The three-large-primes variant of the number field sieve

S. Cavallar

REPORT MAS-R0219 AUGUST 31, 2002
CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)
The Three-Large-Primes Variant of the Number Field Sieve

Stefania Cavallar
Technische Universiteit Eindhoven, Postbus 513, 5600 MB Eindhoven, The Netherlands
S.Cavallar@TUE.nl

ABSTRACT
The Number Field Sieve (NFS) is the asymptotically fastest known factoring algorithm for large integers. This method was proposed by John Pollard [20] in 1988. Since then several variants have been implemented with the objective of improving the siever which is the most time consuming part of this method (but fortunately also the easiest to parallelise). Pollard’s original method allowed one large prime. After that the two-large-primes variant led to substantial improvements [11]. In this paper we investigate whether the three-large-primes variant may lead to any further improvement. We present theoretical expectations and experimental results. We assume the reader to be familiar with the NFS.

As a side result, we improved some formulae for Taylor coefficients of Dickman’s ρ function given by Patterson and Rumsey[3] and Marsaglia, Zaman and Marsaglia[16].

2000 Mathematics Subject Classification: Primary 11Y05. Secondary 11A51.
Keywords and Phrases: Number Field Sieve, three large primes, semi-smoothness probabilities, Dickman’s Rho function.

Note: This report was previously published as Chapter 2 of the author’s PhD thesis [8]. The research was carried out under project MAS2.2 “Computational number theory and data security”.

1. Introduction
In [11], B. Dodson and A. K. Lenstra describe their experiments with the two-large-primes method in the Number Field Sieve (NFS) which showed that the turnover point between the one-large-prime method and the two-large-primes method was passed for numbers ranging from 107 to 119 digits. After that, the two-large-primes method soon became widely used for larger number factorisations. In this paper we describe experiments with the three-large-primes variant which was also employed for the special number factorisation record of 233 digits [21]. So far, the experiments do not indicate a distinct advantage over the two-large-primes version, presumably because we still have not reached the turnover point.

For sufficiently large numbers, the relations with three large primes will outnumber the relations with two large primes. But the passage from two to three large primes is not so straightforward as the passage from zero to one or from one to two large primes. Even when the three-large-primes relations outnumber the two-large-primes relations, it can still be too expensive (in time) to find sufficiently many three-large-primes relations in the sieving region. The reason for this is the rareness of successfully factored tri-composites (and with all prime factors below the large prime bound) amongst the many candidate cofactors tested.

Here we enlist a few advantages and disadvantages one expects from the three-large-primes version above the two-large-primes version. We assume that not explicitly mentioned parameters are the same in both methods. When comparing it to the two-large-primes version, one can either keep the same size of the factor base or use a smaller factor base.

If we keep the factor base the same size, we have

**Advantage 1** A smaller sieving region can be taken to produce the same number of relations.
**Disadvantage 1** The time needed to find a useful relation is higher. One reason is the high number of bi-composites among the candidate cofactors tested for tri-compositeness. Another reason is the larger composites which have to be factored.

**Disadvantage 2** For the same number of relations, one can expect more primes to occur in the relations. It will be more difficult to combine the relations with the additional third prime to full relations. As a consequence more sieving will have to be done which might annihilate Advantage 1.

If we keep a smaller factor base, we have

**Advantage 2** Less memory is needed. This is useful to sieve on machines with little memory.

**Advantage 3** For a sufficiently smaller factor base, for the same number of relations, one can expect fewer primes to occur in the relations.

The two example factorisations with three large primes which we treat will show that we have not yet reached the crossover point to the three-large-primes method. For one example with 179 decimal digits \((7^{211} + 1)\) we kept the factor base bound artificially small (Advantage 2) to create the need of three large primes (we sieved \(7^{211} - 1\) with two large primes for comparison), while still not making use of all the three large primes relations since it would have been too costly in time to produce them (compared with the two-large-primes method). In this case we noticed Advantage 3.

Also for the other example, the 233-digit number \(2^{773} + 1\), we handled a rather small factor base bound (Advantage 2) as the number was going to be sieved in parallel on different machines and, for simplicity, we kept the same parameters for all the computers involved. Again, time considerations induced us not to detect all possible three-large-prime relations.

We did not make a comparison between the two-large-prime version and the three-large-prime version with the same factor base. This is left to further research.

In this paper we give methods to predict the number of relations with \(i\) large primes which are found in the sieving part and compare this with real-sieved data for \(i\) at most 5. Most of these comparisons are done for numbers which were sieved with the two-large-primes method (as this is the common method at the moment) and we will see that we can reasonably well estimate the number of relations.

2. **Outline**

In Section 3, we give a description of the sieving step. Here we also introduce most of the notation and terminology needed in later sections.

In Section 4, based on de Bruijn’s \(\Psi\) function, we introduce \(\Psi_i\) to count numbers with \(i\) large primes and show a way to predict the number of smooth polynomial values in the sieving region based on heuristics originating from Peter Montgomery.

In Section 5, we discuss two ways to approximate \(\Psi\) by Dickman’s \(\rho\) function and extend this to \(\Psi_i\) by introducing the functions \(G_i\) and \(H_i\). We thereby generalise work done by Bach and Peralta [3] \((G_1)\) and Lambert [15] \((G_2)\) to three and more large primes.

In Section 6, we generalise (and slightly improve) a theorem by Bach and Peralta from one to two and more large primes: The \(\Delta_i\) defined is an upper bound for how much the \(G_i\) approximation for \(\Psi_i\) is worse than the \(\rho\) approximation of \(\Psi\). We measure that the \(\Delta_i\)s grow with \(i\) but, for \(i \leq 5\) are all below 4% for the range of numbers we are interested in.

In Section 7, we present the numerical methods by Patterson/Rumsey and Marsaglia/Zaman/Marsaglia to compute \(\rho\) and improve upon both methods.

In Section 8 we present actual sieving data for two-large-primes-sieved numbers which we compare to the theoretical estimates.

In Section 9 we describe the obstructions which are encountered when going from two to three large primes in more detail.

In Section 10 we analyse data from a three-large-primes-sieved number with more details.
3. Description of the sieving step

In Section 11 we compare a two-large-primes-sieved number with a three-large-primes-sieved number.

Conclusions are given in Section 11.

3. Description of the sieving step

We only describe the sieving step of the Number Field Sieve. For a complete and detailed description of the Number Field Sieve we refer to [12].

Let $N$ be the number we want to factor. In the NFS two polynomials

$$f_j(x) = c_{j0} + c_{j1}x + \cdots + c_{jd_j}x^{d_j} \in \mathbb{Z}[x], \quad j = 1, 2,$$

are selected which are irreducible over $\mathbb{Z}$ and have a common root modulo $N$. We denote by

$$F_j(x, y) = f_j(x/y)y^{d_j} = c_{j0}y^{d_j} + c_{j1}x y^{d_j-1} + \cdots + c_{jd_j}x^{d_j}$$

the homogeneous form of $f_j(x)$. We call

$$R = [-A, A] \times [1, B] \bigcap \mathbb{Z} \times \mathbb{N} \quad (3.1)$$

the sieving region, where $A$ and $B$ are in $\mathbb{N}$. The siever looks for $(a, b) \in R$ with $a$ and $b$ coprime such that both $F_1(a, b)$ and $F_2(a, b)$ factor completely over the primes below the factor base bounds $B_1$ and $B_2$, respectively, except for at most $k$ and $l$ large primes which should not exceed the so-called large prime bounds $L_1$ and $L_2$, respectively. We call such $(a, b)$ pairs relations. Following [11] we shall denote relations with $k$ large primes in $F_1$ and $l$ large primes in $F_2$ as $k,l$-partial relations whereas relations with no large primes are called full relations. We will call the method an $i$-large-primes variant when allowing $k,l$-partial relations with $\max(k, l) \leq i$.

We allow three large primes for the polynomial which we expect to give the larger values on the sieving region, i.e., the one with larger

$$|c_{j0}B^{d_j}| + |c_{j1}AB^{d_j-1}| + \cdots + |c_{jd_j}A^{d_j}|.$$

Let us assume in this section this is polynomial 2. Thus, our three-large-primes variant allows $2,3$-partial relations.

We sieve the roots of $F_1$ modulo a prime $p$. A triple $(p, q, i)$ denotes $0 < q < p$ such that $F_i(q, 1) \equiv 0 \mod p$. With $(p, \infty, i)$ we denote a projective root $F_i(1, 0) \equiv 0 \mod p$ which occurs for $p \mid c_{id_i}$. These are the two different ways in which $p$ can divide $F_i(a, b) = f_i(a/b)b^k$ with $\gcd(a, b) = 1$, namely $a/b \equiv q \mod p$ for $(p, q, i)$ or $p \mid b$ for $(p, \infty, i)$.

The siever sieves the triples for all the primes $p \in [d, B_1]$ where $d$ is chosen by the user. For these triples, powers are not sieved. For triples with $p < d$ and not dividing the discriminant of the polynomial, the highest power of $p$ below $d$ is sieved. During the sieving process the candidate relations are marked. A relation $(a, b)$ is considered a candidate if

$$|F_1(a, b)| \leq S_1 \prod_{(p, q, i)} p^{\log_2 d} \prod_{p \mid F_1(a, b)} p \prod_{p \mid \text{disc}(f_i)} p \prod_{d \leq p \leq B_1} p L_1^2$$
and
\[ |F_2(a, b)| \leq S_j \prod_{2 \leq p < d} p^{[\log_2 a]} \prod_{p \mid b \text{ and } p \mid F_2(a, b)} p^{-L_2^3}. \]

Here \( S_j, j = 1, 2, \) are user-chosen constants that have to take account of the primes and prime powers which are not sieved. The default value for \( d \) is 31.

The polynomials are sieved one after another. The siever we use is the so-called line-by-line siever. We give a short description for sieving polynomial 1 (2 is done analogously): the siever keeps an array for a fixed \( b \) and the single entries are indexed by \( a \). The entries are initialized with 0. When sieving with \( (p, q, 1) \), we add \( \log p \) to each entry \( a \equiv b q \). For a projective root \( (p, \infty, 1) \), we add \( \log p \) to each entry \( a \) if \( p \mid b \). By storing the logarithms we can add instead of multiply the factors, which results in a time reduction even though we have to sieve afterwards. Here we used the natural logarithm, but in the implementation a more suitable base is chosen. The logarithms will be approximated.

How do we track down the candidates? We recall that \((a, b)\) is a candidate relation if the entry with index \( a \) exceeds \( \log \left( \frac{|F_2(a, b)|}{S_j L_2^2} \right) \) and \( \log \left( \frac{|F_2(a, b)|}{S_j L_2^2} \right) \) after the sieving of polynomial 1 and 2, respectively.

We divide the corresponding polynomial values by the sieved primes determined by resolving above a user-chosen threshold and by trial division below this threshold. After that we check whether the remaining parts are divisible by a higher power of those primes, or by small unsieved primes. The cofactors \( C_1 \) and \( C_2 \) do not contain any primes below \( B_1 \) and \( B_2 \), respectively.

We set a condition on the relationship between \( B_j \) and \( L_j \). This restriction is not essential but enables us to know the maximum possible number of factors of a cofactor. We require \( L_2^3 < B_1^2 \) and \( L_2^3 < B_2^2 \). Note that it follows that \( L_1 < B_1^2, L_2 < B_2^2 \) and \( L_2^3 < B_2^3 \).

A cofactor \( C_2 \) of polynomial 2 is considered only if it falls into one of the three disjoint intervals \( I_{21} = [B_2, L_2], I_{22} = [B_2^2, L_2^2], I_{23} = [B_2^3, L_2^3] \). If \( C_2 \in I_{21} \) we can immediately conclude that the cofactor \( C_2 \) is prime since otherwise \( C_2 \geq B_2^2 > L_2 \). Similarly, if \( C_2 \in I_{22} \), then \( C_2 \) has a maximum of \( i \) primes for \( i = 2, 3 \). See Figure 1.

If a cofactor falls into \( I_{23} \), we first perform a Rabin’s probable prime test and for composites we try to find a factor with Pollard’s \( P - 1 \) method (see section 3.1). If a factor smaller than \( L_2 \) is found and the remaining part belongs to \( I_{22} \) we proceed as for the cofactors falling into \( I_{22} \), namely we do a probable prime test on the cofactor and, if it is composite, factor it with Shanks’s SQUFOF or (if SQUFOF fails) with Pollard Rho. Then we check that the prime factors are in \( I_{21} \).

Since the cofactors which lie centrally in the intervals are the most promising, we restrict the search
to the subintervals

\[ I_{21} = \left[ B_2^{0.1} L_2^{1.9}, B_2^{0.8} L_2^{1.9} \right] \]

(3.2)

These cut-off exponents were chosen based on few experiments. In Table 11 in Section 9 we present some experiments to measure which interval parts are more useful than others. The choice of optimal cut-off exponents however is left for further research.

The processing of \( C_1 \) is done analogously except that \( C_1 \) can only be bi-composite or prime.

The actual factorisation of cofactors for a relation is attempted only after testing size and primality of both cofactors. If there remain two composite cofactors \( C_1 \) and \( C_2 \) they are factored in the following order: the bi-composite candidate \( C_1 \) precedes a tri-composite candidate \( C_2 \). If \( C_2 \) is a bi-composite candidate, the larger of \( C_1 \) and \( C_2 \) is factored first.

### 3.1 The \( P - 1 \) method

The \( P - 1 \) method finds a factor \( p \) of \( n \) if \( p - 1 \) is a product of primes below \( K_1 \) (i.e. \( p - 1 \) is \( K_1 \)-smooth) or if it is a product of primes below \( K_1 \) and one prime between \( K_1 \) and \( K_2 \) (in the terminology of Section 4 this means \( p - 1 \) is \( (1, K_2, K_1) \)-smooth). The algorithm is split into two steps. Step 1 finds the \( p \) for which \( p - 1 \) is \( K_1 \)-smooth, step 2 the \( p \) where \( p - 1 \) is \( (1, K_2, K_1) \)-smooth. In step 1, \( b \equiv 2^M \mod n \) is calculated for \( M \) the product of the prime powers below \( K_1 \). If the gcd \((b - 1, n)\) does not reveal a factor, step 2 is started using Lucas' functions. If \( p - 1 \) is \( (1, K_2, K_1) \)-smooth, we will find gcd\((2^{Mp} + 2^{-Mp} - 2, n) > 1\) for a prime \( p_i \) in \([K_1, K_2]\). The approach with Lucas' functions is less straightforward, but nice properties of these functions allow to check many gcd's at the same time. This speeds up step 2 enormously. On the other hand, fewer gcd checks can also mean that factors are found multiplied together. In that case the algorithm starts over again, but this time with the \( P + 1 \) method (this is repeated a few times, if necessary). The same code for Lucas' functions can be used for this. More gcd checks will be performed in step 1. Note that the \( P + 1 \) method is not invoked if step 2 terminates with a gcd equal to 1. This is better for the average performance (the ratio time per found factor is smaller) as step 1 of \( P + 1 \) is more expensive than step 1 of \( P - 1 \). Only if \( P - 1 \) has found the trivial factor \( n \), we (repeatedly) try \( P + 1 \), hoping that this method will not reduce to the \( P - 1 \) method for all the factors.

