Lemma whose proof we omit here (it can be found in [HR11, Lemma 3.1] while for Theorem 4.3.2 and its proof we refer to [HR11, Theorem 3.10]).

Lemma 4.3.2. *Let K be an integer. With*

$$H_K(x) := \sum_{\substack{d<x \\ (d,K)=1}} \frac{\mu^2(d)}{\varphi(d)}$$

the following estimate

$$H_K(x) \geq \log x \prod_{p|K} \left(1 - \frac{1}{p}\right)$$

holds.

Theorem 4.3.2. *Let $K \neq 0$ be an even integer and let*

$$\omega(p) = \frac{p}{p-1}, \text{ for } p \in \mathcal{P}_K.$$

Then

$$\mathcal{S}(\mathcal{A},\mathcal{P}_K,z) \le 2\prod_{p>2}\left(1-\frac{1}{(p-1)^2}\right)\prod_{2<p|K}\frac{p-1}{p-2}\frac{X}{\log z}\left\{1+\mathcal{O}\left(\frac{1}{\log z}\right)\right\}+$$
$$+\sum_{\substack{d<z^2\\(d,K)=1}}\mu^2(d)3^{\nu(d)}|R_d|.$$

Proof. First we note that the function g is well defined with our choice of ω since K is even. We now use Theorem 4.3.1 and see that it is sufficient to prove that

$$\frac{1}{G(z)} \le 2\prod_{p>2}\left(1-\frac{1}{(p-1)^2}\right)\prod_{2<p|K}\frac{p-1}{p-2}\frac{1}{\log z}\left\{1+\mathcal{O}\left(\frac{1}{\log z}\right)\right\}.$$

We start out by having a closer look at $g(p)$ for $p \in \mathcal{P}_K$:

$$g(p) = \frac{\omega(p)}{p\left(1-\frac{\omega(p)}{p}\right)} = \frac{p}{p(p-1)\left(\frac{p-2}{p-1}\right)} = \frac{1}{p-2} = \frac{1}{p-1}\left(1+\frac{1}{p-2}\right) =$$
$$= \frac{1}{\varphi(p)}(1+g(p))$$

hence for d with $(d,K)=1$ we get

$$g(d) = \frac{1}{\varphi(d)}\sum_{l|d}\mu^2(l)g(l).$$

By definition of $G(z)$ we have

$$G(z) = \sum_{\substack{d<z\\(d,K)=1}}\frac{\mu^2(d)}{\varphi(d)}\sum_{l|d}\mu^2(l)g(l) = \sum_{\substack{l<z\\(l,K)=1}}\mu^2(l)g(l)\sum_{\substack{m<\frac{z}{l}\\(m,K)=1\\(m,l)=1}}\frac{\mu^2(lm)}{\varphi(lm)} =$$
$$= \sum_{\substack{l<z\\(l,K)=1}}\frac{\mu^2(l)g(l)}{\varphi(l)}\sum_{\substack{m<\frac{z}{l}\\(m,K)=1\\(m,l)=1}}\frac{\mu^2(m)}{\varphi(m)} = \sum_{\substack{l<z\\(l,K)=1}}\frac{\mu^2(l)g(l)}{\varphi(l)}H_{Kl}\left(\frac{z}{l}\right).$$

Applying Lemma 4.3.2 we arrive at

$$G(z) \ge \prod_{p|K}\left(1-\frac{1}{p}\right)\sum_{\substack{l<z\\(l,K)=1}}\frac{\mu^2(l)g(l)}{\varphi(l)}\log\frac{z}{l} \ge$$
$$\ge \prod_{p|K}\left(1-\frac{1}{p}\right)\sum_{\substack{l=1\\(l,K)=1}}^{\infty}\frac{\mu^2(l)g(l)}{l}\log\frac{z}{l}.$$

With $g(p) = \frac{1}{p-2}$ we additionally get

$$\sum_{\substack{l=1 \\ (l,K)=1}}^{\infty} \frac{\mu^2(l)g(l)}{l} = \prod_{p \nmid K} \left(1 + \frac{1}{p(p-2)}\right)$$

hence

$$\begin{aligned}
\sum_{\substack{l=1 \\ (l,K)=1}}^{\infty} \frac{\mu^2(l)g(l)}{l} \log l &= \sum_{\substack{l=1 \\ (l,K)=1}}^{\infty} \frac{\mu^2(l)g(l)}{l} \sum_{p|l} \log p = \\
&= \sum_{p \nmid K} \frac{\log p}{p(p-2)} \sum_{\substack{m=1 \\ (m,K)=1 \\ (m,p)=1}}^{\infty} \frac{\mu^2(m)g(m)}{m} = \\
&= \sum_{p_1 \nmid K} \frac{\log p_1}{p_1(p_1-2)} \frac{1}{1 + \frac{1}{p_1(p_1-2)}} \prod_{p_2 \nmid K} \left(1 + \frac{1}{p_2(p_2-2)}\right).
\end{aligned}$$

Putting everything together yields

$$G(z) \geq \prod_{p|K} \left(1 - \frac{1}{p}\right) \prod_{p \nmid K} \left(1 + \frac{1}{p(p-2)}\right) \left(\log z - \sum_{p \in \mathbb{P}} \frac{\log p}{(p-1)^2}\right).$$

Dividing the last inequality by both its sides and simplifying the expressions proves the theorem. $\square$

A proof of the following Lemma can again be found in [HR11, Lemma 3.5]. It will be useful to get an estimate for the remainder term in Theorem 4.3.1 for our setting.

Lemma 4.3.3 (Bombieri)**.** *Let*

$$E(x,d) := \max_{2 \leq y \leq x} \max_{(l,d)=1} \left| \pi(y;d,l) - \frac{\operatorname{li} y}{\varphi(d)} \right|$$

and h, k be positive integers. If $k \leq \log^A x$ given any constant U there exists a positive constant C depending on U, h and A such that

$$\sum_{d < \frac{\sqrt{x}}{k \log^C x}} \mu^2(d) h^{\nu(d)} E(x, dk) = \mathcal{O}_{U,h,A}\left(\frac{x}{\varphi(k) \log^U x}\right).$$

4.4 An upper-bound sieve

With the intermediate results in the previous section we are now ready to prove the upper bound for the set $\mathcal{A}$ we introduced in Section 4.2 (for Theorem 4.4.1 and its proof we refer to [HR11, Theorem 3.12]).

Theorem 4.4.1. *Let b, k and l be integers where b is non-zero and even. Let furthermore k and l be coprime with*

$$1 \leq k \leq \log^A x.$$

For $x \to \infty$ we have, uniformly in b, k and l

$$|\{p \in \mathbb{P} : p \leq x, p \equiv l \bmod k, p + b = p' \in \mathbb{P}\}| \leq$$

$$\leq 8 \prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right) \prod_{2<p|kb} \frac{p-1}{p-2} \frac{x}{\varphi(k)\log^2 x} \left(1 + \mathcal{O}_A\left(\frac{\log\log x}{\log x}\right)\right).$$

Proof. As in Section 4.2 we choose

$$\mathcal{A} := \{p + b : p \in \mathbb{P}, p \leq x, p \equiv l \bmod k\}$$

and we choose

$$X := \frac{\operatorname{li} x}{\varphi(k)},\ \omega(p) := \frac{p}{p-1} \text{ for } p \text{ with } (p, kb) = 1.$$