This \( P - 1 \) implementation does not attempt a full factorisation when there are three or more factors. Once a partial factorisation is found, SQUFOF or Pollard Rho can finish the factorisation if the cofactor is composite and in range.

The bounds \( K_1 \) and \( K_2 \) are user-chosen. The default values are \( K_1 = 2000 \) and \( K_2 = 50000 \).

This algorithm was developed and implemented by Montgomery [17].

### 4. Counting smooth numbers with \( \Psi \)

De Bruijn's function \( \Psi(x, y) \) denotes the number of positive integers up to \( x \) having no prime factors larger than \( y \). We shall call such integers \( y \)-smooth. Analogously we define, for \( i \) a positive integer and \( x > y > z \), \( \Psi_i(x, y, z) \) to be the number of positive integers up to \( x \) having exactly \( i \) prime factors \( > z \) and \( \leq y \) and the remaining prime factors \( \leq z \). We shall call such integers \((i, y, z)\)-smooth. Note that

\[ \Psi_i(x, y, z) = \sum_{z < p_i \leq y} \sum_{z < p_i \leq y} \cdots \sum_{z < p_i \leq y} \Psi \left( \frac{x}{p_1 \cdots p_i}, z \right). \]

(4.1)

We do not know simple and fast ways to calculate \( \Psi(x, y) \) for large \( x \), so we are going to use an approximation for \( \Psi(x, y) \) (see Section 5).
4.1 Approximation of the number of smooth numbers among polynomial values $F(a, b)$ with \( \gcd(a, b) = 1 \)

For our purposes, we want to approximate the number of smooth values among polynomial values given by a homogeneous polynomial in two variables $F(a, b)$ with $\gcd(a, b) = 1$.

In Subsection 4.2, we will compute the expected contribution to $F(a, b)$ of all primes $p$ smaller than the factor base bound $B$. Therefore, we calculate the average exponent of $p$ in the factorisation of $F(a, b)$; we shall call this $\text{cont}_p(F)$. Before that, we will calculate the corresponding value for random numbers, $\text{cont}_p(r)$. We build on research by Montgomery [18], Boender [4, Chapter 4] and Murphy [19, Chapter 4].

The estimated logarithmic norm after dividing out primes in the factor base is then $\log F(a, b) - \sum_{p \leq B} \text{cont}_p(F) \log p$. The corresponding value for a random number $y$ is $\log y - \sum_{p \leq B} \text{cont}_p(r) \log p$. According to this, we make the following

**Assumption 1.** The polynomial values $F(a, b)$ are about as $B$-smooth as random integers with logarithmic norm $\log F(a, b) + \alpha(F, B)$ where

$$
\alpha(F, B) = \sum_{p \leq B} (\text{cont}_p(r) - \text{cont}_p(F)) \log p.
$$

According to the definition, $\Psi(x, B)/x$ gives the portion of $B$-smooth numbers among the numbers from 1 to $x$. The average size of the numbers is $(1 + x)/2$ which is approximately $x/2$.

The average size of $F(a, b)$ over the sieving region $R$ (usually $R$ is given by (3.1) is

$$
\mathcal{F} = \frac{\iiint_{R} |F(a, b)| \, da \, db}{\iiint_{R} da \, db}.
$$

According to Assumption 1, we can treat the $F(a, b)$ values like random values of average size $x' = F_{x', a(F, B)}$, so we use $\Psi(x', B)/x'$ with $x' = 2\mathcal{F}$ to approximate the portion of $B$-smooth polynomial values among the $(a, b)$ pairs from $R$ with $\gcd(a, b) = 1$.

In the sieving region we have approximately $X = \iiint_{R} \, da \, db \, dx$ pairs such that $\gcd(a, b) = 1$ (see [14, Section 4.5.2]). So we expect $X \Psi_{x}(x, B)$ B-smooth norms among them. Because of (4.1) we can use $X \Psi_{x}(x, B)_{x'}$ as approximations for the number of $(i, L, B)$-smooth norms in the sieving region. If we assume that the two polynomials are independent, then we can use $X \Psi_{x}(x, L, B) \Psi_{x}(x, L, B)$ as an approximation of the number of $i, j$-partial relations. This means that we treat them as if the smoothness of $F_{x}(a, b)$ is unrelated to the smoothness of $F_{x}(a, b)$. By this we neglect some minor effect which can happen for primes which divide the resultant of the two polynomials.

4.2 Calculation of $\text{cont}_p(r)$ and $\text{cont}_p(F)$

In the sequel $k$ shall always denote a positive integer.

**Calculation of $\text{cont}_p(r)$** For random numbers we expect that approximately every $p^k$-th number is divisible by $p^k$, so, taken a random number, we have probability $\frac{1}{p^k} \left( 1 - \frac{1}{p} \right)$ that it is divisible by $p^k$ but not by $p^{k+1}$. Thus, the average exponent for $p$ is equal to $\text{cont}_p(r) = \sum_{k=1}^{\infty} \frac{1}{p^k} \left( 1 - \frac{1}{p} \right) = \frac{1}{p-1}$.

**Calculation of $\text{cont}_p(F)$** Let us first do the calculations for the simpler case of a univariate polynomial $f(x)$ which has $n_{p^k}$ distinct roots modulo $p^k$. Hence a polynomial value is divisible by $p^k$ with probability $n_{p^k}/p^k$. By Heussel’s lemma [10], we know that if we have a non-multiple root modulo $p$, we have a unique corresponding non-multiple root modulo $p^k$. That means that for the primes which do not divide $\text{disc} f$ we have $n_{p^k} = n_p$ for every $k \geq 1$ and we can easily sum the exponents of $p$ to
5. Approximating $\Psi_{\alpha}$

The canonical way to get an approximation for $\Psi$ is by using Dickman's $\rho$ function defined by

$$\rho(x) = 1 \quad \text{for} \quad 0 \leq x \leq 1 \quad \text{and} \quad \rho'(x) = -\frac{\rho(x-1)}{x} \quad \text{for} \quad x > 1. \quad (5.1)$$

By using (1.4) and (5.3) from [6], Bach and Peralta [3] deduced that if $0 < \gamma \leq \alpha < 1$ and $x^\gamma \geq 2$, then

$$\Psi(x, x^{\alpha}) = x \rho \left( \frac{1}{\alpha} \right) + O \left( \frac{x}{\gamma \log x} \right). \quad (5.2)$$

We will add the condition $\log x > \frac{1}{\alpha}$ which is asked for in de Bruijn's (1.4).

We have that

$$G_0(\alpha) := \lim_{x \to \infty} \frac{\Psi(x, x^{\alpha})}{x} = \rho \left( \frac{1}{\alpha} \right).$$
Bach and Peralta [3] proved that, for $0 < \alpha < \beta < 1$

$$G_1(\alpha, \beta) := \lim_{x \to \infty} \frac{\Psi_1(x, x^\beta, x^\alpha)}{x} = \int_0^\beta \rho \left( \frac{1 - \lambda}{\alpha} \right) \frac{d\lambda}{\lambda}.$$ 

Lambert [15] treated the case $i = 2$: for $0 < \alpha < \beta < 1/2$,

$$G_2(\alpha, \beta) := \lim_{x \to \infty} \frac{\Psi_2(x, x^\beta, x^\alpha)}{x} = \frac{1}{2} \int_0^\beta \int_0^\beta \int_0^\beta \rho \left( \frac{1 - (\lambda_1 + \lambda_2)}{\alpha} \right) \frac{d\lambda_1}{\lambda_1} \frac{d\lambda_2}{\lambda_2} d\lambda_3.$$ 

We generalise this further to: for $0 < \alpha < \beta < 1/i$ we have

$$G_i(\alpha, \beta) := \lim_{x \to \infty} \frac{\Psi_i(x, x^\beta, x^\alpha)}{x} = \frac{1}{i!} \int_0^\beta \int_0^\beta \cdots \int_0^\beta \rho \left( \frac{1 - (\lambda_1 + \lambda_2 + \cdots + \lambda_i)}{\alpha} \right) \frac{d\lambda_1}{\lambda_1} \frac{d\lambda_2}{\lambda_2} \cdots \frac{d\lambda_i}{\lambda_i} + O \left( \frac{\log^i((i\alpha)^{-1})}{\alpha(1 - i\beta)} \right). \tag{5.3}$$

In fact, in Section 5.1 we will prove

**Theorem 1.** For a positive integer $i$, $0 < \alpha < \beta < 1/i$ and

$$\log x > \frac{1}{\alpha} \max \left( \log 2, \frac{1 - \alpha}{\alpha} \frac{1}{\log ((i\alpha)^{-1})} \right)$$

we have

$$\Psi_i(x, x^\beta, x^\alpha) = \frac{x}{i!} \int_0^\beta \int_0^\beta \cdots \int_0^\beta \rho \left( \frac{1 - (\lambda_1 + \lambda_2 + \cdots + \lambda_i)}{\alpha} \right) \frac{d\lambda_1}{\lambda_1} \frac{d\lambda_2}{\lambda_2} \cdots \frac{d\lambda_i}{\lambda_i} + O \left( \frac{\log^i((i\alpha)^{-1})}{\alpha(1 - i\beta)} \right).$$

The error bound is uniform in $i$, $\alpha$ and $\beta$.

By $\log^i(x)$ we mean $(\log x)^i$. Since we study $\Psi_i(x, y, z)$ in the form $\Psi_i(x, x^\beta, x^\alpha)$, we have $\alpha = \log z / \log x$ and $\beta = \log y / \log x$.

A more sophisticated approximation for $\Psi(x, y) / x$ than

$$G_0 \left( \frac{\log y}{\log x} \right) = \rho \left( \frac{\log x}{\log y} \right)$$

is given by

$$H_0(x, y) = \rho \left( \frac{\log x}{\log y} \right) + \frac{1 - \gamma}{\log x} \rho \left( \frac{\log x}{\log y} - 1 \right).$$

This approximation was used by Boender [4, Chapter 4] as well as by Murphy [19, Chapter 4] in their approximations. We define the corresponding approximations for $\Psi_i(x, y, z)$, namely

$$H_i(x, z, y) = G_i(\log x, \log x, \log y) = \frac{1 - \gamma}{\log x} \int_{\log x}^{\log y} \frac{dt}{t} \int_{\log x}^{\log t_1} \frac{dt_1}{t_1} \cdots \int_{\log x}^{\log t_i} \frac{dt_i}{t_i} \log z \log (t_1 \cdots t_i) - 1 \right) \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_i}{t_i \log t_i} \tag{5.4}$$
5. Approximating \( \Psi_i \)

with \( \log x \geq \log z + i \log y \).

For completeness we want to mention work by Vershik [23]. He gives formulae for calculating

\[
\Phi_i(\alpha_1, \ldots, \alpha_i) = \lim_{x \to \infty} x^{-1} \{ n \leq x \mid p_k(n) \leq x^{\alpha_k}, k = 1, \ldots, i \}
\]

where \( 1 \geq \alpha_i \geq \cdots \geq \alpha_1 \geq 0 \), and \( p_1(n) \geq \cdots \geq p_i(n) \) are the \( i \) largest prime divisors (counting multiplicity) of the integer \( n \). Vershik’s formulae contain (possibly different) upper bounds and no lower bound for the \( p_i(n) \) values whereas our \( G_i(\alpha, \beta) \) formulae contain one upper and one lower bound for the \( p_i(n) \) values.

5.1 Proof of Theorem 1

Let us first prove some intermediate results we will use several times during the proof. We will use that

\[
\pi(t) = li(t) + \epsilon(t) \tag{5.5}
\]

with \( li(t) = \int_0^t dx / \log x = \lim_{x \to \infty} \left( \int_0^{x^\frac{1}{\alpha}} \frac{dx}{\log x} + \int_{x^\frac{1}{\alpha}}^t \frac{dx}{\log x} \right) \) and

\[
\epsilon(t) = O \left( \frac{t}{\log^c t} \right) \quad \text{for any } c > 0. \tag{5.6}
\]

Lemma 1. For a positive integer \( i \), \( 0 < \alpha < \beta < \frac{1}{i} \), \( \log x > \frac{1}{\alpha \log(1/\alpha)} \) and \( x^\alpha \leq s \leq x^\beta \), we have

\[
\int_{x^\alpha}^s \frac{dt}{\log t} < \log ((i \alpha)^{-1}) \tag{5.7}
\]

\[
\int_{x^\alpha}^s \frac{d\pi(t)}{t} = O \left( \frac{1}{\alpha \log x} \right) \tag{5.8}
\]

\[
\sum_{x^\alpha < p \leq s} \frac{1}{p} = \int_{x^\alpha}^s \frac{d\pi(t)}{t} = O \left( \log ((i \alpha)^{-1}) \right). \tag{5.9}
\]

Proof. For (5.7) we have

\[
\int_{x^\alpha}^s \frac{dt}{\log t} = \log \log t \bigg|_{x^\alpha}^s \leq \log \log x^\beta - \log \log x^\alpha = \log \beta - \log \alpha
\]

\[
< \log \left( \frac{1}{i} \right) - \log \alpha = \log ((i \alpha)^{-1}).
\]

For (5.8) we integrate by parts and use (5.6) with \( c = 1 \) and \( c = 2 \) for the resulting first and second term, respectively,

\[
\int_{x^\alpha}^s \frac{d\pi(t)}{t} = \frac{\epsilon(t)}{t} \bigg|_{x^\alpha}^s + \int_{x^\alpha}^s \frac{\epsilon(t) dt}{t^2} = O \left( \frac{1}{\log t} \right) \bigg|_{x^\alpha}^s + O \left( \int_{x^\alpha}^s \frac{dt}{t \log^2 t} \right) = O \left( \frac{1}{\alpha \log x} \right).
\]

For (5.9) we use Stieltjes integration and find \( \sum_{x^\alpha < p \leq s} \frac{1}{p} = \int_{x^\alpha}^s \frac{d\pi(t)}{t} \). Then we substitute (5.5),

\[
\int_{x^\alpha}^s \frac{d\pi(t)}{t} = \int_{x^\alpha}^s \frac{dt}{t \log t} + \int_{x^\alpha}^s \frac{d\epsilon(t)}{t}.
\]

Thanks to (5.7) and (5.8) we can conclude the proof. \qed

\footnote{or, equivalently, \( \alpha + i \beta \leq 1 \).}
Proof of Theorem 1.

\[
\Psi_i(x, x^\beta, x^\alpha) = \sum_{x^\alpha < p_i \leq x^\beta} \cdots \sum_{x^\alpha < p_i \leq x} \Psi \left( \frac{x}{p_1 \cdots p_i} x^\alpha \right) 
\]

\[
= \sum_{x^\alpha < p_i \leq x^\beta} \cdots \sum_{x^\alpha < p_i \leq x} \Psi \left( \frac{x}{p_1 \cdots p_i} \left( \frac{x}{p_1 \cdots p_i} \right)^{\frac{1}{\log(p_1 \cdots p_i) / \log x}} \right) \tag{5.10}
\]

We use \( \alpha < \frac{1}{\log(p_1 \cdots p_i) / \log x} \) and the theorem hypothesis in order to apply (5.2) to the latter expression. Then we divide by \( x \) which results into

\[
\frac{\Psi_i(x, x^\beta, x^\alpha)}{x} = \sum_{x^\alpha < p_i \leq x^\beta} \cdots \sum_{x^\alpha < p_i \leq x} \frac{1}{p_1 \cdots p_i} \left( \frac{1 - \log(p_1 \cdots p_i) / \log x}{\alpha} \right) + \left( \frac{1}{\alpha} \sum_{x^\alpha < p_i \leq x^\beta} \cdots \sum_{x^\alpha < p_i \leq x} \frac{1}{p_1 \cdots p_i} \right).
\tag{5.11}
\]

By \( \beta < 1/i \) and (5.9), the error term from (5.11) becomes

\[
O \left( \frac{1}{\alpha \log x} \sum_{x^\alpha < p_i \leq x^\beta} \cdots \sum_{x^\alpha < p_i \leq x} \frac{1}{p_1 \cdots p_i} \left( \frac{1 - \log(p_1 \cdots p_i) / \log x}{\alpha} \right) \right) = \left( \frac{1}{\alpha(1-i\beta)} \log^i((\alpha^{-1})^{-1}) \frac{1}{\log x} \right)
\]

and by using Stieltjes integration (5.11) transforms into

\[
\frac{\Psi_i(x, x^\beta, x^\alpha)}{x} = \int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta} \frac{1 - \log(t_1 \cdots t_i) / \log x}{\alpha} \frac{d\pi(t_1)}{t_1} \cdots \frac{d\pi(t_i)}{t_i} + O \left( \frac{\log^i((\alpha^{-1})^{-1})}{\alpha(1-i\beta)} \right) \tag{5.12}
\]

We will prove, for \( x^\beta \geq \cdots \geq t_{k+1} \geq t_k \geq \cdots \geq x^\alpha \), that

\[
\int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta} \frac{1 - \log(t_1 \cdots t_i) / \log x}{\alpha} \frac{d\pi(t_1)}{t_1} \cdots \frac{d\pi(t_i)}{t_i} = \int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta} \frac{1 - \log(t_1 \cdots t_i) / \log x}{\alpha} \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_k}{t_k \log t_k} + O \left( \frac{\log^{k-1}((\alpha^{-1})^{-1})}{\alpha \log x} \right) \tag{5.13}
\]

for \( k \leq i \) by induction on \( k \). We define \( f(t_1, \ldots, t_i) = \frac{1 - \log(t_1 \cdots t_i) / \log x}{\alpha} \).

First we prove (5.13) for \( k = 1 \). We substitute (5.5), use \( \rho(f(t_1, \ldots, t_i)) \leq 1 \) and get

\[
\int_{x^\alpha}^{x^\beta} \rho(f(t_1, \ldots, t_i)) \frac{d\pi(t_1)}{t_1} = \int_{x^\alpha}^{x^\beta} \rho(f(t_1, \ldots, t_i)) \frac{dt_1}{t_1 \log t_1} + \int_{x^\alpha}^{x^\beta} \rho(f(t_1, \ldots, t_i)) \frac{de(t_1)}{t_1} = \int_{x^\alpha}^{x^\beta} \rho(f(t_1, \ldots, t_i)) \frac{dt_1}{t_1 \log t_1} + O(1) \int_{x^\alpha}^{x^\beta} \frac{de(t_1)}{t_1}.
\]
By (5.8) we can conclude the proof for \( k = 1 \). Next, we assume (5.13) is true for \( k < i \). Then, by the induction hypothesis, (5.5) and \( \rho(f(t_1, \ldots, t_i)) \leq 1 \),

\[
\int_{x^\alpha}^{t_{k+2}} \cdots \int_{x^\alpha}^{t_2} \rho(f(t_1, \ldots, t_i)) \frac{d\pi(t_1)}{t_1} \cdots \frac{d\pi(t_{k+1})}{t_{k+1}} = \\
\int_{x^\alpha}^{t_{k+2}} \left( \int_{x^\alpha}^{t_{k+1}} \cdots \int_{x^\alpha}^{t_2} \rho(f(t_1, \ldots, t_i)) \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_{k+1}}{t_{k+1} \log t_{k+1}} \right) + O \left( \frac{\log^{k-1}(i \alpha)^{-1}}{\alpha \log x} \right) \frac{d\pi(t_{k+1})}{t_{k+1}} \\
= \int_{x^\alpha}^{t_{k+2}} \cdots \int_{x^\alpha}^{t_2} \rho(f(t_1, \ldots, t_i)) \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_{k+1}}{t_{k+1} \log t_{k+1}} + O(1) \int_{x^\alpha}^{t_{k+2}} \int_{x^\alpha}^{t_{k+1}} \cdots \int_{x^\alpha}^{t_2} \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_k}{t_k \log t_k} \frac{dt_{k+1}}{t_{k+1} \log t_{k+1}} + O \left( \frac{\log^{k-1}(i \alpha)^{-1}}{\alpha \log x} \right) \int_{x^\alpha}^{t_{k+2}} \frac{d\pi(t_{k+1})}{t_{k+1}}.
\]

We use \( k \) times (5.7) as well as (5.8) and (5.9) and find

\[
O \left( \frac{\log^k((i \alpha)^{-1})}{\alpha \log x} \right)
\]

as error term. This proves (5.13) for \( k + 1 \).

We will use (5.13) with \( k = i \) and substitute \( x^\beta \) for \( t_{k+1} \) (this is possible, since in the last induction step we only used Lemma 1 which is also valid for \( s = x^\beta \)). So the first term on the right hand side of (5.12) transforms into

\[
\int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta} \rho(f(t_1, \ldots, t_i)) \frac{1 - \log(t_1 \cdots t_i)}{\alpha} \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_k}{t_k \log t_k} + O \left( \frac{\log^{i-1}(i \alpha)^{-1}}{\alpha \log x} \right).
\]

The error term in (5.14) is contained in the error term from (5.12) since

\[
\log((i \alpha)^{-1}) \leq \log((i \beta)^{-1}) > 1 - i \beta.
\]

By symmetrising the integral bounds in (5.14) and making the substitution \( \lambda_j = \log t_j / \log x \) we conclude the proof. \( \square \)

6. **Analysis of the \( G_i \) approximations**

With the help of Theorem 2 we will measure how well \( G_i(\alpha, \beta) \) approximates \( \Psi(x, x^\beta, x^\alpha)/x \) under the assumption that \( \rho(1/\alpha) \) is a good approximation for \( \Psi(x, x^\alpha)/x \). To get an idea about the latter, see Hunter and Sorenson [13, Table 2].

**Theorem 2.** Let \( 0 < \alpha < \beta < \frac{1}{4} \) and \( i \) be a positive integer. Assume the Riemann hypothesis. Choose \( c_1 \) and \( c_2 \) so that

\[
c_1 \leq \frac{\Psi(t, \ell^1)}{t \rho(1/\gamma)} \leq c_2
\]

whenever \( \frac{2}{\alpha + 1} \leq \gamma \leq \frac{\alpha^2}{12} \) and \( x^{1-i \beta} \leq t \leq x^{1-i \alpha} \). Then, if \( x^\alpha \geq 2657 \), we have

\[
c_1 (1 - \Delta_{1,i}) \leq \frac{\Psi(x, x^\beta, x^\alpha)}{x G_i(\alpha, \beta)} \leq c_2 (1 + \Delta_{2,i}),
\]
where $\Delta_{j,i} \leq \Lambda_i(\alpha, \beta, x)$ for $j = 1, 2$ and

$$\Lambda_i(\alpha, \beta, x) = \frac{\rho \left( \frac{1-i\beta}{\alpha} \right)}{G_i(\alpha, \beta)} \frac{3\beta \log x}{8\pi x^{\alpha/2}}$$

(6.2)

and, for $i > 1$,

$$\Lambda_i(\alpha, \beta, x) = \frac{\rho \left( \frac{1-i\beta}{\alpha} \right)}{G_i(\alpha, \beta)} \left( \frac{1}{i!} \left( \frac{\log \left( \frac{\beta}{\alpha} \right)}{8\pi x^{\alpha/2}} \right)^i - \log \left( \frac{\beta}{\alpha} \right) \right)$$

$$+ \left( \frac{1}{\alpha x^\alpha \log x} + \frac{\alpha \log x}{2\pi x^\alpha \log x} \right) \left( \log \left( \frac{\beta}{\alpha} \right) + \frac{3\beta \log x}{8\pi x^{\alpha/2}} \right)^{i-2}.$$

(6.3)

This theorem is a generalisation of Bach and Peralta’s [3] Theorem 6.1 handling $i = 1$. We give a slightly better $\Lambda_1$. Table 1 contains values of $\Lambda_i$ for $i \leq 5$ for the numbers (and their parameters) we used in Sections 8, 10 and 11 and, in the last row, for the example from [3]. For the latter, the values of $\Lambda_2$ to $\Lambda_5$ are too large to be useful in contrast with the remaining values which are all below 4%.

In Table 1, $x'_k = 2F_k e^{\alpha x_k}$ with $F_k$ and $\alpha_k(F_k, B_k)$ defined in Section 4.2 and $\alpha'_k = \log x_k$, $B_k$ and $\beta'_k = \log x_k$, $L$ for $k = 1, 2$. The entries “n.a.” (not applicable) indicate that $i > 1/\beta'_k$. The calculation of the entries with “*” took too long and was abandoned.

For the proof of Theorem 2 we need some estimates comparable to Lemma 1. In the following we shall use Schoenfeld’s [22, Corollary 1] bound

$$|e(x)| < \frac{\sqrt{x} \log x}{8\pi}$$

(6.4)

which is valid under the Riemann hypothesis for $x \geq 2657$.

**Lemma 2.** Assume the Riemann hypothesis. For $0 < \alpha < \beta$ and $x^\alpha \geq 2657$, we have

$$\left| \int_0^x \frac{e(t)}{t} \, dt \right| \leq \frac{3\beta \log x}{8\pi x^{\alpha/2}}$$

(6.5)

$$\int_0^x \frac{\pi(t)}{t} \, dt \leq \log \left( \frac{\beta}{\alpha} \right) + \frac{3\beta \log x}{8\pi x^{\alpha/2}}$$

(6.6)

$$\int_0^x \frac{d\pi(t)}{t} \leq \frac{1}{\alpha x^\alpha \log x} + \frac{\alpha \log x}{2\pi x^{3\alpha/2}}$$

(6.7)

**Proof.** For (6.5) we integrate by parts, apply (6.4) and find

$$\left| \int_0^x \frac{e(t)}{t} \, dt \right| \leq \left| \frac{e(t)}{x} \right|_0^x + \left| \int_0^x \frac{e(t)}{t} \, dt \right|$$

$$\leq \left| \frac{e(x)}{x} \right| + \left| \frac{e(x^n)}{x^n} \right| + \frac{1}{8\pi} \int_0^x \frac{\log t}{t^{3/2}} \, dt$$

$$\leq \frac{\beta \log x}{8\pi x^{\alpha/2}} + \frac{\alpha \log x}{8\pi x^{\alpha/2}} + \frac{\beta \log x}{8\pi} \int_0^x \frac{dt}{t^{3/2}}$$