We have seen that with this choice we get

$$|R_d| \leq E(x, dk).$$

Now

$$|\{p \in \mathbb{P} : p \leq x, p \equiv l \bmod k, p + b = p' \in \mathbb{P}\}| \leq \mathcal{S}(A, \mathcal{P}_{kb}, z) + z$$

because $\mathcal{S}(A, \mathcal{P}_{kb}, z)$ counts at least all primes larger than z occurring in the sequence $\mathcal{A}$. Since $2|kb$ we may apply Theorem 4.3.2 and get

$$|\{p \in \mathbb{P} : p \leq x, p \equiv l \bmod k, p + b = p' \in \mathbb{P}\}| \leq$$

$$2 \prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right) \prod_{2<p|kb} \frac{p-1}{p-2} \frac{\operatorname{li} x}{\varphi(k)\log z} \left(1 + \mathcal{O}_A\left(\frac{1}{\log z}\right)\right) +$$

$$+ \sum_{d<z^2} \mu^2(d) 3^{\nu(d)} E(x, dk) + z.$$

For sufficiently large x we may use Lemma 4.3.3 with $U = h = 3$ and get the existence of a constant C depending on A such that with

$$z^2 = \frac{\sqrt{x}}{k \log^C x}$$

we have

$$\sum_{d<z^2} \mu^2(d) 3^{\nu(d)} E(x, dk) = \mathcal{O}_A\left(\frac{x}{\varphi(k)\log^3 x}\right).$$

With

$$\log z = \frac{1}{4}\log x \left(1 + \mathcal{O}_A\left(\frac{\log\log x}{\log x}\right)\right)$$

we finally get

$$\frac{\operatorname{li} x}{\log z} = \frac{4x}{\log^2 x}\left(1 + \mathcal{O}_A\left(\frac{\log\log x}{\log x}\right)\right)$$

hence the Theorem. $\square$

5

The general situation - numbers of the form $p + g^n$

5.1 Application of the sieve

For $g \in \mathbb{N}$, $g \geq 2$ we recall that

$$\underline{d_g} := \liminf_{x \to \infty} \frac{|\{n \leq x : r_g(n) > 0\}|}{x}$$

and

$$\overline{d_g} := \limsup_{x \to \infty} \frac{|\{n \leq x : r_g(n) > 0\}|}{x}.$$

Our goal is to get a lower bound for $\underline{d_g}$ by applying the idea of Elsholtz and Schlage-Puchta to the general case. We are basically going to work through their proof of a lower bound for Romanov's constant given in [ESP12].

Romanov [Rom34] proved that the lower density of integers representable as the sum of a prime and a power of g is positive but few is known on explicit values in the case of $g \neq 2$. In an unpublished manuscript [GL03] Gugg and Ledoan compute values for $g = 2$ and $g = 3$. They come up with $\underline{d_2} \geq 0.002815971796$ and $\underline{d_3} \geq 0.001259610985$. The main result of this chapter will be the following theorem giving lower bounds for $\underline{d_g}$ for $g \in \{3, \ldots, 18\}$.

Theorem 5.1.1. *The following lower bounds for $\underline{d_g}$ hold:*

g	3	4	5	6
$\underline{d_g} \geq$	0.073588	0.069875	0.081702	0.057118
g	7	8	9	10
$\underline{d_g} \geq$	0.060297	0.081948	0.067671	0.056825
g	11	12	13	14
$\underline{d_g} \geq$	0.074303	0.060415	0.059801	0.066363
g	15	16	17	18
$\underline{d_g} \geq$	0.048125	0.052867	0.110158	0.063678

Table 5.1: Lower bounds for $\underline{d_g}$.

Even though the lower bounds for the lower densities seems to be somehow decreasing there are g with relatively high lower bounds. Especially the good lower bound for integers representable as the sum of a prime and a power of 17 was quite surprising. First empirical tests suggest that the Elsholtz and Schlage-Puchta method works very well for g for which $r_g(n)$ is typically low (remember that $r_g(n)$ counts the number of possibilities to write n as the sum of a prime and a power of g).

The proof of Theorem 5.1.1 is the concern of this chapter. For the rest of this chapter let $g \in \mathbb{N}$, $g \geq 2$ be fixed. We start out by considering the sums

$$S_g^{(1)}(x,k,l) := \sum_{\substack{n \leq x \\ n \equiv k \bmod l}} r_g(n)$$

and

$$S_g^{(2)}(x,k,l) := \sum_{\substack{n \leq x \\ n \equiv k \bmod l}} r_g^2(n)$$

The integer l shall be chosen in the form $l = g^m - 1$ for some $m \in \mathbb{N}$ and k and l shall be coprime. This ensures that l and g are coprime and we can split $S_g^{(2)}(x,k,l)$ as follows:

$$\begin{aligned}
S_g^{(2)}(x,k,l) = |\{p_1 + g^{a_1} =& p_2 + g^{a_2} \equiv k \bmod l, p_1 + g^{a_1} \le x\}| = \\
= \sum_{\substack{\kappa \le l \\ \alpha \le \epsilon_g(l) \\ \kappa + g^\alpha \equiv k \bmod l}} & |\{p + g^a \equiv k \bmod l, p \equiv \kappa \bmod l, \\
& a \equiv \alpha \bmod \epsilon_g(l), p + g^a \le x\}| + \\
+ \sum_{\substack{\kappa_1,\kappa_2 \le l \\ \alpha_1,\alpha_2 \le \epsilon_g(l) \\ \kappa_i + g^{\alpha_i} \equiv k \bmod l}} & |\{p_1 + g^{a_1} = p_2 + g^{a_2}, p_1 \ne p_2, p_i \equiv \kappa_i \bmod l, \\
& a_i \equiv \alpha_i \bmod \epsilon_g(l), p_1 + g^{a_1} \le x\}|.
\end{aligned}$$

For the second sum on the righthand side of the last equation we fix κ_1, κ_2, a_1 and a_2 such that $\kappa_1 + g^{a_1} \equiv \kappa_2 + g^{a_2} \bmod l$. By Theorem 4.4.1 the number of primes $p_i \equiv \kappa_i \bmod l$ with $p_1 - p_2 = g^{a_1} - g^{a_2}$ is bounded from above by

$$C_1 \prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right) \prod_{2<p|l(g^{a_1}-g^{a_2})} \frac{p-1}{p-2} \frac{x}{\varphi(l)\log^2 x}$$

where C_1 is a constant tending to 8 as x tends towards infinity. For

$$\prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right)$$

an upper bound is known (as in [ESP12] we will take the upper bound 0.6601) hence it remains to find an estimate for

$$\prod_{2<p|l(g^{a_1}-g^{a_2})} \frac{p-1}{p-2}.$$

To get this estimate it suffices to consider $a_1, a_2 \le L_g := \frac{\log x}{\log g}$ and we have

$$\sum_{\substack{a_1,a_2 \le L_g \\ a_i \equiv \alpha_i \bmod m}} \prod_{2<p|l(g^{a_2}-g^{a_2})} \frac{p-1}{p-2} =$$

$$= \prod_{2<p|l} \frac{p-1}{p-2} \prod_{2<p|g} \frac{p-1}{p-2} \sum_{\substack{a_1<a_2\le L_g \\ a_i \equiv \alpha_i \bmod m}} \sum_{\substack{(d,2lg)=1 \\ d|(g^{a_1-a_2}-1)}} f(d) \sim$$

$$\sim C_2(l,g) \frac{L_g^2}{2} \frac{1}{m^2} \sum_{\substack{(d,2lg)=1 \\ (\epsilon_g(d),m)|(\alpha_1-\alpha_2,m)}} \frac{f(d)(\epsilon_g(d),m)}{\epsilon_g(d)}$$

$$= C_2(l,g) \frac{L_g^2}{2} \frac{1}{m^2} S_g((\alpha_1-\alpha_2,m),m)$$

where $C_2(l,g) = \prod_{2<p|l} \frac{p-1}{p-2} \prod_{2<p|g} \frac{p-1}{p-2}$,

$$f(n) := \begin{cases} \prod_{p|n} \frac{1}{p-2}, & \text{if } \mu^2(n) = 1 \text{ and } n \text{ is odd} \\ 0, & \text{otherwise} \end{cases}$$

and $S_g(t,m) := \sum_{\substack{(d,2(g^m-1))=1 \\ (\epsilon_g(d),m)|t}} \frac{f(d)(\epsilon_g(d),m)}{\epsilon_g(d)}$.

Now as in [ESP12] we sum over all possible choices of κ_1, κ_2, α_1 and α_2 and obtain the analogous results

$$\begin{aligned} S_g^{(1)}(x,k,l) &\sim \frac{x}{\varphi(l) m \log g} |\{\kappa, \alpha | (\kappa, l) = 1, \kappa + g^\alpha \equiv k \bmod l\}| \\ S_g^{(2)}(x,k,l) &\le \frac{x}{\varphi(l) m \log g} |\{\kappa, \alpha | (\kappa, l) = 1, \kappa + g^\alpha \equiv k \bmod l\}| + \\ &\quad + C_1 C_2(l,g) \prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right) \frac{x}{\varphi(l) m^2 \log^2 g} \cdot \\ &\quad \cdot \sum_{\substack{\kappa_1 + g^{\alpha_1} \equiv k \bmod l \\ \kappa_2 + g^{\alpha_2} \equiv k \bmod l}} S_g((\alpha_1 - \alpha_2, m), m) \end{aligned}$$

and again finding an upper bound for $S_g^{(2)}(x,k,l)$ amounts to finding upper bounds for $S_g(t,m)$.

5.2 Bounding the sum $S_g(t,m)$

The quality of our lower bound for the density of integers of the form $p+g^n$ depends essentially on the quality of the upper bound we can give for the sums $S_g(t,m)$, with $t|m$. Now the idea of Elsholtz and Schlage-Puchta in the case of $g=2$ was splitting these sums in four parts, computing two of them explicitly and giving upper bounds for the other two and again we will follow their steps. For fixed $t|m$ we split $S_g(t,m)$ as follows:

$$S_g(t,m) = \sum_{(d,m)|t} \frac{(d,m)}{d} \sum_{\substack{\epsilon_g(n)=d \\ (n,2(g^m-1))=1}} f(n) = \sum_{\substack{d<D_1 \\ (d,m)|t}} \frac{(d,m)}{d} \sum_{\substack{\epsilon_g(n)=d \\ (n,2(g^m-1))=1}} f(n) +$$

$$+ \sum_{\substack{D_1 \le d \le D_2 \\ (d,m)|t}} \frac{(d,m)}{d} \sum_{\substack{\epsilon_g(n)=d \\ (n,2(g^m-1))=1 \\ P^+(n) \le P}} f(n) + \sum_{\substack{D_1 \le d \le D_2 \\ (d,m)|t}} \frac{(d,m)}{d} \sum_{\substack{\epsilon_g(n)=d \\ (n,2(g^m-1))=1 \\ P^+(n) > P}} f(n) +$$

$$+ \sum_{\substack{d > D_2 \\ (d,m)|t}} \frac{(d,m)}{d} \sum_{\substack{\epsilon_g(n)=d \\ (n,2(g^m-1))=1}} f(n).$$

Using a computer algebra system the first two of the last sums can be computed explicitly for appropriate values of D_1, D_2 and P depending on g and m and for the last two sums we will give two theoretical results.

5.2.1 Theoretical results

To get an upper bound for the last of the previous four sums, Elsholtz and Schlage-Puchta used Lemma 4 in [CS04]. We want to use a similar result and therefore work through Chen and Sun's proof of this Lemmma with general g.

Lemma 5.2.1. *For any integer g and positive real D let*

$$V_g(D) = \prod_{\frac{D}{2} < k \le D} (g^k - 1).$$

Let n be a lower bound for the number of prime factors of $V_g(D)$ and $\hat{p}_n$ the $n-th$ prime. Then we have for any $D \ge 75$

$$\sum_{\epsilon_g(n) > D} \frac{f(n)}{\epsilon_g(n)} \le \frac{0.922913686}{0.66016} \left(2 \left(\frac{1}{D} + \frac{\log D}{D} \right) + \frac{C_1}{D} \right)$$

where $C_1 = \log \left(\log(g) \frac{\log \hat{p}_n}{\log \hat{p}_n - C_2} \frac{77}{200} \right)$ and C_2 is a constant depending on the number of prime factors of $V_g(D)$[1].

[1] We leave the fraction $\frac{0.922913686}{0.66016}$ since $\prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right) = 0.66016\ldots$ is a well known constant in prime number theory arising from the study of twin primes.

Proof. To begin with we introduce the sums

$$w_g(r) = \sum_{\epsilon_g(d)=r} f(d)$$

and

$$W_g(r) = \sum_{s \le r} w_g(s).$$

Using this notation we are obviously looking for an upper bound of the sum

$$\sum_{r>D} \frac{w_g(r)}{r}.$$

To get this bound we have a closer look at $V_g(x) = \prod_{\frac{x}{2}<k\le x} (g^k - 1)$ and observe, that any $m \in \mathbb{N}$ with $\epsilon_g(m) \le x$ is a divisor of $V_g(x)$. Now let $q_1, \ldots, q_n$ be the different prime divisors of $V_g(x)$ and $p_1, \ldots, p_n$ be the first n prime numbers and we get

$$p_1 \cdot \ldots \cdot p_n \le q_1 \cdot \ldots \cdot q_n \le V_g(x) < \prod_{\frac{x}{2}<k\le x} g^k.$$

Taking the logarithm on both sides of the above inequality we obtain

$$\sum_{i=1}^{n} \log p_i < \log g \sum_{\frac{x}{2}<k\le x} k.$$

We now use a lower bound from Corollary 2 in [Sch76] which states

$$\sum_{i=1}^{n} \log p_i > p_n \left(1 - \frac{C_2}{\log p_n}\right)$$

where explicit values for C_2 depending on n can be found in a table in [Sch76]. With this we obtain

$$\left(1 - \frac{C_2}{\log \hat{p}_n}\right) p_n \le \left(1 - \frac{C_2}{\log p_n}\right) p_n < \log g \sum_{\frac{x}{2}<k\le x} k \le$$

$$\le \log(g) \left(\frac{3x^2}{8} + \frac{3x}{4}\right) \le \log(g) \left(\frac{3x^2}{8} + \frac{x^2}{100}\right)$$

where in the last inequality we needed that $x \ge 75$. For p_n we therefore get

$$p_n < \left(\frac{\log \hat{p}_n}{\log \hat{p}_n - C_2}\right) \log(g) \left(\frac{3x^2}{8} + \frac{x^2}{100}\right).$$

Analogous as in the proof of Lemma 4 in [CS04] for $x \geq 74$ we have as well

$$W_g(x) \leq \frac{0.922913686}{0.66016} \log p_n.$$

With our bound for p_n from above we altogether get

$$W_g(x) \leq \frac{0.922913686}{0.66016}(C_1 + 2\log x).$$

In a next step we use partial summation (see for example [Brü95] Lemma 1.1.3) and the fact that $\frac{W_g(n)}{n}$ tends to zero for n tending to infinity (note that $0 \leq \frac{W_g(n)}{n} \leq \frac{\frac{0.922913686}{0.66016}(C_1+2\log n)}{n}$). We arrive at

$$\begin{aligned}\sum_{D\leq n} \frac{w_g(n)}{n} &= \int_D^\infty \frac{W_g(u)}{u^2} du \leq \int_D^\infty \frac{\frac{0.922913686}{0.66016}(C_1 + 2\log u)}{u^2} du = \\ &= \frac{0.922913686}{0.66016}\left(2\left(\frac{1}{D} + \frac{\log D}{D}\right) + \frac{C_1}{D}\right)\end{aligned}$$

as desired. □

Values for $\hat{p}_n$ and a lower bound for the number of prime divisors of $V_g(D)$ as needed in the previous Lemma can easily be found using computer algebra systems.