$$\leq \frac{3\beta \log x}{8\pi x^{\alpha/2}}.$$
<table>
<thead>
<tr>
<th>Number</th>
<th>Polyn. (k)</th>
<th>$\alpha'_k$</th>
<th>$\beta'_k$</th>
<th>$x'_k$</th>
<th>$\Lambda_1$</th>
<th>$\Lambda_2$</th>
<th>$\Lambda_3$</th>
<th>$\Lambda_4$</th>
<th>$\Lambda_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.993M</td>
<td>1</td>
<td>0.186</td>
<td>0.205</td>
<td>$8.2 \cdot 10^3$</td>
<td>0.007</td>
<td>0.016</td>
<td>0.025</td>
<td>0.031</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.993M</td>
<td>2</td>
<td>0.203</td>
<td>0.238</td>
<td>$5.2 \cdot 10^2$</td>
<td>0.008</td>
<td>0.018</td>
<td>0.026</td>
<td>0.026</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.999L</td>
<td>1</td>
<td>0.211</td>
<td>0.240</td>
<td>$7.8 \cdot 10^2$</td>
<td>0.007</td>
<td>0.015</td>
<td>0.022</td>
<td>0.023</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.999L</td>
<td>2</td>
<td>0.181</td>
<td>0.205</td>
<td>$4.3 \cdot 10^8$</td>
<td>0.007</td>
<td>0.015</td>
<td>0.024</td>
<td>0.030</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.407+</td>
<td>1</td>
<td>0.169</td>
<td>0.190</td>
<td>$1.1 \cdot 10^2$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.022</td>
<td>0.028</td>
<td>0.026</td>
</tr>
<tr>
<td>3.407+</td>
<td>2</td>
<td>0.211</td>
<td>0.241</td>
<td>$1.6 \cdot 10^3$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.020</td>
<td>0.021</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.413+</td>
<td>1</td>
<td>0.202</td>
<td>0.230</td>
<td>$6.0 \cdot 10^4$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.020</td>
<td>0.021</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.413+</td>
<td>2</td>
<td>0.179</td>
<td>0.201</td>
<td>$6.6 \cdot 10^9$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.022</td>
<td>0.027</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.427+</td>
<td>1</td>
<td>0.200</td>
<td>0.223</td>
<td>$7.2 \cdot 10^{15}$</td>
<td>0.006</td>
<td>0.013</td>
<td>0.020</td>
<td>0.021</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.427+</td>
<td>2</td>
<td>0.179</td>
<td>0.198</td>
<td>$2.7 \cdot 10^6$</td>
<td>0.006</td>
<td>0.013</td>
<td>0.021</td>
<td>0.027</td>
<td>0.027</td>
</tr>
<tr>
<td>3.516+</td>
<td>1</td>
<td>0.204</td>
<td>0.234</td>
<td>$1.1 \cdot 10^4$</td>
<td>0.006</td>
<td>0.015</td>
<td>0.021</td>
<td>0.022</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.516+</td>
<td>2</td>
<td>0.177</td>
<td>0.201</td>
<td>$3.5 \cdot 10^9$</td>
<td>0.006</td>
<td>0.015</td>
<td>0.023</td>
<td>0.029</td>
<td>n.a.</td>
</tr>
<tr>
<td>F857</td>
<td>1</td>
<td>0.166</td>
<td>0.188</td>
<td>$3.2 \cdot 10^2$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.023</td>
<td>0.029</td>
<td>0.026</td>
</tr>
<tr>
<td>F857</td>
<td>2</td>
<td>0.208</td>
<td>0.234</td>
<td>$1.7 \cdot 10^4$</td>
<td>0.006</td>
<td>0.013</td>
<td>0.019</td>
<td>0.021</td>
<td>n.a.</td>
</tr>
<tr>
<td>F949</td>
<td>1</td>
<td>0.188</td>
<td>0.214</td>
<td>$2.5 \cdot 10^7$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.022</td>
<td>–</td>
<td>n.a.</td>
</tr>
<tr>
<td>F949</td>
<td>2</td>
<td>0.169</td>
<td>0.190</td>
<td>$1.3 \cdot 10^2$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.022</td>
<td>0.028</td>
<td>0.026</td>
</tr>
<tr>
<td>3.433+</td>
<td>1</td>
<td>0.252</td>
<td>0.305</td>
<td>$1.8 \cdot 10^6$</td>
<td>0.007</td>
<td>0.015</td>
<td>0.017</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>3.433+</td>
<td>2</td>
<td>0.154</td>
<td>0.170</td>
<td>$8.5 \cdot 10^6$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.023</td>
<td>0.030</td>
<td>0.034</td>
</tr>
<tr>
<td>2.2130M</td>
<td>1</td>
<td>0.244</td>
<td>0.295</td>
<td>$1.3 \cdot 10^2$</td>
<td>0.007</td>
<td>0.015</td>
<td>0.017</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>2.2130M</td>
<td>2</td>
<td>0.152</td>
<td>0.168</td>
<td>$4.5 \cdot 10^2$</td>
<td>0.006</td>
<td>0.014</td>
<td>0.023</td>
<td>0.031</td>
<td>0.035</td>
</tr>
<tr>
<td>2.773+</td>
<td>1</td>
<td>0.161</td>
<td>0.198</td>
<td>$2.4 \cdot 10^4$</td>
<td>0.004</td>
<td>0.009</td>
<td>0.016</td>
<td>0.018</td>
<td>0.013</td>
</tr>
<tr>
<td>2.773+</td>
<td>2</td>
<td>0.162</td>
<td>0.200</td>
<td>$1.1 \cdot 10^5$</td>
<td>0.004</td>
<td>0.009</td>
<td>0.016</td>
<td>0.018</td>
<td>0.013</td>
</tr>
<tr>
<td>7.211−</td>
<td>1</td>
<td>0.169</td>
<td>0.190</td>
<td>$3.0 \cdot 10^2$</td>
<td>0.006</td>
<td>0.013</td>
<td>0.021</td>
<td>0.026</td>
<td>0.024</td>
</tr>
<tr>
<td>7.211−</td>
<td>2</td>
<td>0.200</td>
<td>0.239</td>
<td>$6.6 \cdot 10^3$</td>
<td>0.007</td>
<td>0.015</td>
<td>0.022</td>
<td>0.021</td>
<td>n.a.</td>
</tr>
<tr>
<td>7.211+</td>
<td>1</td>
<td>0.160</td>
<td>0.190</td>
<td>$3.0 \cdot 10^2$</td>
<td>0.007</td>
<td>0.017</td>
<td>0.028</td>
<td>0.034</td>
<td>–</td>
</tr>
<tr>
<td>7.211+</td>
<td>2</td>
<td>0.200</td>
<td>0.239</td>
<td>$6.6 \cdot 10^3$</td>
<td>0.007</td>
<td>0.015</td>
<td>0.022</td>
<td>0.021</td>
<td>n.a.</td>
</tr>
<tr>
<td>B&amp;P</td>
<td>1</td>
<td>0.083</td>
<td>0.133</td>
<td>$1.0 \cdot 10^0$</td>
<td>0.040</td>
<td>0.203</td>
<td>0.687</td>
<td>1.769</td>
<td>3.378</td>
</tr>
</tbody>
</table>

Table 1: $\Lambda_i, i = 1, \ldots, 5$ for some parameter values (see Section 8 for the nomenclature of the numbers).
For (6.6) we substitute (5.5) and use (6.5). For (6.7) we substitute (5.5), integrate by parts, apply (6.4) and use that \(\log x/t^2\) is decreasing in the interval \([x^\alpha, x^\beta]\). We find

\[
\int_{x^\alpha}^{x^\beta} \frac{d\sigma(t)}{t^2} = \int_{x^\alpha}^{x^\beta} \frac{dt}{t^2 \log t} + \frac{e(t)}{t^2} \bigg|_{x^\alpha}^{x^\beta} + 2 \int_{x^\alpha}^{x^\beta} \frac{e(t)}{t^3} dt \\
\leq \frac{1}{\alpha \log x} \int_{x^\alpha}^{x^\beta} \frac{dt}{t^2} + \frac{\log x}{x^{2\beta}} + \frac{\log x}{x^{2\alpha}} + \frac{1}{4\pi} \int_{x^\alpha}^{x^\beta} \frac{\log t}{t^{5/2}} dt \\
\leq \frac{1}{\alpha \log x} \left( \frac{1}{x^\alpha} - \frac{1}{x^\beta} \right) + \frac{\log x}{8\pi x^{3\beta/2}} + \frac{\log x}{8\pi x^{3\alpha/2}} + \frac{\alpha \log x}{4\pi x^{3/2} \int_{x^\alpha}^{x^\beta} dt} \\
\leq \frac{1}{\alpha x^\alpha \log x} + \frac{\alpha \log x}{2\pi x^{3\alpha/2}}.
\]

\[
\square
\]

**Proof of Theorem 2.** We start from equation (5.10) and symmetrise the summation bounds. In the first term on the right hand side of (6.8) we miscount the cases that some of the \(p_1, \ldots, p_i\) are equal. The second term is a large upper bound for the correction.

\[
\Psi_t(x, x^\beta, x^\alpha) \leq \frac{1}{i!} \sum_{x^\alpha \leq p_1 \leq x^\beta} \cdots \sum_{x^\alpha \leq p_i \leq x^\beta} \Psi \left( \frac{x}{p_1 \cdots p_i} \left( \frac{x}{p_1 \cdots p_i} \right) \right) \\
\sum_{x^\alpha \leq p_{i-1} \leq x^\beta} \cdots \sum_{x^\alpha \leq p_i \leq x^\beta} \Psi \left( \frac{x}{p_1 \cdots p_i-1} \left( \frac{x}{p_1 \cdots p_i-1} \right) \right) (6.8)
\]

By using Stieltjes integration and from the assumption (6.1) we get

\[
\frac{\Psi_t(x, x^\beta, x^\alpha)}{x} \leq c_2 \left( \frac{1}{i!} \int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta} \rho \left( \frac{1-\log(t_1 \cdots t_i)/\log x}{\alpha} \right) \frac{d\pi(t_1)}{t_1} \cdots \frac{d\pi(t_i)}{t_i} \right) \\
\int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta} \rho \left( \frac{1-\log(t_{i-1} \cdots t_i-2)/\log x}{\alpha} \right) \frac{d\pi(t_1)}{t_1} \cdots \frac{d\pi(t_{i-2})}{t_{i-2}} \frac{d\pi(t_{i-1})}{t_{i-1}} (6.9)
\]

We substitute (5.5) for the first part and for the second part we bound

\[
\rho \left( \frac{1-\log(t_1 \cdots t_i)/\log x}{\alpha} \right)
\]

by \(\rho \left( \frac{1-i\beta}{\alpha} \right)\), apply (6.7) as well as \(i-2\) times (6.6) and get

\[
\frac{\Psi_t(x, x^\beta, x^\alpha)}{x} \leq c_2 \left( G_i(\alpha, \beta) + \frac{i}{i!} \sum_{j=1}^{i} \binom{i}{j} E_{i-j,j} + \rho \left( \frac{1-i\beta}{\alpha} \right) \frac{1}{\alpha x^\alpha \log x} \right) \sum_{j=0}^{i-2} \binom{i-2}{j} \log^{i-2-j} \left( \frac{\beta}{\alpha} \right) \frac{3\beta \log x}{8\pi x^{3\beta/2}} (6.10)
\]
where
\[ E_{i-j,j} = \int_{x^n}^{x^*} \int_{a^*}^{a^n} \rho \left( \frac{1 - \log(t_1 \cdots t_j)}{\alpha} \right) \frac{dt_1}{t_1} \cdots \frac{dt_{i-j}}{t_{i-j} \log t_{i-j}} \frac{de(t_{i-j+1})}{t_{i-j+1} \log t_{i-j+1}} \cdots \frac{de(t_i)}{t_i}. \] (6.11)

In order to find an upper bound for \( E_{i-j,j} \) we bound \( \rho \) in (6.11) by \( \rho \left( \frac{1-i\beta}{\alpha} \right) \), apply \( i - j \) times
\[ \int_{x^n}^{x^*} \frac{dt}{t \log t} = \log \left( \frac{2}{x} \right) \] and \( j \) times (6.5) and obtain
\[ E_{i-j,j} \leq \rho \left( \frac{1-i\beta}{\alpha} \right) \log^{i-j} \left( \frac{\beta}{\alpha} \right) \left( \frac{3\beta \log x}{8x^2 \alpha^2} \right)^j. \] (6.12)

We use (6.12) to bound the right hand side of (6.10). We divide both sides of the new inequality by \( G_i(\alpha, \beta) \) to get
\[ \frac{\Psi_i(x, x^\beta, x^{\beta^2})}{x G_i(\alpha, \beta)} \leq c_2 (1 + \Lambda_i) \]
with \( \Lambda_i \) given by (6.2) and (6.3), respectively.

The lower bound is proven by an entirely analogous argument, starting from the left hand side inequality in (6.1). Actually, the proof is simpler for the left hand side since we do not need a correction term as given by the second term on the right hand side of (6.9).

7. Calculating \( \rho \)

We want to calculate Dickman’s \( \rho \) function to high precision, as our estimates of smooth numbers rely on values of the \( \rho \) function.

The \( \rho \) function is the solution of a so-called differential-difference equation. This implies that it is piecewise analytic. For example
\[ \rho(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 1 - \log x & \text{if } 1 \leq x \leq 2 \end{cases} \]

For all the other intervals of length 1 we can write \( \rho \) as a Taylor series where the coefficients depend on the Taylor series coefficients of \( \rho \) in the left adjacent interval. In order to guarantee correct results up to a certain precision, we use two methods. One was used first by Bach and Peralta [3] and is due to Patterson and Ramsey; it expands the series on the right end of the intervals. The other method is by Marsaglia, Zaman and Marsaglia [6] and expands the series around the midpoints of the intervals. Although M&Z&M give nearly correct values for \( \rho(10) \), \( \rho(15) \) and \( \rho(20) \), there are a few oversights in their formulae. We provide the correct relations in Table 2.

Bach and Peralta found that for computing \( \rho(x) \) in the range \( 0 \leq x \leq 20 \) with a relative error of about \( 10^{-17} \), it is sufficient to approximate the infinite sums in P&R’s method with the sums of the first 55 terms. Table 3 reproduces the number of terms and the working precision required to guarantee 16, 32 or 64 correct digits of \( \rho \) in the range \( 0 \leq x \leq 20 \) for each of the two methods. The calculations were done with MATHEMATICA’s arbitrary precision: MATHEMATICA maintains as much precision as possible and, if necessary, performs internal intermediate calculations to up to 50 more digits. If it loses precision because of roundoff errors, only the correct digits are returned. In Table 3 we also reproduce the time needed to calculate all the \( c_i \)’s on an SGI O2 MIPS R5000 180 MHz.

\[ \text{One might also think of expanding the series on the left end of the intervals, i.e. writing } \rho(k + \xi) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i \text{ for } 0 \leq \xi \leq 1. \text{ The formulae can} \]
\[ \text{be got analogously and would be even simpler as in the other two cases, but the method is impractical as too many terms are needed due to the very slow convergence of the sums. For example, } c_i^{(k)} = \frac{(-1)^i}{i!}. \]
Patterson & Rumsey

\[0 \leq \xi \leq 1\]
\[\rho(k - \xi) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i\]
\[k = 1, 2, \ldots\]

\[c_0^{(1)} = 1\]
\[c_i^{(1)} = 0 \text{ for } i > 0\]
\[c_0^{(2)} = 1 - \log 2\]
\[c_i^{(2)} = \frac{1}{i^2} \text{ for } i > 0\]

\[c_i^{(k)} = \frac{1}{k} \sum_{j=0}^{i-1} \frac{c_j^{(k-1)}}{j+1} \text{ for } i > 0\]
\[c_0^{(k)} = \frac{1}{k} \sum_{j=1}^{\infty} \frac{c_j^{(k-1)}}{j+1}\]

for \(k > 1\)

Marsaglia & Zaman & Marsaglia

\[-1 \leq \xi \leq 1\]
\[\rho \left( k + \frac{1}{2} + \frac{1}{2} \xi \right) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i\]
\[k = 0, 1, \ldots\]

\[c_0^{(0)} = 1\]
\[c_i^{(0)} = 0 \text{ for } i > 0\]
\[c_0^{(1)} = 1 - \log \left( \frac{3}{2} \right)\]
\[c_i^{(1)} = \frac{(-1)^i}{i!} \text{ for } i > 0\]

\[c_i^{(k)} = \frac{c_i^{(k-1)} + c_k^{(k-1)}}{i(k+1)} \text{ for } i > 0\]
\[c_0^{(k)} = \sum_{j=0}^{\infty} \left( c_j^{(k-1)} + \frac{(-1)^j + m_j^{(k-1)}}{j+1} \right)\]
\[m_0^{(k)} = \frac{c_0^{(k)}}{2k+3}\]
\[m_i^{(k)} = \frac{c_i^{(k)} - m_i^{(k-1)}}{2k+3} \text{ for } i > 0\]

for \(k > 0\)

Table 2: Taylor coefficients for \(\rho\) - original methods

In order to determine how many terms and how many digits of working precision were needed, we first chose sufficiently high values for both methods to have the results \(\rho(i)\) for \(i = 1, \ldots, 20\) coincide to a high number of digits for the two methods. We took that as a reference solution. Then we varied MATHEMATICA’s working precision to find the lower bound for the working precision and after that we determined the minimum number of terms.

<table>
<thead>
<tr>
<th>order rel. error</th>
<th>Patterson &amp; Rumsey</th>
<th>Marsaglia &amp; Zaman &amp; Marsaglia</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>number of terms</td>
<td>working time in s</td>
</tr>
<tr>
<td>(10^{-17})</td>
<td>42</td>
<td>17</td>
</tr>
<tr>
<td>(10^{-33})</td>
<td>91</td>
<td>32</td>
</tr>
<tr>
<td>(10^{-65})</td>
<td>195</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 3: Original methods

Both methods have some drawback. P&R have to calculate a sum of \(i\) terms for each \(c_i^{(k)} (i > 0, k > 2)\) which becomes costly in time when we need high precision and thus many terms. M&Z&M on the other hand have a very involved way to calculate \(c_0^{(k)} (k > 1)\). Their treatment requires many more digits of precision than P&R.

We were able to simplify both P&R’s \(c_i^{(k)} (i > 0, k > 2)\) and M&Z&M’s \(c_0^{(k)} (k > 1)\) so they resemble the corresponding \(c\) in the other method. Therefore we only needed to follow the approach taken in the other method. For the derivation of \(c_0^{(k)}\) this means to use \(\rho(x) = \frac{1}{x} \int_{x-1}^{x} \rho(t) dt\) for \(x > 1\) instead of the integral form of (5.1) which was used by M&Z&M. For M&Z&M, this avoids the need of developing
\( \frac{\zeta(k+1/2, \zeta(2))}{2^{k+1} \pi^k} \) into power series which leads to the auxiliary coefficients \( m_i \). For the derivation of \( c_i^{(k)} (i > 0) \) we used (5.1). Then, for P&R, we did NOT develop \( 1/(1 - \zeta/k) \) as a power series as described in [2]. For the new proofs, see Appendix A. In Table 4 we give the simplified terms together with the similar term in the other method. The recursive formula for P&R’s \( c_i^{(k)} (i > 1, k > 2) \) can be derived from

\[
\begin{align*}
c_i^{(k)} &= \frac{c_{i-1}^{(k-1)} + i (i-1) c_i^{(k-1)}}{i k} \quad i > 0 \\
c_0^{(k)} &= \frac{1}{i k} \sum_{j=1}^{\infty} \frac{c_j^{(k-1)}}{j+1} \\
&= \frac{1}{i k} \left( (i-1)c_i^{(k-1)} + c_i^{(k-1)} \right) 
\end{align*}
\]

With the new recursion formula P&R has become 6 to 44 times faster. The simplified M&Z&M needs 20 fewer digits of precision and between 56 and 58 fewer terms to produce the same relative error as the original version. Moreover, it became 2 to 3 times faster. See Table 3 and 5 for some numerical data.

<table>
<thead>
<tr>
<th>order rel. error</th>
<th>Patterson &amp; Runsey</th>
<th>Marsaglia &amp; Zaman &amp; Marsaglia</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>number of terms</td>
<td>working precision in s</td>
</tr>
<tr>
<td>10^{-17}</td>
<td>42</td>
<td>17</td>
</tr>
<tr>
<td>10^{-33}</td>
<td>91</td>
<td>32</td>
</tr>
<tr>
<td>10^{-65}</td>
<td>195</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 4: New form of Taylor coefficients for \( \rho \) compared to the corresponding term in the other method

Also easily be derived from

\[
c_i^{(k)} = \frac{1}{i k} \sum_{j=0}^{i-1} \frac{c_j^{(k-1)}}{k^{i-j}} = \frac{1}{i k} \sum_{j=0}^{i-1} \frac{c_j^{(k-1)}}{k^{i-j}} = \frac{1}{i k} \left( \sum_{j=0}^{i-2} \frac{c_j^{(k-1)}}{k^{i-j}} + c_i^{(k-1)} \right)
\]

A further slight improvement in time can be achieved by also using the recursive forms for P&R’s \( c_i^{(2)} \) and M&Z&M’s \( c_i^{(1)} (i > 0) \).

Note that in order not to bump into MATHEMATICA’s recursion limit one should give P&R’s \( c_i^{(k)} (k > 2) \) and M&Z&M’s \( c_i^{(k)} (k > 1) \) explicitly.

8. Examples

In this section we compare real sieve data with the approximations we derived in 4.1 for several example numbers. We tried to have some variety in our examples by including Cunningham numbers as well as Fibonacci numbers. All the numbers were factored by Montgomery at the time of our experiments. In Tables 7 and 8 we reproduce the parameters used in the factorisations.

With \( x, y+ \) we denote \( x^y + 1 \). With 2, 2hM we denote the Aurifeuillian factor \( [5, \text{III, C.2}] \frac{2^h + 2^{h+1}}{2} + 1 \) of 2, 2h+. Similarly, 3, 3hM is short for \( 3^k + 3^{k+1} + 1 \) and 3, 3hL for \( 3^k - 3^{k+1} + 1 \). The numbers \( Fx \)
<table>
<thead>
<tr>
<th>Polynomial</th>
<th>$f_1(x)$</th>
<th>$f_2(x)$</th>
<th>$\text{cont}_3(F_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.993M</td>
<td>$f_1(x) = 3^{55}x - 1$</td>
<td>$f_2(x) = x^6 + 3x^3 + 3$</td>
<td>$\frac{1}{t}$</td>
</tr>
<tr>
<td>3.999L</td>
<td>$f_1(x) = 3^{55}x - 1$</td>
<td>$f_2(x) = x^6 - 9x^3 + 27$</td>
<td>$\frac{3}{4}$</td>
</tr>
<tr>
<td>3.407+</td>
<td>$f_1(x) = 3^{37}x - 3^{34} - 1$</td>
<td>$f_2(x) = x^5 - x^4 - 4x^3 + 3x^2 + 3x - 1$</td>
<td>$\frac{1}{7t}$</td>
</tr>
<tr>
<td>3.413+</td>
<td>$f_1(x) = x - 3^{30}$</td>
<td>$f_2(x) = x^6 - x^5 + x^4 - x^3 + x^2 - x + 1$</td>
<td>$\frac{1}{8}$</td>
</tr>
<tr>
<td>3.427+</td>
<td>$f_1(x) = x - 3^{34}$</td>
<td>$f_2(x) = x^6 - x^5 + x^4 - x^3 + x^2 - x + 1$</td>
<td>$\frac{1}{7t}$</td>
</tr>
<tr>
<td>3.516+</td>
<td>$f_1(x) = 3^{57}x - 1$</td>
<td>$f_2(x) = x^6 + 3x^3 + 9$</td>
<td>$\frac{1}{t}$</td>
</tr>
<tr>
<td>F857</td>
<td>$f_1(x) = F171x - F172$</td>
<td>$f_2(x) = x^5 + 5x^4 + 10x^2 - 5x + 2$</td>
<td>$\frac{1}{7t}$</td>
</tr>
<tr>
<td>F949</td>
<td>$f_1(x) = x - L146$</td>
<td>$f_2(x) = x^6 - x^5 - 5x^4 + 4x^3 + 6x^2 - 3x - 1$</td>
<td>$\frac{1}{7t}$</td>
</tr>
<tr>
<td>3.433+</td>
<td>$f_1(x) = x - 1018022109428884191058$</td>
<td>$f_2(x) = 582157800x^5 - 13767381653260x^4 - 3504111252981476x^3 + 5033731003610092975x^2 + 41414643218036780062x - 563572130841392284366681$</td>
<td>$\frac{1}{7t}$</td>
</tr>
<tr>
<td>2.2130M</td>
<td>$f_1(x) = x - 53100003123331135610192$</td>
<td>$f_2(x) = 6590263680x^5 - 710580832922906x^4 + 10126751094225398x^3 + 3498676764197537945x^2 - 5404582433335517396810x + 2581400262310033907312415$</td>
<td>$\frac{1}{801012008}$</td>
</tr>
</tbody>
</table>

Table 6: Detailed polynomials data
<table>
<thead>
<tr>
<th></th>
<th>( \text{3,993M} )</th>
<th>( \text{3,999L} )</th>
<th>( \text{3,407+} )</th>
<th>( \text{3,413+} )</th>
<th>( \text{3,427+} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNFS difficulty</td>
<td>158</td>
<td>159</td>
<td>177</td>
<td>169</td>
<td>175</td>
</tr>
<tr>
<td>cofactor size</td>
<td>144</td>
<td>149</td>
<td>148</td>
<td>135</td>
<td>169</td>
</tr>
<tr>
<td>degree ( f_1(x) )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>degree ( f_2(x) )</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>( A )</td>
<td>1680000</td>
<td>2520000</td>
<td>3600000</td>
<td>3360000</td>
<td>4000000</td>
</tr>
<tr>
<td>( B )</td>
<td>1560000</td>
<td>1250000</td>
<td>3000000</td>
<td>2400000</td>
<td>3200000</td>
</tr>
<tr>
<td>( X )</td>
<td>( 3.18651 \cdot 10^{12} )</td>
<td>( 3.82999 \cdot 10^{12} )</td>
<td>( 1.31312 \cdot 10^{13} )</td>
<td>( 9.80474 \cdot 10^{12} )</td>
<td>( 1.63411 \cdot 10^{13} )</td>
</tr>
<tr>
<td>( B_1 )</td>
<td>4400000</td>
<td>8500000</td>
<td>1300000</td>
<td>1100000</td>
<td>1450000</td>
</tr>
<tr>
<td>( B_2 )</td>
<td>1100000</td>
<td>1000000</td>
<td>1000000</td>
<td>1300000</td>
<td>1700000</td>
</tr>
<tr>
<td>( L )</td>
<td>6000000</td>
<td>8000000</td>
<td>1000000</td>
<td>1000000</td>
<td>1000000</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>15</td>
<td>30</td>
<td>40</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>60</td>
<td>30</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>( 2.93074 \cdot 10^{22} )</td>
<td>( 4.39611 \cdot 10^{22} )</td>
<td>( 6.08267 \cdot 10^{21} )</td>
<td>( 3.39129 \cdot 10^{24} )</td>
<td>( 4.06955 \cdot 10^{25} )</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>( 1.87775 \cdot 10^{27} )</td>
<td>( 1.02598 \cdot 10^{28} )</td>
<td>( 1.53030 \cdot 10^{27} )</td>
<td>( 6.13546 \cdot 10^{28} )</td>
<td>( 2.54655 \cdot 10^{29} )</td>
</tr>
<tr>
<td>( \alpha(F_1, B_1) )</td>
<td>0.569915</td>
<td>0.569915</td>
<td>0.569915</td>
<td>0.569915</td>
<td>0.569915</td>
</tr>
<tr>
<td>( \alpha(F_2, B_2) )</td>
<td>1.468072</td>
<td>1.429203</td>
<td>2.319329</td>
<td>2.378699</td>
<td>2.377064</td>
</tr>
<tr>
<td>full relations</td>
<td>297961/0.56/0.60</td>
<td>412555/0.54/0.62</td>
<td>385672/0.63/0.73</td>
<td>502027/0.53/0.61</td>
<td>684987/0.55/0.63</td>
</tr>
<tr>
<td>0,1-partial rels.</td>
<td>481365/0.61/0.69</td>
<td>873553/0.58/0.66</td>
<td>737783/0.67/0.76</td>
<td>1047129/0.58/0.65</td>
<td>1205720/0.59/0.67</td>
</tr>
<tr>
<td>0,2-partial rels.</td>
<td>268380/0.68/0.76</td>
<td>633695/0.66/0.73</td>
<td>446398/0.74/0.81</td>
<td>759311/0.64/0.72</td>
<td>741788/0.66/0.73</td>
</tr>
<tr>
<td>1,0-partial rels.</td>
<td>769170/0.57/0.65</td>
<td>806649/0.55/0.62</td>
<td>944266/0.64/0.72</td>
<td>1008690/0.54/0.61</td>
<td>1194986/0.56/0.63</td>
</tr>
<tr>
<td>1,1-partial rels.</td>
<td>1248973/0.62/0.69</td>
<td>1711506/0.59/0.66</td>
<td>1799413/0.68/0.75</td>
<td>2116479/0.59/0.65</td>
<td>2107447/0.60//0.66</td>
</tr>
<tr>
<td>1,2-partial rels.</td>
<td>694993/0.70/0.76</td>
<td>1245009/0.67/0.73</td>
<td>1085377/0.74/0.81</td>
<td>1532260/0.66/0.71</td>
<td>1299863/0.67/0.72</td>
</tr>
<tr>
<td>2,0-partial rels.</td>
<td>627188/0.61/0.68</td>
<td>500656/0.58/0.64</td>
<td>819125/0.66/0.74</td>
<td>655488/0.58/0.64</td>
<td>686676/0.59/0.65</td>
</tr>
<tr>
<td>2,1-partial rels.</td>
<td>1018741/0.66/0.72</td>
<td>1061956/0.62/0.68</td>
<td>1565368/0.70/0.77</td>
<td>1374882/0.62/0.68</td>
<td>1217910/0.63//0.68</td>
</tr>
<tr>
<td>2,2-partial rels.</td>
<td>568849/0.74/0.79</td>
<td>780025/0.70/0.74</td>
<td>946628/0.77/0.82</td>
<td>1003843/0.69/0.74</td>
<td>750130/0.70/0.75</td>
</tr>
<tr>
<td>total relations</td>
<td>5975620/0.64/0.71</td>
<td>8028843/0.61/0.68</td>
<td>8732030/0.69/0.77</td>
<td>1000010/0.61/0.67</td>
<td>9891300/0.61/0.68</td>
</tr>
<tr>
<td>( 1.6\mathrm{h}(L) )</td>
<td>5700294</td>
<td>7472145</td>
<td>9219535</td>
<td>9219535</td>
<td>9219535</td>
</tr>
<tr>
<td>sieving time (days)</td>
<td>148</td>
<td>131</td>
<td>98</td>
<td>59</td>
<td>112</td>
</tr>
</tbody>
</table>