To deal with the third sum we use the following Lemma which is an analogue to [ESP12, Lemma 2] and again the proof is done by working through the original proof by Elsholtz and Schlage-Puchta for general g.

Lemma 5.2.2. *Let t, D_1, D_2 and P be integers where $D_1 \geq 4$ and $D_1 < D_2$. Then we have*

$$\begin{aligned}\sum_{\substack{D_1 \leq \epsilon_g(n) \leq D2 \\ P^+(n) > P \\ (n, 2(g^m-1))=1 \\ (\epsilon_g(n), 24) \mid t}} \frac{f(n)(\epsilon_g(n), m)}{\epsilon_g(n)} \leq \frac{D_1 \log g}{2(P-2)\log P}\Sigma_1^{(g)} + \frac{1}{(P-2)\log P}\Sigma_2^{(g)} + \\ + t\frac{0.922913686}{0.66016}(\log D_1 + 1 + \frac{C_1}{2})\frac{D_1 \log g}{(P-2)\log P} + \\ + \frac{0.922913686}{0.66016}\frac{2t(D_2 \log D_2 + 2D_2 + \frac{C_1 D_2}{2})}{(P-2)\log P}\end{aligned}$$

where

$$\Sigma_1^{(g)} = \sum_{\substack{(n,2(g^m-1))=1 \\ (\epsilon_g(n),m)|t \\ \epsilon_g(n)\le D_1}} f(n)(\epsilon_g(n), m)$$

and

$$\Sigma_2^{(g)} = \sum_{D_1\le d\le D_2} \frac{\varphi(d)}{d} \sum_{\substack{(n,2(g^m-1))=1 \\ (\epsilon_g(n),m)|t \\ \epsilon_g(n)\le d}} f(n)([\epsilon_g(n), d], m)$$

.

Proof. To begin with we fix $p \in \mathbb{P}$ and define

$$g_g(n) := ([\epsilon_g(n), \epsilon_g(p)], m)$$

$$h_g(n) := [\epsilon_g(n), \epsilon_g(p)]$$

$$\mu_g := \max(D_1, \epsilon_g(p))$$

then

$$\sum_{\substack{D_1\le\epsilon_g(n)\le D_2 \\ p|n \\ (n,2(g^m-1))=1 \\ (\epsilon_g(n),m)|t}} \frac{f(n)(\epsilon_g(n), m)}{\epsilon_g(n)} = \frac{1}{p-2} \sum_{\substack{D_1\le h_g(n)\le D_2 \\ p\nmid n \\ (n,2(g^m-1))=1 \\ g_g(n)|t}} \frac{f(n)g_g(n)}{h_g(n)} \le$$

$$\le \frac{1}{p-2} \sum_{\substack{D_1\le h_g(n)\le D_2 \\ p\nmid n \\ (n,2(g^m-1))=1 \\ g_g(n)|t}} \frac{f(n)g_g(n)}{\max(\mu_g, \epsilon_g(n))} \le$$

$$\le \frac{1}{p-2}(\frac{1}{\mu_g} \sum_{\substack{D_1\le h_g(n)\le D_2 \\ p\nmid n \\ (n,2(g^m-1))=1 \\ g_g(n)|t \\ \epsilon_g(n)\le\mu_g}} f(n)g_g(n) + tC(\mu_g))$$

where the first inequality holds because of Lemma 2.1.1 and $C(\mu_g)$ is a constant depending on μ_g we get from Lemma 5.2.1. Since

$$P^n \le g^d - 1 \Leftrightarrow n \le \frac{\log(g^d-1)}{\log P}$$

we have at most $\frac{d \log g}{\log P}$ primes $p > P$ with $p | g^d - 1$. In the case $\epsilon_g(p) < D_1$ we get $\mu_g = D_1$. The number of those primes is at most

$$\sum_{d<D_1} \frac{d \log g}{\log P} \leq \frac{\log g}{\log P} \frac{D_1(D_1-1)}{2} < \frac{\log g}{\log P} \frac{D_1^2}{2}$$

so those primes contribute at most

$$\frac{D_1^2 \log g}{2(P-2)\log P} \Big(\frac{1}{D_1} \sum_{\substack{(n,2(g^m-1))=1 \\ (\epsilon_g(n),m)|t \\ \epsilon_g(n) \leq D_1}} f(n)(\epsilon_g(n), m) + tC(D_1) \Big)$$

to the sum. It remains to deal with the primes p with $\epsilon_g(p) = d$ and $D_1 \leq d \leq D_2$. Since there are at most $\frac{\varphi(d) \log g}{\log P}$ such primes we get a contribution of at most

$$\frac{\varphi(d) \log g}{(P-2)\log P} \Big(\frac{1}{d} \sum_{\substack{(n,2(g^m-1))=1 \\ g_g(n)|t \\ \epsilon_g(n) \leq d}} f(n) g_g(n) + tC(d) \Big)$$

and we need to sum over all d with $D_1 \leq d \leq D_2$. Using $\frac{\varphi(d)}{d} \leq 1$ and

$$\sum_{D_1 \leq d \leq D_2} \log d \leq \int_{D_1-1}^{D_2} \log t dt =$$

$$= D_2 \log D_2 - D_2 - (D_1-1)\log(D_1-1) + D_1 - 1 \leq$$
$$\leq D_2 \log D_2$$

for $D_1 \geq e+1$ we get an upper bound of

$$\frac{\log g}{(P-2)\log P} \sum_{D_1 \leq d \leq D_2} \frac{\varphi(d)}{d} \sum_{\substack{(n,2(g^m-1))=1 \\ g_g(n)|t \\ \epsilon_g(n) \leq d}} f(n)([\epsilon_g(n), d], m) +$$

$$+ \frac{0.922913686}{0.66016} \frac{2t \log g (D_2 \log D_2 + 2D_2 + \frac{C_1 D2}{2})}{(P-2)\log P}.$$

$\square$

5.3 Computational results

This section describes the main ideas behind the computation of the upper bound for $S_g(t, m)$ and hence a lower bound for $\underline{d_g}$.

5.3.1 Theoretical results used to improve computational efficiency

The first detail we need to have a closer look at is the computation of sums of the form $\sum_{\epsilon_g(d)=e} f(d)$. Computing these sums directly is practically impossible because it involves computing the primitive squarefree divisors of integers of the form $g^n - 1$, i.e. all divisors $d | g^n - 1$ with $d \nmid g^m - 1$ for every positive integer $m < n$. To do this we would need to store, search and update a large set of previous divisors which is inefficient for reasonable values of D_1 and D_2. Prof. Schlage-Puchta suggested a more elegant and efficient way of computing these sums. The following idea is also explained in the latest version of their paper. They used the following Möbius inversion like identity (cf. Theorem 2.1.2):

Lemma 5.3.1. *The following equation holds true for any positive integer e:*

$$\sum_{\epsilon_g(d)=e} f(d) = \sum_{t|e} \mu\left(\frac{e}{t}\right) \sum_{\epsilon_g(d)|e} f(d).$$

Proof. This holds true since

$$\sum_{\epsilon_g(d)=e} f(d) = \sum_{\epsilon_g(d)|e} f(d) \sum_{t|\frac{e}{\epsilon_g(d)}} \mu\left(\frac{e}{t\epsilon_g(d)}\right) = \sum_{t|e} \mu\left(\frac{e}{t}\right) \sum_{\epsilon_g(d)|t} f(d)$$

where the first equation holds because of Lemma 2.1.3 and the second results from just exchanging the order of summation. □

The above Lemma allows to compute the sums $\sum_{\epsilon_g(d)=e} f(d)$ using the values of the sums $\sum_{\epsilon_g(d)|e} f(d)$ which can be computed efficiently from the divisors of the numbers $g^e - 1$.