Table 7: Examples
<table>
<thead>
<tr>
<th>name</th>
<th>3,516+</th>
<th>F857</th>
<th>F949</th>
<th>3,433+</th>
<th>2,2130M</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNFS difficulty</td>
<td>165</td>
<td>179</td>
<td>184</td>
<td>n.a</td>
<td>n.a</td>
</tr>
<tr>
<td>cofactor size</td>
<td>161</td>
<td>179</td>
<td>157</td>
<td>115</td>
<td>118</td>
</tr>
<tr>
<td>degree f₁(x)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>degree f₂(x)</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>A</td>
<td>3000000</td>
<td>6000000</td>
<td>8400000</td>
<td>70200000</td>
<td>97200000</td>
</tr>
<tr>
<td>B</td>
<td>1600000</td>
<td>3050000</td>
<td>4400000</td>
<td>1000000</td>
<td>1350000</td>
</tr>
<tr>
<td>X</td>
<td>7.58694 \times 10^{12}</td>
<td>2.22502 \times 10^{13}</td>
<td>4.49380 \times 10^{13}</td>
<td>8.53532 \times 10^{12}</td>
<td>1.59545 \times 10^{13}</td>
</tr>
<tr>
<td>B₁</td>
<td>850000</td>
<td>1100000</td>
<td>1100000</td>
<td>4200000</td>
<td>4200000</td>
</tr>
<tr>
<td>B₂</td>
<td>10000000</td>
<td>13000000</td>
<td>13000000</td>
<td>16777215</td>
<td>16777215</td>
</tr>
<tr>
<td>L</td>
<td>90000000</td>
<td>100000000</td>
<td>100000000</td>
<td>100000000</td>
<td>100000000</td>
</tr>
<tr>
<td>S₁</td>
<td>20</td>
<td>30</td>
<td>30</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>S₂</td>
<td>30</td>
<td>25</td>
<td>1</td>
<td>100</td>
<td>2000</td>
</tr>
<tr>
<td>x₁</td>
<td>6.12316 \times 10^{23}</td>
<td>1.79418 \times 10^{22}</td>
<td>1.43103 \times 10^{27}</td>
<td>1.01802 \times 10^{26}</td>
<td>7.16972 \times 10^{26}</td>
</tr>
<tr>
<td>x₂</td>
<td>1.04850 \times 10^{29}</td>
<td>6.25192 \times 10^{23}</td>
<td>5.73419 \times 10^{49}</td>
<td>4.87448 \times 10^{48}</td>
<td>1.65831 \times 10^{40}</td>
</tr>
<tr>
<td>α(F₁, B₁)</td>
<td>0.569915</td>
<td>0.569915</td>
<td>0.569915</td>
<td>0.569915</td>
<td>0.569915</td>
</tr>
<tr>
<td>α(F₂, B₂)</td>
<td>1.193893</td>
<td>1.002230</td>
<td>3.153286</td>
<td>-4.046483</td>
<td>-5.915719</td>
</tr>
<tr>
<td>full relations</td>
<td>408537/0.47/0.54</td>
<td>393688/0.59/0.68</td>
<td>359222/0.46/0.53</td>
<td>446527/0.44/0.51</td>
<td>364736/0.54/0.62</td>
</tr>
<tr>
<td>0, 1-partial rels.</td>
<td>935790/0.52/0.59</td>
<td>652752/0.64/0.72</td>
<td>802483/0.51/0.58</td>
<td>963530/0.49/0.56</td>
<td>812613/0.60/0.68</td>
</tr>
<tr>
<td>0, 2-partial rels.</td>
<td>742778/0.60/0.66</td>
<td>336153/0.74/0.82</td>
<td>636000/0.58/0.65</td>
<td>593371/0.74/0.82</td>
<td>621128/0.74/0.82</td>
</tr>
<tr>
<td>1, 0-partial rels.</td>
<td>889385/0.48/0.54</td>
<td>1095953/0.60/0.68</td>
<td>808649/0.47/0.53</td>
<td>1014837/0.44/0.50</td>
<td>865394/0.55/0.62</td>
</tr>
<tr>
<td>1, 1-partial rels.</td>
<td>2049612/0.53/0.59</td>
<td>1817042/0.64/0.72</td>
<td>1817656/0.52/0.57</td>
<td>2189528/0.50/0.55</td>
<td>1930024/0.61/0.67</td>
</tr>
<tr>
<td>1, 2-partial rels.</td>
<td>1628450/0.61/0.66</td>
<td>937005/0.74/0.81</td>
<td>1442146/0.59/0.65</td>
<td>1338927/0.75/0.81</td>
<td>1471358/0.75/0.82</td>
</tr>
<tr>
<td>2, 0-partial rels.</td>
<td>623474/0.51/0.56</td>
<td>1071958/0.63/0.70</td>
<td>610361/0.50/0.55</td>
<td>622358/0.46/0.50</td>
<td>574197/0.57/0.63</td>
</tr>
<tr>
<td>2, 1-partial rels.</td>
<td>1441725/0.56/0.61</td>
<td>1779908/0.68/0.74</td>
<td>1374875/0.55/0.60</td>
<td>1345966/0.51/0.55</td>
<td>1279510/0.63/0.68</td>
</tr>
<tr>
<td>2, 2-partial rels.</td>
<td>1148798/0.64/0.68</td>
<td>916616/0.79/0.84</td>
<td>1094971/0.62/0.67</td>
<td>813549/0.78/0.82</td>
<td>972034/0.78/0.83</td>
</tr>
<tr>
<td>total relations</td>
<td>986562/0.56/0.61</td>
<td>9001145/0.67/0.74</td>
<td>8947023/0.54/0.60</td>
<td>9328593/0.56/0.62</td>
<td>8899994/0.65/0.72</td>
</tr>
<tr>
<td>L6</td>
<td>L(L)</td>
<td>8348496</td>
<td>9219535</td>
<td>9219535</td>
<td>9219535</td>
</tr>
<tr>
<td>sieving time (days)</td>
<td>90</td>
<td>215</td>
<td>281</td>
<td>68</td>
<td>162</td>
</tr>
</tbody>
</table>

Table 8: Examples (continued)
are Fibonacci numbers, the numbers $L_x$ Lucas numbers.

The numbers were sieved with the Special Number Field Sieve (SNFS), except for $2,2130M$ and $3,433+$, which were sieved with the General Number Field Sieve (GNFS). Also for the latter two we write the numbers to be factored as algebraic factors, even though we are actually factoring a cofactor. The cofactor sizes are stated in Tables 7 and 8. The SNFS difficulty is given by the resultant of the polynomials.

In Table 6 we give the polynomials used for the sieving and the cont_p's for primes dividing the resultant of the polynomials.

In Tables 7 and 8, the triple entries $a/b/c$ for the $i,j$-partial relations contain

\[ a := \text{number } r_{i,j} \text{ of } i,j \text{-partial relations}, \]
\[ b := \frac{X \cdot G_i(a_i', \beta_i') \cdot G_j(a_j', \beta_j')}{r_{i,j}} \]
\[ c := \frac{X \cdot H_i(x_i', B_1, L) \cdot H_j(x_j', B_2, L)}{r_{i,j}} \]

with $x_k = 2^{\tau_k}$ (see 4.2) and $x_k' = x_k \cdot e^{\alpha(F_k, B_k)}$, $\alpha_k' = \log_{x_k'} B_k$, $\beta_k' = \log_{x_k'} L$ for $k = 1, 2$. The time unit is a day.

We also state the value of $1.6\ell_i(L)$ which is a heuristic estimate by Montgomery (private communication) of the number of total relations needed when sieving with large prime bound $L$. All the examples were tuned with simulations to yield approximately that number of total relations.

Examples $3,413+$ and $3,427+$ used the same higher-degree polynomial.

The polynomials for the two GNFS examples were chosen to have many factors modulo small primes. This is reflected by the negative $a$ for the high degree polynomials.

The estimates with $G_i$ vary from 44% to 79%. The $H_i$ estimates are from 4% to 10% higher than the $G_i$ estimates. The estimates tend to be lower for full relations than for partial relations with many large primes. The estimated number of total relations varies from 54% to 66% and 60% to 77% for $G$ and $H$, respectively.

If one likes to know whether certain parameters yield enough data with the two-large-primes sieve without sieving, we suggest to tune the parameters to yield approximately $0.6 \cdot 1.6 \ell_i(L)$ or $0.7 \cdot 1.6 \ell_i(L)$ estimated total relations, respectively for $G$ and $H$.

The sieving time seems to be hardly correlated with the numbers or parameters. This may be due to the use of different machines.

9. Obstructions when going from two to three large primes

In this section we use $B$ and $L$ without indices meaning the factor base bound and the large prime bound of the polynomial allowing three large primes. In the sequel, by candidate bi- or tri-composite we mean cofactors in $[B^3, L^3]$ and $[B^3, L^3]$, respectively. A candidate bi-composite cofactor can be either bi-composite or prime, a candidate tri-composite can be tri-composite, prime or bi-composite (see Figure 1 in Section 3).

We distinguish two types of bi-composites: either both primes are below the large prime bound $L$ or one prime exceeds $L$. We discard the bi-composites of the second type. All bi-composite cofactors between $B^3$ and $L^3$ have at least one factor exceeding $L$ (since we assumed $L^2 < B^3$), so we will discard those. Similarly among the tri-composites, we keep the ones with all three primes below $L$ and discard the ones with at least one factor larger than $L$. The major obstruction when switching from two to three large primes is that only a small fraction of the composite candidate tri-composites really will be useful tri-composites, while most of them will be useless bi-composites.

Filtering out the primes is easy, as probable prime tests can be performed in time orders of magnitude smaller than what factoring takes. Unfortunately, distinguishing between bi- and tri-composites is not so quick on average. A known but not very fast method is to trial-divide primes starting from $B$ to the cubic root of the number to be factored. The massive presence of bi-composites
among the composite candidate tri-composites has a big impact on the average sieving time per useful relation, since a lot of effort is put into the factoring of cofactors which are not useful.

We are interested in a factorisation method which detects tri-composites quickly and gives up on factoring bi-composites in $[B^3, L^3]$ early. We found that Pollard's $P - 1$ method is well-suited giving a good yield for numbers of the size of our candidate tri-composites.

The method finds factors $p$ where $p - 1$ has all factors below a given limit $K_1$ and possibly one factor between $K_1$ and a second limit $K_2$. For a description of the $P - 1$ implementation, see Section 3.1. We are interested in small limits, as this means quitting the factorisation of bi-composites in $[B^3, L^3]$ early. On the other hand, we want large enough limits to guarantee that a high percentage of useful tri-composites will be found. Actually, only one factor needs to be found. If it is possible to find a fraction $f$ of all the prime factors of the useful tri-composites, we estimate that a fraction $1 - (1 - f)^3$ of the useful tri-composites can be identified. We computed these fractions and the actual numbers of factored tri-composites for a series of 292 useful tri-composites in the interval $[10^{21}, 10^{27}]$ with factor base $10^7$ and large prime bound $10^8$ and different $P - 1$ limits. Some results are reported in Table 9. We do not give the time for factoring the tri-composites here, as this is negligible compared with the time for factoring the bad bi-composites. Good limits can be investigated with a few simulation runs of the siever. The default values chosen for the implementation of $P - 1$ are 2000 and 50000. For these values and for $B = 10^7$ and $L = 10^8$ only 50% of the factors of the useful tri-composites were found, but this accounts for the partial factoring of 87% of the useful tri-composites.

Table 10 compares some three-large-primes sieved runs for different $P - 1$ bounds. We used the number $2^{773} + 1$ (see Section 10). We sieved a sublattice of the sieving region used, namely the points $(a, 9073 \cdot b)$ with $a$ an integer in $[-28,875,000, 28,875,000]$ and $b = 1, \ldots, 2200$. The first polynomial was allowed to have three large primes. The factor base bound was $B = B_1 = 2 \cdot 10^7$, the large prime bound $L = 10^8$.

A total of 1765 748 candidate tri-composites were marked, 947 992 of which resulted prime. After checking that the cofactor from the other polynomial value is okay (a bi-composite candidate there will be factored before attempting the factorisation of the tri-composite), a total of 60 531 tri-composite

<table>
<thead>
<tr>
<th>$K_1$</th>
<th>$K_2$</th>
<th>#factors found</th>
<th>$f$</th>
<th>$1 - (1 - f)^3$</th>
<th>#tri-composites found</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>10000</td>
<td>247</td>
<td>0.28</td>
<td>0.63</td>
<td>183</td>
<td>0.63</td>
</tr>
<tr>
<td>500</td>
<td>20000</td>
<td>299</td>
<td>0.34</td>
<td>0.71</td>
<td>205</td>
<td>0.70</td>
</tr>
<tr>
<td>1000</td>
<td>20000</td>
<td>332</td>
<td>0.38</td>
<td>0.76</td>
<td>221</td>
<td>0.76</td>
</tr>
<tr>
<td>1000</td>
<td>50000</td>
<td>404</td>
<td>0.46</td>
<td>0.84</td>
<td>244</td>
<td>0.84</td>
</tr>
<tr>
<td>1200</td>
<td>60000</td>
<td>429</td>
<td>0.49</td>
<td>0.87</td>
<td>252</td>
<td>0.86</td>
</tr>
<tr>
<td>2000</td>
<td>50000</td>
<td>434</td>
<td>0.50</td>
<td>0.87</td>
<td>255</td>
<td>0.87</td>
</tr>
<tr>
<td>2000</td>
<td>100000</td>
<td>488</td>
<td>0.56</td>
<td>0.91</td>
<td>265</td>
<td>0.91</td>
</tr>
<tr>
<td>10000</td>
<td>100000</td>
<td>509</td>
<td>0.58</td>
<td>0.93</td>
<td>267</td>
<td>0.91</td>
</tr>
<tr>
<td>2000</td>
<td>200000</td>
<td>540</td>
<td>0.62</td>
<td>0.94</td>
<td>277</td>
<td>0.95</td>
</tr>
<tr>
<td>5000</td>
<td>250000</td>
<td>568</td>
<td>0.65</td>
<td>0.96</td>
<td>280</td>
<td>0.96</td>
</tr>
<tr>
<td>10000</td>
<td>500000</td>
<td>621</td>
<td>0.71</td>
<td>0.98</td>
<td>281</td>
<td>0.96</td>
</tr>
<tr>
<td>12000</td>
<td>600000</td>
<td>630</td>
<td>0.72</td>
<td>0.98</td>
<td>281</td>
<td>0.96</td>
</tr>
<tr>
<td>20000</td>
<td>100000</td>
<td>676</td>
<td>0.77</td>
<td>0.99</td>
<td>285</td>
<td>0.98</td>
</tr>
<tr>
<td>50000</td>
<td>100000</td>
<td>676</td>
<td>0.77</td>
<td>0.99</td>
<td>285</td>
<td>0.98</td>
</tr>
<tr>
<td>20000</td>
<td>150000</td>
<td>702</td>
<td>0.80</td>
<td>0.99</td>
<td>288</td>
<td>0.99</td>
</tr>
<tr>
<td>50000</td>
<td>150000</td>
<td>702</td>
<td>0.80</td>
<td>0.99</td>
<td>288</td>
<td>0.99</td>
</tr>
<tr>
<td>60000</td>
<td>300000</td>
<td>730</td>
<td>0.83</td>
<td>1.00</td>
<td>290</td>
<td>0.99</td>
</tr>
<tr>
<td>60000</td>
<td>300000</td>
<td>737</td>
<td>0.84</td>
<td>1.00</td>
<td>291</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 9: Percentages found tri-composites for some $P - 1$ limits
candidates were tried to be factored by $P - 1$. In the following listing we describe what the first 6 columns in Table 10 mean:

$K_1$ and $K_2$. These are the limits for the $P - 1$ method.

**not factored** This gives the number of composites for which the $P - 1$ method could not find a factor.

**factor too large** Here either the factor found by $P - 1$ is too large or the remaining cofactor (prime or composite—not tested here) is too large. This count also includes tri-composites where the second found factor (by SQUFOF or Pollard Rho) or the corresponding cofactor is too large.

**cofactor prime** After finding the first factor, the size of the factor and the cofactor is checked (this is covered by the previous column). If okay, a probable prime test is performed on the cofactor. This column gives the number of probable prime cofactors. A prime cofactor corresponds to a bi-composite with a too large factor (because of $L^2 < B^3$).

**three factors** All three factors are smaller than $L$. These are the wanted tri-composites.

The last two columns of Table 10 give the total number of relations found and the average time (in seconds) to find such a relation on a Silicon Graphics Origin 2000 MIPS R12000 300MHz. The lowest time on this list is with $P - 1$ bounds 1 200 and 60000. These are also the bounds used in the factorisation of $2^{73} + 1$ (see Section 10).