We will use the multiplicativity of f to compute the values of its summatory function as a product instead a sum, i.e.

$$\sum_{\epsilon_g(d)=e} f(n) = \prod_{\substack{p^\nu || g^e-1 \\ p \neq 2}} \left(1 + \frac{1}{p-2}\right).$$

As Elsholtz and Schlage-Puchta we use the following Lemma of Pintz to improve the use of the Cauchy-Schwarz inequality, which is more a numerical improvement then an improvement of computational efficiency (cf. [Pin06, Lemma 4']):

Lemma 5.3.2. *Suppose* $b(n) \in \mathbb{N}_0$ *for* $n \in \mathbb{N}$ *and*

$$\sum_{n=1}^{N} b(n) = M, \ \sum_{n=1}^{N} b(n)^2 \leq DM.$$

Then we have

$$\#\{n \in \mathbb{N} : b(n) > 0\} \leq \delta(D)M$$

with

$$\delta(D) = \frac{\lfloor D \rfloor + \lceil D \rceil - D}{\lfloor D \rfloor \lceil D \rceil}.$$

Besides of these theoretical results large parts of the implementation are suited for parallel computation. Examples would be the computation of the integer factorizations, computing sums from precomputed values and the application of Lemma 5.3.1. In sage we can use the @parallel decorator to use parallel computation for the functions implementing these tasks.

5.3.2 Tables for different choices of g

All computations were done in Sage 6.3 on a machine running Ubuntu 14.04 using an Intel Core i7-4700MQ CPU and 16 GB RAM. The Sage code can be found in the appendix. The tables have a similar structure as the one given in [ESP12]: The first line gives the lower bound of $S_g(t, m)$ by summing over all integers n with $\epsilon_g(n) \leq D_1$. The second row shows the improved lower bound which is derived by additionally including those n with $D_1 < \epsilon_g(n) \leq D_2$ and $P^+(n) \leq$ Pmax. The third and fourth rows show upper bounds for $\Sigma_1^{(g)}$ and $\Sigma_2^{(g)}$ from Lemma 5.2.2 and the last row shows the upper bound we get by additionally using the bound from Lemma 5.2.1. The parameters m, D_1, D_2 and Pmax were chosen in a way to guarantee a reasonable computation time. To get the bound in the last row we need to combine $\Sigma_1^{(g)}$ and $\Sigma_2^{(g)}$ according to Lemma 5.2.2, add this value to the lower bound given in the second row and finally add the bound from Lemma 5.2.1.

t	1	2	3	4	6	12
$S_3(t,12) \geq$	1.030197	1.037333	1.033488	1.079253	1.087416	1.135565
$S_3(t,12) \geq$	1.030432	1.037742	1.033876	1.081058	1.090051	1.141829
$\Sigma_1^{(3)}$	1.317828	1.477935	1.455201	2.487499	3.101464	4.596925
$\Sigma_2^{(3)}$	4322.36	7442.61	9554.89	20487.87	29770.15	73652.41
$S_3(t,12) \leq$	1.033424	1.04092	1.03724	1.084614	1.093982	1.146896

Table 5.2: Computational results for $g = 3$ for the following choice of parameters: $m = 12$, Pmax $= 10^8$,$D_1 = 200$ and $D_2 = 10^4$. This leads to a lower bound of 0.073588 for the lower density of integers representable as the sum of a prime and a power of 3. The time needed to compute this bound was 4h 54m 38s.

t	1	2	3	4	6	12
$S_4(t,12) \geq$	1.044461	1.061943	1.077189	1.064669	1.114484	1.126143
$S_4(t,12) \geq$	1.045144	1.063684	1.079379	1.066886	1.120083	1.133619
$\Sigma_1^{(4)}$	1.409361	1.753333	1.940290	1.808988	2.815383	3.161645
$\Sigma_2^{(4)}$	4961.75	10158.26	14340.09	17101.24	34780.76	61386.97
$S_4(t,12) \leq$	1.048215	1.066994	1.082927	1.07067	1.12435	1.139315

Table 5.3: Computational results for $g = 4$ for the following choice of parameters: $m = 12$, Pmax $= 10^8$,$D_1 = 100$ and $D_2 = 10^4$. This leads to a lower bound of 0.069875 for the lower density of integers representable as the sum of a prime and a power of 4. The time needed to compute this bound was 3h 43m 43s.

t	1	2	3	6
$S_5(t,6) \geq$	1.028001	1.104068	1.054138	1.150444
$S_5(t,6) \geq$	1.028343	1.106953	1.055179	1.159859
$\Sigma_1^{(5)}$	1.211916	2.075240	1.595227	3.315778
$\Sigma_2^{(5)}$	3999.11	15664.63	10613.66	55509.93
$S_5(t,6) \leq$	1.031479	1.110373	1.05887	1.164413

Table 5.4: Computational results for $g = 5$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 100$ and $D_2 = 10^4$. This leads to a lower bound of 0.081702 for the lower density of integers representable as the sum of a prime and a power of 5. The time needed to compute this bound was 3h 6m 37s.

t	1	2	3	6
$S_6(t,6) \geq$	1.007996	1.068560	1.030853	1.156635
$S_6(t,6) \geq$	1.008325	1.071183	1.031781	1.167115
$\Sigma_1^{(6)}$	1.138097	1.963645	1.426712	3.576626
$\Sigma_2^{(6)}$	3779.54	14683.40	9571.61	56735.60
$S_6(t,6) \leq$	1.011509	1.074685	1.035585	1.171887

Table 5.5: Computational results for $g = 6$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 100$ and $D_2 = 10^4$. This leads to a lower bound of 0.057118 for the lower density of integers representable as the sum of a prime and a power of 6. The time needed to compute this bound was 4h 26m 52s.

t	1	2	3	6
$S_7(t,6) \geq$	1.008074	1.229033	1.027334	1.336181
$S_7(t,6) \geq$	1.008391	1.232697	1.028409	1.348882
$\Sigma_1^{(7)}$	1.126026	2.864722	1.430407	5.042929
$\Sigma_2^{(7)}$	3705.64	20065.62	9675.87	75545.99
$S_7(t,6) \leq$	1.011614	1.236272	1.032309	1.353856

Table 5.6: Computational results for $g = 7$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 100$ and $D_2 = 10^4$. This leads to a lower bound of 0.060297 for the lower density of integers representable as the sum of a prime and a power of 7. The time needed to compute this bound was 3h 10m 41s.

t	1	2	4	8
$S_8(t,8) \geq$	1.024447	1.077194	1.114253	1.126134
$S_8(t,8) \geq$	1.024932	1.079382	1.120642	1.134060
$\Sigma_1^{(8)}$	1.250673	1.899520	2.861352	3.166677
$\Sigma_2^{(8)}$	5681.73	14603.33	32106.20	48443.71
$S_8(t,8) \leq$	1.028186	1.083005	1.125001	1.139872

Table 5.7: Computational results for $g = 8$ for the following choice of parameters: $m = 8$, Pmax $= 10^8$,$D_1 = 100$ and $D_2 = 10^4$. This leads to a lower bound of 0.081948 for the lower density of integers representable as the sum of a prime and a power of 8. The time needed to compute this bound was 3h 31m 19s.

t	1	2	3	6
$S_9(t,6) \geq$	1.037051	1.078971	1.086906	1.135054
$S_9(t,6) \geq$	1.037743	1.081073	1.090071	1.141894
$\Sigma_1^{(9)}$	1.359175	1.863957	2.208383	2.956114
$\Sigma_2^{(9)}$	4779.35	14174.54	16475.80	49129.83
$S_9(t,6) \leq$	1.041026	1.084747	1.094128	1.14713

Table 5.8: Computational results for $g = 9$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 100$ and $D_2 = 10^4$. This leads to a lower bound of 0.067671 for the lower density of integers representable as the sum of a prime and a power of 9. The time needed to compute this bound was 3h 18m 11s.