We can see that for $P - 1$ bounds 1 200 and 60000, more than 10 times as many useless bi-composites than useful tri-composites were found. Note that the factor 10 is a rough lower bound for the ratio between bi- and tri-composites, as most of the numbers falling into column 3 and 4 are also bi-composites. The time per relation augments for larger $P - 1$ bounds as more time-expensive bi-composites get factored and the number of tri-composites saturates. With $K_1 = 1$ 200 and $K_2 = 60000$, in average a $P - 1$ run on a composite candidate tri-composite was about 61 times the time of a probable prime test on a candidate tri-composite.

In all the sieving experiments we reduced the intervals for bi-composites $[B_i^2, L_i^2]$, for $i = 1, 2$, and for tri-composites $[B_i^3, L_i^3]$, for $i$ equal to 1 or 2, according to (3.2). This cuts down the sieving time.
as the search in the central parts is more effective. This can be seen in Table 11. The number sieved is $2^{773} + 1$ on the same sublattice as above with $K_1 = 1200$ and $K_2 = 600000$. In this table, every line gives data considering only candidate tri-composites in $[B^6 L^{3.5-v}, B^{-0.5} L^{3.5-v}]$ while keeping the whole interval for the bi-composites, $[B^2 L^2]$. The interval $[B^{2.5} L^{0.8}, B^{1.1} L^{1.6}]$ from (3.2) is contained in $[B^{2.5} L^{0.8}, B^{1.0} L^{2.0}]$, which data is given by line 2 to line 4 of Table 11 and which have better rates. The time is for a SGI Origin 2000 MIPS R1200 300MHz processor.

10. AN EXAMPLE WITH THREE LARGE PRIMES
In this section, we test how well we can approximate the number of relations, especially the ones with three large primes. We consider a simplified case. Instead of sieving both polynomials simultaneously, we sieve them separately.

We use the special number $2,773+ = 2^{773} + 1$ for this experiment. This 233-digit number was factored [21] in October, 2000 by the NFS using a linear and a degree-6 polynomial. The factor base bounds where $B_1 = B_2 = 2 \cdot 10^9$ and the large prime bound $L = 10^9$. On the linear side three large primes were allowed, on the other side two large primes. The sieving region (3.1) had $A = 28875 000$ and $B = 22 000 000$. Further, we chose $S_1 = 0.1$ and $S_2 = 6.0$. The $P-1$ bounds (see Section 3.1) were set to 1 200 and 60 000.

We sieved values for the linear homogeneous polynomial $F_1(a, b) = a - 2^{129}b$ and the degree-6 polynomial $F_2(a, b) = a^6 + 2b^6$ separately.

The discriminant of $f_2(x) = x^6 + 2$ is divisible by 2 and 3, so we calculate $\text{cont}_2(F_2)$ and $\text{cont}_3(F_2)$ manually. Modulo 2 the polynomial is $x^6$ which has 0 as the only multiple root which means $n_2 = 1$. The polynomial has no roots modulo 2$^2$, so $n_{22} = n_{220} = \cdots = 0$. It follows that $\text{cont}_2(F_2)$ consists only of the term (4.3) with $k = 1$, so $\text{cont}_2(F_2) = \frac{2}{27} = \frac{1}{27}$. In an analogous way we find that $\text{cont}_3(F_2) = \frac{2^{23}}{3^7} = \frac{1}{3}$. The correction values are $\alpha(F_1, B_1) = 0.5 699 015$ and $\alpha(F_2, B_2) = 1.938 592$.

For this example, we sieved over a small part of the sieving region, namely all integer pairs $(a, b) \in [-A, A] \times [1000000, 1000100]$ with $\text{gcd}(a, b) = 1$.

This corresponds to about $X = 2A \sum_{1000000}^{1000100} \frac{\mu(a)}{a^7} \approx 3.5277 2 \cdot 10^9$ candidate pairs. The mean value of

\[ \left| X \right| \approx 3.52772 \cdot 10^9 \]
polynomial $F_1$ over this region is

$$F_1 = \frac{1}{2AB} \int_{-A}^{A} \int_{-B}^{B} \left| F_1(a, b) \right| da \, db \approx 6.73793 \cdot 10^{44},$$

whereas $F_2 \approx 8 \cdot 197267 \cdot 10^{43}$. We put $x_i = 2 \cdot F_1$ for $i = 1, 2$. We assume (see Assumption 1) we have got to do with random numbers of maximal size $x'_1 = x_1 e^{a(F_1, B)} \approx 2.38 \cdot 10^{45}$ and $x'_2 = x_2 e^{a(F_2, B)} \approx 1.14 \cdot 10^{45}$.

Table 12 gives the results for the linear polynomial. The real siever did not find all good $(a, b)$ pairs. Most of those missed are with three large primes, but also a few with two large primes were missed. Some were discarded because the unsieved part does not belong to one of the intervals given in (3.2). Others were missed because no factor can be found by the $P - 1$ method according to the chosen bounds (see 3.1). Another reason for missing a pair is that more small primes appear in the factorisation than anticipated with the choices of $S_1$ and $S_2$. Therefore we also ran a special (expensive) sieve which found all smooth numbers so that we can better compare with the theoretical expectations. The numbers of relations from the real siever are reported in column 3 of Table 12 whereas column 2 gives the results from the ideal siever. The estimates outnumber the number of relations from the ideal siever. The values of the fractions $XG_i/R_i$ decrease when $i$ increases. The same happens with the corresponding fraction with $H_i$ instead of $G_i$.

$$\begin{array}{|c|c|c|c|c|c|c|}
\hline
i & R_i & x_1 & XH_i/R_i & x'_1 & XH'_i/R_i \\
\hline
0 & 36214 & 1.11 & 1.19 & 1.00 & 1.07 \\
1 & 201002 & 1.09 & 1.16 & 1.00 & 1.05 \\
2 & 400217 & 1.08 & 1.12 & 1.00 & 1.03 \\
3 & 347230 & 1.07 & 1.09 & 1.00 & 1.02 \\
4 & 122983 & 1.04 & 1.05 & 0.99 & 1.00 \\
5 & 118260 & 1.00 & n.a. & 1.00 & n.a. \\
\hline
\end{array}$$

Table 12: Numbers of smooth values of the linear polynomial $F_1(a, b)$ with $(a, b)$ satisfying (10.1)

Table 13 gives the data for the degree-6 polynomial. Only the ideal sieve data is given, but no substantial difference with the real data should be expected here as this polynomial was only sieved with two large primes and only a small part of the pairs with two large primes get discarded. Here, as in the examples from Section 7, we can see again that the approximations get better when more large primes are allowed.

$$\begin{array}{|c|c|c|c|c|c|c|}
\hline
i & R_i & x_2 & XH_i/R_i & x'_2 & XH'_i/R_i \\
\hline
0 & 120758 & 0.48 & 0.52 & 0.34 & 0.37 \\
1 & 577746 & 0.54 & 0.57 & 0.39 & 0.41 \\
2 & 959430 & 0.61 & 0.64 & 0.46 & 0.48 \\
3 & 663086 & 0.71 & 0.73 & 0.57 & 0.58 \\
4 & 170016 & 0.88 & 0.88 & 0.76 & 0.77 \\
5 & 10232 & n.a. & n.a. & 1.15 & n.a. \\
\hline
\end{array}$$

Table 13: Numbers of smooth values of the degree-6 polynomial $F_2(a, b)$ with $(a, b)$ satisfying (10.1)

The approximations are within 20% for the linear polynomial, but rather poor for the higher-degree polynomial (up to 66% off). This is because the linear polynomial is near-constant over all of (10.1)
while the degree-six polynomial grows from $2 \cdot 10^{39}$ to $5 \cdot 10^{44}$. To get better results for the higher-degree polynomial we should split up the sieving region in smaller pieces and do the approximations on the smaller pieces. This is left for further research.

We gave the results from the approximations by using the real size of the numbers ($x$) as well as the size when comparing to random numbers ($x'$). For the linear case, the latter gives better results, for the other polynomial it is exactly the other way around. However, in both cases the estimates with $x'$ are lower than the ones with $x$, due to the positive $\alpha$.

10.1 Approximation for the number of smooth numbers in an interval

We investigate how our formulae work for numbers in an interval instead of for polynomial values. Let us choose the interval

$$ [-X/2 + 2^{129} \cdot 1000 \cdot 100, X/2 + 2^{129} \cdot 1000 \cdot 100], $$

(10.2)

which treats the same number $X$ of candidates as in the previous section. The numbers in this interval are larger than $x = 2^{72}$ but still of the same order of magnitude.

The approximations $xG_i(x, B_1, log_x L)$ and $xH_i(x, B_1, L)$ actually approximate the portion of smooth numbers between 1 and $x$, so, for the special case of intervals, we will define $G_i^{\text{int}}$ and $H_i^{\text{int}}$ and use $XG_i^{\text{int}}(log_x B_1, log_x L)$ and $XH_i^{\text{int}}(x, B_1, L)$ with $x$ being some element in the interval.

Let us construct $G_i^{\text{int}}$ and $H_i^{\text{int}}$. As an estimate for smooth values in an interval $[x_1, x_2]$ we approximate $\Psi_i(x, y, z) - \Psi_i(x_1, y, z)$ by the derivative of the approximation (5.3)

$$ G_i^{\text{int}}(x, z, y) = \frac{d}{dx} xG_i(\log x, z, \log x y) = $$

$$ G_i(\log x, z, \log x y) = \frac{1}{i!} \int_0^y \cdots \int_z^y \frac{\rho \left( \log x - \log(t_1 \cdots t_i) \right)}{\log x - \log(t_1 \cdots t_i) \log t_1 \cdots \log t_i} \frac{dt_1}{t_1 \log t_i} \cdots \frac{dt_i}{t_i \log t_i} \quad (10.3) $$
times $x_r - x_1$ for $log x \geq log z + i log y$, or the derivative of the approximation (5.4)

$$ H_i^{\text{int}}(x, z, y) = \frac{d}{dx} xH_i(x, z, y) = H_i(x, z, y) - $$

$$ \frac{1}{i!} \int_0^y \cdots \int_z^y \frac{\rho \left( \log x - \log(t_1 \cdots t_i) \right)}{\log x - \log(t_1 \cdots t_i) \log t_1 \cdots \log t_i} \frac{dt_1}{t_1 \log t_i} \cdots \frac{dt_i}{t_i \log t_i} \quad $$

$$ \frac{1 - \gamma}{\log x} \frac{1}{i!} \int_0^y \int_z^y \left( \frac{\rho \left( \log x - \log(t_1 \cdots t_i) \right)}{\log x} - 1 \right) \frac{dt_1}{t_1 \log t_i} \cdots \frac{dt_i}{t_i \log t_i} \quad (10.4) $$
times $x_r - x_1$ for $log x \geq 2 log z + i log y$.

Again, the estimates are within 20% from the real data. This is comparable with the results for linear polynomials (Table 12).

11. Comparing the two- and the three-large-primes method

The numbers $7, 211 - = 2^{11} - 1$ and $7, 211 + = 2^{11} + 1$ differ by 2 and are therefore suited for comparison purposes. We sieved $7, 211 -$ while allowing two large primes on both polynomials whereas for $7, 211 +$ we allowed up to three large primes on the linear side, i.e., polynomial 1.

---

*In our examples the interval bounds $x_r - x_1 \ll x_1$, so it does not matter which $x \in [x_1, x_r]$ we use.*
11. Comparing the two- and the three-large-primes method

<table>
<thead>
<tr>
<th>$i$</th>
<th>$R_i$</th>
<th>$XG_i/R_i$</th>
<th>$XH_i/R_i$</th>
<th>$XG_i^{\text{int}}/R_i$</th>
<th>$XH_i^{\text{int}}/R_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>40920</td>
<td>1.11</td>
<td>1.19</td>
<td>0.91</td>
<td>0.98</td>
</tr>
<tr>
<td>1</td>
<td>233495</td>
<td>1.10</td>
<td>1.16</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>2</td>
<td>439114</td>
<td>1.09</td>
<td>1.13</td>
<td>0.93</td>
<td>0.97</td>
</tr>
<tr>
<td>3</td>
<td>374335</td>
<td>1.07</td>
<td>1.10</td>
<td>0.95</td>
<td>0.97</td>
</tr>
<tr>
<td>4</td>
<td>128293</td>
<td>1.05</td>
<td>1.05</td>
<td>1.01</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Table 14: Numbers of smooth numbers in interval (10.2) with $x = 2^{129} \cdot 1000000$, $z = B_1$ and $y = L$

We did not take advantage of already known factors

\[
7^{211} - 1 = 2 \cdot 3 \cdot 141793 \cdot c173
\]

\[
7^{211} + 1 = 2^3 \cdot 25571219
\]

by taking for $7, 211 \pm$ the polynomials $f_1(x) = 7^{42}x - 1$ and $f_2(x) = x^5 \pm 7$ with root $7^{-42}$ modulo $c128$ and $c173$, respectively.\(^6\) So both numbers have the same SNFS difficulty. For both numbers we have $\text{cont}_2(F_2) = \frac{1}{3}$ and $\text{cont}_7(F_2) = \frac{1}{3}$. Both numbers had the same sieving region and used identical large prime bounds, but $7, 211 -$ had a large factor base with the linear side while $7, 211 +$ allowed three primes there.

There is a minor secondary effect of the known factors. Since, for example, $7^{211} + 1$ is divisible by 8, the polynomial $f_1 = 7^{42}x - 1$ and $f_2(x) = x^5 + 7$ share a root $x \equiv 1 \mod 8$, increasing the likelihood that both are simultaneously smooth.

Sieving simulations indicated that sieving with three large primes would be more costly in time, see the value for the estimated time per relation in Table 15. For the $P - 1$ method we used the default bounds $K_1 = 2000$ and $K_2 = 50000$ 3.1.

In Table 15, a ‘$i,j$-partial rels.’ entry gives the number of sieved $i,j$-partial relations in the first column as well as the estimates $X \cdot G_i(\log_{B_1} B_1, \log_{L} L) \cdot G_j(\log_{B_2} B_2, \log_{L} L)$ and $X \cdot H_i(x'_1, B_1, L) \cdot H_j(x'_2, B_2, L)$ in the second and third column, respectively. The values $x'_1$ and $\alpha(F_k, B_k)$, $k = 1, 2$, are defined in Section 4.1. In the total relations entry we give the percentage of the real relations instead of the estimates themselves.

The detailed real sieving data for $7, 211 -$ have unfortunately been lost but we expect their ratios to the estimates to be comparable with those for $7, 211 +$. The theoretical estimates for relations with fewer than three large primes for $7, 211 +$ vary between 61% and 86% of the real number of relations.

For the three-large-primes relations the estimates outnumber the real numbers, since (because of time considerations) the siever discards many three-large-primes candidates.

The real number of relations for $7, 211 +$ is smaller than expected from the sieving simulations. A 2% deviation could be expected, but in fact it is 5%.