t	1	2	3	6
$S_{10}(t,6) \geq$	1.010255	1.040919	1.023283	1.080011
$S_{10}(t,6) \geq$	1.010669	1.043448	1.024521	1.088573
$\Sigma_1^{(10)}$	1.118329	1.687450	1.408524	2.653526
$\Sigma_2^{(10)}$	3727.45	13108.33	9625.91	45702.21
$S_{10}(t,6) \leq$	1.013981	1.047171	1.028638	1.093934

Table 5.9: Computational results for $g = 10$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 90$ and $D_2 = 10^4$. This leads to a lower bound of 0.056825 for the lower density of integers representable as the sum of a prime and a power of 10. The time needed to compute this bound was 4h 51m 28s.

t	1	2	3	6
$S_{11}(t,6) \geq$	1.004681	1.037197	1.007626	1.101581
$S_{11}(t,6) \geq$	1.005020	1.039454	1.008714	1.110337
$\Sigma_1^{(11)}$	1.076364	1.763010	1.216868	3.155543
$\Sigma_2^{(11)}$	3567.49	13302.01	8593.33	48872.69
$S_{11}(t,6) \leq$	1.008355	1.043218	1.012888	1.115813

Table 5.10: Computational results for $g = 11$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 90$ and $D_2 = 10^4$. This leads to a lower bound of 0.074303 for the lower density of integers representable as the sum of a prime and a power of 11. The time needed to compute this bound was 3h 28m 5s.

t	1	2	3	6
$S_{12}(t,6) \geq$	1.007583	1.224606	1.021063	1.264296
$S_{12}(t,6) \geq$	1.008071	1.228118	1.021957	1.277184
$\Sigma_1^{(12)}$	1.157413	2.435262	1.316613	3.706043
$\Sigma_2^{(12)}$	3956.46	17799.38	9048.31	61499.79
$S_{12}(t,6) \leq$	1.011427	1.231925	1.026184	1.282779

Table 5.11: Computational results for $g = 12$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 90$ and $D_2 = 10^4$. This leads to a lower bound of 0.060415 for the lower density of integers representable as the sum of a prime and a power of 12. The time needed to compute this bound was 3h 21m 59s.

t	1	2	3	6
$S_{13}(t,6) \geq$	1.007744	1.261664	1.014210	1.317593
$S_{13}(t,6) \geq$	1.008136	1.265514	1.015356	1.330976
$\Sigma_1^{(13)}$	1.122570	2.864503	1.320680	4.693841
$\Sigma_2^{(13)}$	3722.69	20369.99	9214.76	72251.80
$S_{13}(t,6) \leq$	1.011511	1.269359	1.019632	1.33668

Table 5.12: Computational results for $g = 13$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 90$ and $D_2 = 10^4$. This leads to a lower bound of 0.059801 for the lower density of integers representable as the sum of a prime and a power of 13. The time needed to compute this bound was 4h 17m 18s.

t	1	2	3	6
$S_{14}(t,6) \geq$	1.026546	1.062292	1.039852	1.113770
$S_{14}(t,6) \geq$	1.027021	1.065715	1.041523	1.125508
$\Sigma_1^{(14)}$	1.202567	1.834646	1.453945	2.815730
$\Sigma_2^{(14)}$	4024.82	14532.42	10171.73	51443.13
$S_{14}(t,6) \leq$	1.030414	1.069583	1.045845	1.131269

Table 5.13: Computational results for $g = 14$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 75$ and $D_2 = 10^4$. This leads to a lower bound of 0.066363 for the lower density of integers representable as the sum of a prime and a power of 14. The time needed to compute this bound was 5h 7m 27s.

t	1	2	4
$S_{15}(t,4) \geq$	1.032118	1.055917	1.138769
$S_{15}(t,4) \geq$	1.033028	1.059468	1.149475
$\Sigma_1^{(15)}$	1.301988	1.723907	3.066767
$\Sigma_2^{(15)}$	6160.21	14896.15	44572.85
$S_{15}(t,4) \leq$	1.03644	1.063366	1.154362

Table 5.14: Computational results for $g = 15$ for the following choice of parameters: $m = 4$, Pmax $= 10^8$,$D_1 = 75$ and $D_2 = 10^4$. This leads to a lower bound of 0.048125 for the lower density of integers representable as the sum of a prime and a power of 15. The time needed to compute this bound was 3h 23m 4s.

t	1	2	3	6
$S_{16}(t,6) \geq$	1.061956	1.064789	1.114447	1.126650
$S_{16}(t,6) \geq$	1.063689	1.066772	1.120169	1.133596
$\Sigma_1^{(16)}$	1.574904	1.608730	2.349005	2.556255
$\Sigma_2^{(16)}$	6160.10	12833.23	20340.40	44661.61
$S_{16}(t,6) \leq$	1.067109	1.070685	1.124577	1.139491

Table 5.15: Computational results for $g = 16$ for the following choice of parameters: $m = 6$, Pmax $= 10^8$,$D_1 = 75$ and $D_2 = 10^4$. This leads to a lower bound of 0.052867 for the lower density of integers representable as the sum of a prime and a power of 16. The time needed to compute this bound was 3h 36m 7s.

t	1	2	4
$S_{17}(t,4) \geq$	1.012715	1.161305	1.178614
$S_{17}(t,4) \geq$	1.013558	1.165581	1.188386
$\Sigma_1^{(17)}$	1.220064	2.626354	3.296018
$\Sigma_2^{(17)}$	5759	20950.17	46669.28
$S_{17}(t,4) \leq$	1.016999	1.169541	1.193377

Table 5.16: Computational results for $g = 17$ for the following choice of parameters: $m = 4$, Pmax $= 10^8$,$D_1 = 75$ and $D_2 = 10^4$. This leads to a lower bound of 0.110158 for the lower density of integers representable as the sum of a prime and a power of 17. The time needed to compute this bound was 3h 28m 9s.

t	1	2	4
$S_{18}(t,4) \geq$	1.084399	1.116401	1.141535
$S_{18}(t,4) \geq$	1.085361	1.119558	1.152048
$\Sigma_1^{(18)}$	1.450821	1.965536	2.6474
$\Sigma_2^{(18)}$	6724.59	15970.63	42386.97
$S_{18}(t,4) \leq$	1.088816	1.123534	1.157077

Table 5.17: Computational results for $g = 18$ for the following choice of parameters: $m = 4$, Pmax $= 10^8$,$D_1 = 75$ and $D_2 = 10^4$. This leads to a lower bound of 0.063678 for the lower density of integers representable as the sum of a prime and a power of 18. The time needed to compute this bound was 4h 3m 8s.

6 Appendix - Sage code

```
def getFactorizationInputFull(g,l):
    result=[]
    for i in l:
        result.append((g,i))
    return result

@parallel
def FullFactor(g,n):
    factors = factor(g^n-1);
    result=[]
    for i in factors:
        result.append(i[0])
    return result

def computePrimeProduct(Pmax):
    result=1
    for p in primes(Pmax+1):
        result=result*p
    return result

def getFactorizationInputPartial(primeProduct,g,l):
    result=[]
    for i in l:
        result.append((primeProduct,g,i))
    return result

```

```
@parallel
def PartialFactor(primeProduct,g,n):
    factors = factor(gcd(primeProduct,g^n-1))
    result=[]
    for i in factors:
        result.append(i[0])
    return result

def getNotAllowedPrimeFactors(m,g):
    factors=factor(g^m-1)
    result=[2]
    for i in factors:
        result.append(i[0])
    return result

def getSumUpFValuesInput(l,m,g,factors):
    result=[]
    counter=0
    for i in range(len(l)):
        result.append((counter,l[i],m,g,factors))
        counter=counter+1
    return result

@parallel
def sumUpFValues(dummyParallel,l,m,g,factorList):
    divisorList=l.difference(factorList)
    result=RIF(1)
    for i in divisorList:
        result=result*RIF(1+1/(i-2))
    return result

def getApplyMoebiusInversionInput(l,m):
    result=[]
    for i in range(len(l)):
        result.append((i+1,l,m))
    return result

@parallel
def applyMoebiusInversion(d,l,m):
    result=0
```

```
    for i in divisors(d):
        result=result+moebius(d/i)*l[i-1]
    return result

def getSumUpAfterMoebiusInput(l,m,g):
    result=[]
    for i in divisors(m):
        result.append((i,l,m,g))
    return result

@parallel
def sumUpAfterMoebius(t,l,m,g):
    result1=0
    result2=0
    for i in range(len(l)):
        if gcd(i+1,m).divides(t):
            result1=result1+l[i]*gcd(i+1,t)/(i+1)
            result2=result2+l[i]*gcd(i+1,t)
    return [result1, result2]

def getSumUpAfterMoebiusPartialInput(l,m,g,D1):
    result=[]
    for i in divisors(m):
        result.append((i,l,m,g,D1))
    result.reverse()
    return result

@parallel
def sumUpAfterMoebiusPartial(t,l,m,g,D1):
    result1=RIF(0)
    result2=RIF(0)
    for i in range(D1-1,len(l)):
        if gcd(i+1,m).divides(t):
            result1=result1+l[i]*\\
            RIF(gcd(i+1,t)/(i+1))
            intersum=0
            for j in range(i+1):
                if(gcd(j+1,m)).divides(t):
                    intersum = intersum+\\
                    l[j]*gcd(lcm(i+1,j+1),m)
```

```
            result2=result2+intersum*\\
            RIF(euler_phi(i+1)/(i+1))
    return [result1, result2]

def removeLargePrimeFactorsFullFactorization(Pmax):
        fullFactorizations=load(DATA+\\
        'fullFactorizations')
        result=[]
        for i in fullFactorizations:
            tempSet=set([])
            for j in i:
                if j <= Pmax:
                    tempSet.add(j)
            result.append(tempSet)
        return result

def getResultList(l):
    result=[]
    for i in range(len(l)):
        result.append(set(l[i][1]))
    return result

def getResultListItems(l):
    result=[]
    for i in range(len(l)):
        result.append(l[i][1])
    return result

def getChenSunPrimeBound(L1,L2):
    factors = []
    for l in L1:
        factors.extend(l)
        factors = uniq(factors)
    for l in L2:
        factors.extend(l)
        factors = uniq(factors)
    n = len(factors)
    nthPrime = list(primes(1,n+1))[-1]
    return nthPrime

```

```
def precomputation(g,D1,D2,Pmax):
    fullFactorizationsParallel=sorted\\
    (list(FullFactor(getFactorizationInputFull\\
    (g,range(1,D1+1)))))
    fullFactorizations=getResultList\\
    (fullFactorizationsParallel)
    primeProduct=computePrimeProduct(Pmax)
    partialFactorizationsParallel=sorted(list\\
    (PartialFactor(getFactorizationInputPartial\\
    (primeProduct,g,range(D1+1,D2+1)))))
    partialFactorizations=getResultList\\
    (partialFactorizationsParallel)
    pBound = getChenSunPrimeBound\\
    (fullFactorizations,partialFactorizations)\\
    return [fullFactorizations,\\
    partialFactorizations,pBound]

def computeSumsFullFactorization(m,g,\\
    fullFactorizations):
    notAllowedPrimeFactors=\\
    getNotAllowedPrimeFactors(m,g)
    fullSumListParallel=sorted(sumUpFValues\\
    (getSumUpFValuesInput(fullFactorizations,m,\\
    g,notAllowedPrimeFactors)))
    fullSumList=getResultListItems\\
    (fullSumListParallel)
    resultMoebiusInversionParallel=sorted\\
    (applyMoebiusInversion\\
    (getApplyMoebiusInversionInput\\
    (fullSumList,m)))
    resultMoebiusInversion=getResultListItems\\
    (resultMoebiusInversionParallel)
    sumsParallel=sorted(sumUpAfterMoebius\\
    (getSumUpAfterMoebiusInput\\
    (resultMoebiusInversion,m,g)))
    sums=getResultListItems(sumsParallel)
    return sums

def computeSumsPartialFactorization(m,g,Pmax,D1,\\
    \\fullFactorizations,partialFactorizations):
```

```
    allFactorizations = fullFactorizations
    allFactorizations.extend(partialFactorizations)
    notAllowedPrimeFactors=\\
    getNotAllowedPrimeFactors(m,g)
    partialSumListParallel=sorted(sumUpFValues\\
    (getSumUpFValuesInput(allFactorizations,m,g,\\
    notAllowedPrimeFactors)))
    partialSumList=getResultListItems\\
    (partialSumListParallel)
    resultMoebiusInversionParallel=sorted\\
    (applyMoebiusInversion\\
    (getApplyMoebiusInversionInput\\
    (partialSumList,m)))
    resultMoebiusInversion=getResultListItems\\
    (resultMoebiusInversionParallel)
    sumsParallel=sorted(sumUpAfterMoebiusPartial\\
    (getSumUpAfterMoebiusPartialInput\\
    (resultMoebiusInversion,m,g,D1)))
    sums=getResultListItems(sumsParallel)
    return sums

def getCoprimePairs(g,l):
    o = multiplicative_order(mod(g,l))
    ks = []
    result =[[]]
    for i in range (l):
        result.append([])
        if gcd(i,l)==1:
            ks.append(i)
    for k in ks:
        for i in range(o):
            m=mod(k+g^i,l)
            result[m].append([k,i])
    return result

def computeC3(l,g):
    result = 1
    pDivisors=prime_divisors(l)
    for p in pDivisors:
        if p != 2:
```

```
                result = result*RIF(1+1/(p-2))
    pDivisors=prime_divisors(g)
    for p in pDivisors:
        if p != 2:
            result = result*RIF(1+1/(p-2))
    return result

def getSumPairs(m,l,tList):
    result=RIF(0)
    for i in range(len(l)-1):
        result = result+RIF((1/2)*tList[m])
        for j in range(i+1,len(l)):
            pos = gcd(abs(l[j][1]-l[i][1]),m)
            result = result+RIF(tList[pos])
    result=result+RIF((1/2)*tList[m])
    return 2*result

def applyPintzLemma(S1,S2):
    D=S2/S1
    M=S1
    result=((ceil(D)+floor(D)-D)/(ceil(D)*\\
    floor(D)))*M
    return result

def getDeltas(g,m,l,C1,C2,upperBounds):
    coprimePairs=getCoprimePairs(g,l)
    C3 = computeC3(l,g)
    phiL=euler_phi(l)
    factor1 = RIF(1/(phiL*m*log(g)))
    factor2 = RIF(C1*C2*C3/(phiL*(m^2)*(log(g))^2))
    deltaSum=RIF(0)
    deltaCount=0
    maxt=upperBounds[-1][0]
    tList=[0]
    for i in range(maxt):
        tList.append(0)
    for p in upperBounds:
        tList[p[0]]=p[1]
    for i in range(len(coprimePairs)):
        p=coprimePairs.pop()
```

```
            if p != []:
                if len(p) == 1:
                    deltaSum=deltaSum+RIF\\
                    (l*factor1*len(p))
                    deltaCount=deltaCount+1
                else:
                    S1 = RIF(factor1*len(p))
                    S2Sum=getSumPairs(m,p,tList)
                    S2=S1+RIF(factor2*S2Sum)
                    plResult=applyPintzLemma(S1,S2)
                    deltaSum=deltaSum+l*plResult
                    deltaCount=deltaCount+1
    return [deltaSum,deltaCount]

def getSchoenfeldConstant(pBound):
    if pBound>=2657:
        return RIF(7/2)
    if pBound>=1973:
        return RIF(10/3)
    if pBound>=1429:
        return RIF(3)
    if pBound>=809:
        return RIF(5/2)
    if pBound>=599:
        return RIF(7/3)
    if pBound>=563:
        return RIF(2)
    if pBound>=347:
        return RIF(9/5)
    if pBound>=227:
        return RIF(5/3)
    if pBound>=149:
        return RIF(7/5)
    if pBound>=101:
        return RIF(9/7)
    if pBound>=67:
        return RIF(7/6)
    if pBound>=59:
        return RIF(8/7)
    if pBound>=41:
```

```
            return RIF(1)
        return RIF(4/5)

def getSumRemainderBound(g,D,pBound):
    CSchoenfeld = getSchoenfeldConstant(pBound)
    C = RIF(log(log(g)*(77/200)*\\
    \\(log(pBound)/(log(pBound)-CSchoenfeld))))
    result = RIF((0.922913686/0.66016)*\\
    (2*(1/D+log(D)/D)+C/D))
    return result

def combineSigma1Sigma2(Sigma1,Sigma2,g,Pmax,D1,\\
        D2,t,C):
    result=RIF((D1*log(g))/(2*(Pmax-2)*log(Pmax)))*\\
    Sigma1+RIF(log(g)/((Pmax-2)*log(Pmax)))*Sigma2+\\
    RIF((t*0.922913686)/0.66016)*RIF\\
    ((log(D1)+1+C/2)*(D1*log(g))/\\
    ((Pmax-2)*log(Pmax)))+\\
    RIF((2*t*log(g)*0.922913686)/0.66016)*\\
    RIF((D2*log(D2)+2*D2+(C*D2)/2)/\\
    ((Pmax-2)*log(Pmax)))
    return result

def getSUpperBounds(fullSums,partialSums,\\
        sumRemainderBound,g,Pmax,D1,D2,m,pBound):
    result = []
    CSchoenfeld = getSchoenfeldConstant(pBound)
    C = RIF(log(log(g)*(77/200)*\\
    (log(pBound)/(log(pBound)-CSchoenfeld))))
    mDivisors = divisors(m)
    for i in range(len(fullSums)):
        s = fullSums[i][0]+partialSums[i][0]+\\
        combineSigma1Sigma2(fullSums[i][1],\\
        partialSums[i][1],g,Pmax,D1,D2,mDivisors[i],
        \\C)+sumRemainderBound
        result.append([mDivisors[i],s])
    return result

def printSumTable(fullSums,partialSums,\\
        sumRemainderBound,SUpperBounds,m):
```

```
    divList = divisors(m)
    header="          "
    firstLine="<= D1      "
    secondLine="p<=P+       "
    thirdLine="Sigma1      "
    fourthLine="Sigma2      "
    fifthLine="S <=        "
    for i in range(len(divList)):
        header = header + "            " +\\
        str(divList[i])+"            "
        firstLine = firstLine + \\
        str(fullSums[i][0]) + " "
        secondLine = secondLine + \\
        str(partialSums[i][0]+\\
        fullSums[i][0]) + " "
        thirdLine = thirdLine + \\
        str(fullSums[i][1]) + " "
        fourthLine = fourthLine + \\
        str(partialSums[i][1]) + " "
        fifthLine = fifthLine + \\
        str(SUpperBounds[i][1]) + " "
    print(header)
    print(firstLine)
    print(secondLine)
    print(thirdLine)
    print(fourthLine)
    print(fifthLine)

def computeLowerDensityBound(m,g,D1,D2,Pmax):
    print("Computing a lower bound for the\\
    lower density of integers representable\\
    as the sum of a prime and a power\\
    of "+str(g)+".")
    print("Computing integer factorizations.")
    factorizations = precomputation(g,D1,D2,Pmax)
    print("Computing sums involving full\\
    factorizations.")
    fullSums = computeSumsFullFactorization(m,g,\\
    factorizations[0])
    print("Computing sums involving partial\\
```

```
    factorizations.")
    partialSums = computeSumsPartialFactorization\\
    (m,g,Pmax,D1,factorizations[0],factorizations[1])
    sumRemainderBound = getSumRemainderBound(g,D2,\\
    factorizations[2])
    print("Computing upper bounds for Sg.")
    SUpperBounds = getSUpperBounds(fullSums,\\
    partialSums,sumRemainderBound,g,Pmax,D1,D2,\\
    m,factorizations[2])
    printSumTable(fullSums,partialSums,\\
    sumRemainderBound,SUpperBounds,m)
    deltas = getDeltas(g,m,g^m-1,8,0.6601,\\
    SUpperBounds)
    print("The lower density of integers\\
    representable as the sum of a prime\\
    and a power of "+str(g)+" is bounded\\
    from below by " + str(RIF(deltas[0]/\\
    (g^m-1)))+".")
    return RIF(deltas[0]/(g^m-1))
```