We cannot use the total sieving time as an indicator for performance, since $7, 211 -$ was sieved exclusively on low-memory machines, which tend to be slower.

We wanted to analyse which set of relations would give the better (smaller and lighter) matrix when considering the same number of relations. To this end, we truncated each data set to 11.4 million non-duplicate relations. After that we filtered (see [7]) with mrgleve 1, filtmin 1M and keep 200K and later with mrgleve 8, filtmin 500K, maxrela 13.0 and maxdscrd 40K.

The matrix of $7, 211 +$ is lighter and slightly smaller than the matrix of $7, 211 -$ as there are fewer relations with more than 2 linear primes larger than 6 million (the $B_1$ for $7, 211 +$ in the $7, 211 +$ matrix. If all the relations had been considered for the filtering this would probably have led to a smaller matrix for $7, 211 -$.

\(^6\) We inverted the root as it is more more convenient to have $A > B$ with line sieving.
<table>
<thead>
<tr>
<th>method (# large primes)</th>
<th>2+2</th>
<th>3+2</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>7, 211−</td>
<td>7, 211+</td>
</tr>
<tr>
<td>SNFS difficulty</td>
<td>179</td>
<td>179</td>
</tr>
<tr>
<td>cofactor size</td>
<td>173</td>
<td>128</td>
</tr>
<tr>
<td>degree $f_1(x)$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>degree $f_2(x)$</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>A</td>
<td>540000</td>
<td>540000</td>
</tr>
<tr>
<td>B</td>
<td>350000</td>
<td>350000</td>
</tr>
<tr>
<td>X</td>
<td>$2.29796 \times 10^{13}$</td>
<td>$2.29796 \times 10^{13}$</td>
</tr>
<tr>
<td>$B_1$</td>
<td>1500000</td>
<td>600000</td>
</tr>
<tr>
<td>$B_2$</td>
<td>600000</td>
<td>600000</td>
</tr>
<tr>
<td>L</td>
<td>12000000</td>
<td>12000000</td>
</tr>
<tr>
<td>$S_1$</td>
<td>300</td>
<td>1</td>
</tr>
<tr>
<td>$S_2$</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$1.68466 \times 10^{42}$</td>
<td>$1.68466 \times 10^{42}$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$2.36785 \times 10^{33}$</td>
<td>$2.36785 \times 10^{33}$</td>
</tr>
<tr>
<td>$\alpha (F_1, B_1)$</td>
<td>0.369915</td>
<td>0.369915</td>
</tr>
<tr>
<td>$\alpha (F_2, B_2)$</td>
<td>1.027386</td>
<td>1.027386</td>
</tr>
<tr>
<td>full relations</td>
<td>-210279/24333</td>
<td>19560/73240/85466</td>
</tr>
<tr>
<td>0,1-partial relations</td>
<td>-646312/731630</td>
<td>34260/22560/2256941</td>
</tr>
<tr>
<td>0,2-partial relations</td>
<td>-654225/725637</td>
<td>295549/228372/225836</td>
</tr>
<tr>
<td>1,0-partial relations</td>
<td>-524955/596529</td>
<td>525659/326026/272459</td>
</tr>
<tr>
<td>1,1-partial relations</td>
<td>-1613408/1793884</td>
<td>1006575/1000207/1119873</td>
</tr>
<tr>
<td>1,2-partial relations</td>
<td>-313264/1791890</td>
<td>1303147/1304341/1111069</td>
</tr>
<tr>
<td>2,0-partial relations</td>
<td>-480504/336882</td>
<td>831856/330064/394762</td>
</tr>
<tr>
<td>2,1-partial relations</td>
<td>-1476872/1414244</td>
<td>2394438/1629201/1788273</td>
</tr>
<tr>
<td>2,2-partial relations</td>
<td>-1494954/1601021</td>
<td>2077068/1649149/1773624</td>
</tr>
<tr>
<td>3,0-partial relations</td>
<td>n.a.</td>
<td>318335/385258/425221</td>
</tr>
<tr>
<td>3,1-partial relations</td>
<td>n.a.</td>
<td>917002/1181412/1278512</td>
</tr>
<tr>
<td>3,2-partial relations</td>
<td>n.a.</td>
<td>795323/1108624/1286039</td>
</tr>
<tr>
<td>total relations</td>
<td>12112998/0.72/0.79</td>
<td>11427572/0.83/0.90</td>
</tr>
<tr>
<td>1.0i (L)</td>
<td>10947914</td>
<td>10947914</td>
</tr>
<tr>
<td>sieving time (days)</td>
<td>776</td>
<td>707</td>
</tr>
<tr>
<td>estimated time per relation (simulation)</td>
<td>0.95s</td>
<td>1.04s</td>
</tr>
<tr>
<td>estimated # relations (simulation)</td>
<td>12.0M−12.5M</td>
<td>11.8M−12.2M</td>
</tr>
<tr>
<td># relations with more than 2 linear primes &gt;6M</td>
<td>2601059</td>
<td>2026232</td>
</tr>
<tr>
<td># ideals of norm $&gt;1$</td>
<td>11137981</td>
<td>11066890</td>
</tr>
<tr>
<td>matrix size</td>
<td>1163421×1252099</td>
<td>1135638×1224487</td>
</tr>
<tr>
<td># non-duplicate relations in matrix</td>
<td>3529432</td>
<td>3099616</td>
</tr>
<tr>
<td># non-duplicate relations in matrix</td>
<td>539731</td>
<td>395957</td>
</tr>
<tr>
<td>with more than 2 linear primes &gt;6M</td>
<td>23346515</td>
<td>22399998</td>
</tr>
</tbody>
</table>

Table 15: Comparison of 7, 211− with 7, 211+
We conclude from this experiment that the three-large-primes method was still not necessary, so the number 7, 211+ could easily have been sieved by the two-large-primes method. In a further comparison experiment one might try equal factor base bounds for both the 2-large-primes and the 3-large-prime while having a smaller sieving region for the 3-large-primes bound. The examples given in Section 8 show that we can reasonably well estimate the number of partial relations with the formulae provided and can use this for calculating how many total relations to expect in the two-large-primes method for given parameter choices. However, calculating the heuristic \( \alpha(F_k, B_k) \) might be too cumbersome and so a short sieving experiment will usually be preferred. Moreover, a sieving simulation will also provide a global time estimate. To improve the estimates one would probably need to split the sieving region into smaller regions and calculate the mean absolute value of the polynomials over the smaller regions.

In Section 9 we describe the obstructions which are encountered when going from two to three large primes. These obstructions forced us to avoid the “ideal” three-large-primes method which would generate all possible three-large-primes relations and would consequently be too costly in time. Instead we chose for an approach which abandons unpromising candidates quickly.

Our theoretical estimates for the three-large-primes relations indicate how many relations would be obtained with the “ideal” siever and so give a useful measure of how far the real siever (with its parameter choices for \( P - 1 \)) is off from the “ideal” siever.

For the sieving of the record SNFS number 2, 773+ (see Section 10) the three-large-primes method was convenient to keep the factor base small and equal for all participating sieving computers. However, it would also have been possible to sieve with the two-large-primes method with a larger factor base on machines with sufficiently large memory in combination with the three-large-primes method on small memory machines.

The comparison between the sieving of 7, 211− and 7, 211+ which were sieved with the two-large-primes method and the three-large-primes method with a smaller factor base bound, respectively, did not show a significant difference between the two approaches.

The general number RSA-155 (see [9]) was still sieved with the two-large-primes method, though for a considerable part with the lattice siever with two large primes. That method can be seen as a kind of three-large-primes method because of the additional special prime.

The other 15-digit GNFS factorisation [1] was done with the line-by-line siever, presumably with 2 large primes. The sieving took longer than for RSA-155 but, apart from the choice of the siever, this may also be due to the polynomial and other parameter choices.

For further research it might be interesting to study the influence of the three-large-primes method on the matrix by sieving a number twice (or two similar numbers, as we did with 7, 211− and 7, 211+) once with the two-large-primes method, once with the three-large-primes method while using identical parameters (in particular, also the factor base bound and the large primes bound are identical) except for the sieving region which can be smaller for the three-large-primes method.

**Appendix A. Proofs of formulae from Section 7**

The following two propositions give the proofs for the two improved formulae for the Taylor coefficients of \( \rho \).

**Proposition 1 (Patterson Runsey \( c_i^{(k)} \)).** Let \( \rho(k - \xi) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i \) with \( \xi \in [0, 1] \). We have

\[
c_i^{(k)} = \frac{c_i^{(k-1)} + (i-1)c_i^{(k-1)}}{k} \quad \text{for } i > 0 \text{ and } k > 1.
\]

**Proof.** From (5.1) we know

\[
\frac{d}{d\xi} \rho(k - \xi) = \frac{\rho(k - 1 - \xi)}{k - \xi}.
\]
We substitute the Taylor series of $\rho$ for the intervals $[k-1,k]$ and $[k-2,k-1]$, multiply by $k - \xi$ on both sides and get

$$(k - \xi) \frac{d}{dk} \left( \sum_{i=0}^{\infty} c_i^{(k)} \xi^i \right) = \sum_{i=0}^{\infty} c_i^{(k-1)} \xi^i.$$ 

If the sums are uniformly convergent, we can differentiate term by term which leads to

$$k \sum_{i=0}^{\infty} i c_i^{(k)} \xi^{i-1} - \sum_{i=0}^{\infty} i c_i^{(k)} \xi^i = \sum_{i=0}^{\infty} c_i^{(k-1)} \xi^i.$$ 

On comparing coefficients we obtain

$$c_i^{(k)} = \frac{c_i^{(k-1)} + (i-1)c_i^{(k)}}{k}.$$

\[\square\]

**Proposition 2 (Marsaglia Zaman Marsaglia $c_0^{(k)}$).** Let $\rho \left( k + \frac{1}{2} + \frac{1}{2}\xi \right) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i$ with $\xi \in [-1,1]$. We have

$$c_0^{(k)} = \frac{1}{2k} \left( c_0^{(k-1)} + \sum_{j=1}^{\infty} \frac{c_j^{(k-1)} + (-1)j c_j^{(k)}}{j+1} \right) \quad \text{for} \quad k > 0. \quad (A.2)$$

**Proof.** Because of

$$\rho(x) = \frac{1}{x} \int_{x-1}^{x} \rho(t) dt \quad \text{for} \quad x > 1 \quad (A.3)$$

(which is another way of defining Dickman’s $\rho$ function for $x > 1$) we can write

$$c_0^{(k)} = \rho \left( k + \frac{1}{2} \right) = \frac{1}{k + \frac{1}{2}} \int_{k - \frac{1}{2}}^{k + \frac{1}{2}} \rho(t) dt.$$ 

We split the integral in an integral from $k - \frac{1}{2}$ to $k$ and one from $k$ to $k + \frac{1}{2}$. Next we substitute $t = k - \frac{1}{2} + \frac{1}{2}z$ and $t = k + \frac{1}{2} + \frac{1}{2}z$, respectively. Hence,

$$c_0^{(k)} = \frac{1}{2k+1} \left( \int_{0}^{1} \rho \left( k - \frac{1}{2} + \frac{1}{2}z \right) dz + \int_{0}^{1} \rho \left( k + \frac{1}{2} + \frac{1}{2}z \right) dz \right).$$

We substitute the respective Taylor expansions and integrate to obtain

$$c_0^{(k)} = \frac{1}{2k+1} \left( \sum_{j=0}^{\infty} \frac{1}{j+1} \left( c_j^{(k-1)} + c_j^{(k)} (-1)^j \right) \right)$$

which implies

$$c_0^{(k)} = \frac{1}{2k} \left( c_0^{(k-1)} + \sum_{j=1}^{\infty} \frac{c_j^{(k-1)} + (-1)j c_j^{(k)}}{j+1} \right).$$

\[\square\]
For completeness we also give the proofs for the remaining two recurrence relations. The proof of Proposition 4 is taken from [2].

**Proposition 3 (Marzaglia Zaman Maraglia  \(c_i^{(k)}\)).** Let \( \rho \left(k + \frac{1}{2} + \frac{1}{2} \xi \right) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i \) with \( \xi \in [-1, 1] \). We have

\[
c_i^{(k)} = -\frac{c_i^{(k-1)} + (i-1)c_i^{(k)}}{i(2k+1)} \quad \text{for} \quad i > 0 \quad \text{and} \quad k > 0. \tag{A.4}
\]

**Proof.** From (5.1) we know

\[
d \left( \rho \left(k + \frac{1}{2} + \frac{1}{2} \xi \right) \right) = -\frac{1}{2} \rho \left(k - \frac{1}{2} + \frac{1}{2} \xi \right).
\]

We substitute the Taylor series of \( \rho \) for the intervals \([k, k+1]\) and \([k-1, k]\), multiply by \(2k+1+\xi\) on both sides and get

\[
(2k+1+\xi) \frac{d}{d\xi} \left( \sum_{i=0}^{\infty} c_i^{(k)} \xi^i \right) = -\sum_{i=0}^{\infty} c_i^{(k-1)} \xi^i.
\]

If the sums are uniformly convergent, we can differentiate term by term which leads to

\[
(2k+1) \sum_{i=0}^{\infty} c_i^{(k)} \xi^{i-1} + \sum_{i=0}^{\infty} i c_i^{(k)} \xi^i = -\sum_{i=0}^{\infty} c_i^{(k-1)} \xi^i.
\]

On comparing coefficients we obtain

\[
c_i^{(k)} = -\frac{c_i^{(k-1)} + (i-1)c_i^{(k)}}{i(2k+1)}.
\]

\[\square\]

**Proposition 4 (Patterson Rumsey  \(c_0^{(k)}\)).** Let \( \rho(k-\xi) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i \) with \( \xi \in [0, 1] \). We have

\[
c_0^{(k)} = \frac{1}{k-1} \sum_{j=1}^{\infty} c_j^{(k)} \quad \text{for} \quad k > 1. \tag{A.5}
\]

**Proof.** Because of (A.3) we have

\[
c_0^{(k)} = \rho(k) = \frac{1}{k} \int_{k-1}^{k} \rho(t) dt = \frac{1}{k} \int_{0}^{1} \rho(k-z) dz.
\]

We substitute the Taylor expansion and integrate to obtain

\[
c_0^{(k)} = \frac{1}{k} \left( \sum_{j=0}^{\infty} \frac{1}{j+1} c_j^{(k)} \right)
\]

which implies

\[
c_0^{(k)} = \frac{1}{k-1} \sum_{j=1}^{\infty} c_j^{(k)} \cdot
\]

\[\square\]

In all four propositions we implicitly assumed that sums are uniformly convergent. In fact, it can be proven by induction that the radius of convergence equals 2 for the series defined inductively by (A.1) and (A.5) with start series \(c_0^{(1)} = 1\) and \(c_1^{(1)} = 0\) for \(i > 0\), and 3 for the series defined inductively by (A.4) and (A.2) and start series \(c_0^{(0)} = 1\) and \(c_i^{(0)} = 0\) for \(i > 0\).
References