Bibliography

[Brü95] J. Brüdern. *Einführung in die analytische Zahlentheorie*. Springer, Berlin–Heidelberg, 1995.

[CS04] Y.-G. Chen and X.-G. Sun. On Romanoff's constant. *Journal of Number Theory*, 106:275 – 284, 2004.

[Dic57] L. E. Dickson. *Introduction to the theory of numbers*. Dover Publications, Inc., New York, 1957.

[dP49] A. de Polignac. Recherches nouvelles sur les nombres premiers. *Comptes rendus hebdomadaires des sances de l'acadmie des sciences*, 29:397 – 401 and 738–739, 1849.

[Erd50] P. Erdős. On integers of the form $2^k + p$ and some related problems. *Summa Brasiliensis Mathematicae*, 2:113–123, 1950.

[ESP12] C. Elsholtz and J.-C. Schlage-Puchta. On Romanov's constant. *preprint*, 2012.

[Eul] L. Euler. Letter to goldbach, 16.12.1752. URL: `http://eulerarchive.maa.org/correspondence/letters/000879.pdf` [cited 11.10.2013].

[GL03] C. Gugg and A. H. Ledoan. On integers of the form $p + g^k$. Research conducted under the direction of Professors K. Ford and A. Zaharescu at University of Illinois at Urbana-Champaign, Department of Mathematics 2003 REGS program (unpublished), 2003.

[HR06] L. Habsieger and X.-F. Roblot. On integers of the form $p + 2^k$. *Acta Arithmetica*, 122:45–50, 2006.

[HR11] H. Halberstam and H.-E. Richert. *Sieve Methods*. Dover, Mineola, New York, 2011.

[HSF10] L. Habsieger and J. Sivak-Fischler. An effective version of the Bombieri-Vinogradov theorem, and applications to Chen's theorem and to sums of primes and powers of two. *Archiv der Mathematik*, 95(6):557–566, 2010.

[HW08] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers - Sixth Edition*. Oxford University Press, New York, 2008.

[Nat96] M. B. Nathanson. *Additive Number Theory - The Classical Bases*. Springer, New York, 1996.

[Pin06] J. Pintz. A note on romanov's constant. *Acta Mathematica Hungarica*, 112(1–2):1–14, 2006.

[PR03] J. Pintz and I. Z. Rusza. On Linnik's approximation to Goldbach's problem, i. *Acta Arithmetica*, 109.2:169–194, 2003.

[Pra57] K. Prachar. *Primzahlverteilung*. Springer, Berlin – Göttingen – Heidelberg, 1957.

[Rom34] N. P. Romanoff. Über einige Sätze der additiven Zahlentheorie. *Mathematische Annalen*, 109:668–678, 1934.

[Rom83] F. Romani. Computations concerning primes and powers of two. *Calcolo*, 20(3):319–336, 1983.

[Sch76] L. Schoenfeld. Sharper Bounds for the Chebyshev Functions $\theta(x)$ and $\psi(x)$. II. *Mathematics of Computation*, 30:337 – 360, 1976.

[vdC50] J. G. van der Corput. Over het vermoeden van de Polignac. *Simon Stevin*, 27:99–105, 1950.