Centrum voor Wiskunde en Informatica

REPORT *RAPPORT*

MAS

Modelling, Analysis and Simulation

Modelling, Analysis and Simulation

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

S. Cavallar

REPORT MAS-R0219 AUGUST 31, 2002

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

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.
1998 ACM Computing Classification System: F.2.1.
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 Ψ function, we introduce Ψ_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 Ψ by Dickman's ρ function and extend this to Ψ_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 Λ_i defined is an upper bound for how much the G_i approximation for Ψ_i is worse than the ρ approximation of Ψ. We measure that the Λ_is 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 ρ 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.

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

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

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

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

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 $\max(k, l) \leq i$.[1]

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

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

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

The siever sieves the triples for all the primes $p \in [d, B_i]$ 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

$$|F_1(a, b)| \leq S_1 \prod_{\substack{(p,q,1) \\ 2 \leq p < d \\ p \mid F_1(a,b) \\ p \nmid \mathrm{disc}(f_1)}} p^{\lfloor \log_p d \rfloor} \prod_{\substack{(p,q,1) \\ d \leq p \leq B_1 \\ p \mid F_1(a,b)}} p \quad L_1^2$$

[1] In this respect, we differ from the notation in [11] where the method is called an i-large-primes variant, when allowing k, l-partial relations with $k + l \leq i$. In this sense, the *four large primes* in the title of [11] indicate a two-large-primes variant, whereas the three-large-primes variant investigated in this paper corresponds to five large primes in their notation. Our notation makes comparison with the Quadratic Sieve easier, where one has 1 instead of 2 polynomials. Actually, Pollard's original method is already a one-large-prime variant (in either notation), as it allows for $1, 0$-partial relations.

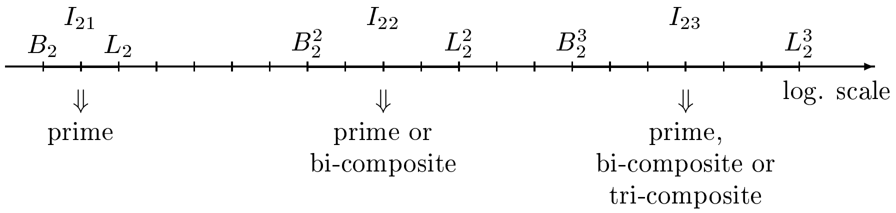

Figure 1:

and

$$|F_2(a,b)| \le S_2 \prod_{\substack{(p,q,2)\\ 2\le p<d\\ p|F_2(a,b)\\ p\nmid \mathrm{disc}(f_2)}} p^{\lfloor \log_p d\rfloor} \prod_{\substack{(p,q,2)\\ d\le p\le B_2\\ p|F_2(a,b)}} p \quad L_2^3.$$

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 initialised with 0. When sieving with $(p, q, 1)$, we add $\log p$ to each entry $a \equiv bq$. 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 resieve 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 $\log\left(\frac{|F_1(a,b)|}{S_1 L_1^2}\right)$ and $\log\left(\frac{|F_2(a,b)|}{S_2 L_2^3}\right)$ after the sieving of polynomial 1 and 2, respectively. We divide the corresponding polynomial values by the sieved primes determined by resieving 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 $L_1^2 < B_1^3$ and $L_2^3 < B_2^4$. Note that it follows that $L_1 < B_1^2$, $L_2 < B_2^2$ and $L_2^2 < B_2^3$.

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 $C_2 \ge B_2^2 > L_2$. Similarly, if $C_2 \in I_{2i}$, 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

$$\begin{array}{l} I_{21} \\ [B_2^2, B_2^{0.1}L_2^{1.9}] \\ [B_2^{2.2}L_2^{0.8}, B_2^{1.1}L_2^{1.9}] \end{array} \tag{3.2}$$

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, $b \equiv 2^M \bmod n$ 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 $\gcd(2^{Mp_i} + 2^{-Mp_i} - 2, n) > 1$ 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 $K_1 = 2\,000$ and $K_2 = 50\,000$.

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

4. Counting smooth numbers with Ψ

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

$$\Psi_i(x, y, z) = \sum_{z < p_i \leq y} \sum_{z < p_{i-1} \leq p_i} \cdots \sum_{z < p_1 \leq p_2} \Psi\left(\frac{x}{p_1 \cdots p_i}, z\right). \tag{4.1}$$

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 $\mathrm{cont}_p(F)$. Before that, we will calculate the corresponding value for random numbers, $\mathrm{cont}_p(r)$. 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 $\log F(a,b) - \sum_{p\le B} \mathrm{cont}_p(F)\log p$. The corresponding value for a random number y is $\log y - \sum_{p\le B}\mathrm{cont}_p(r)\log p$. 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*

$$\alpha(F,B) = \sum_{p\le B}(\mathrm{cont_p(r)} - \mathrm{cont_p(F)})\log p.$$

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 $\overline{x}=x/2$.

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

$$\overline{F} = \frac{\iint_R |F(a,b)|\,da\,db}{\iint_R da\,db} \tag{4.2}$$

According to Assumption 1, we can treat the $F(a,b)$ values like random values of average size $\overline{x}' = \overline{F}e^{\alpha(F,B)}$, so we use $\Psi(x',B)/x'$ with $x'=2\overline{x}'$ 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 $X=\iint_R da\,db\frac{6}{\pi^2}$ pairs such that $\gcd(a,b)=1$ (see [14, Section 4.5.2]), so we expect $X\frac{\Psi(x',B)}{x'}$ B-smooth norms among them. Because of (4.1) we can use $X\frac{\Psi_i(x',L,B)}{x'}$ 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 $X\frac{\Psi_i(x'_1,L_1,B_1)}{x'_1}\frac{\Psi_j(x'_2,L_2,B_2)}{x'_2}$ is an approximation of the number of i,j-partial relations. This means that we treat them as if the smoothness of $F_1(a,b)$ is unrelated to the smoothness of $F_2(a,b)$. By this we neglect some minor effect which can happen for primes which divide the resultant of the two polynomials.

4.2 Calculation of $cont_p(r)$ and $cont_p(F)$

In the sequel k shall always denote a positive integer.

Calculation of $cont_p(r)$ 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 $\mathrm{cont}_p(r)=\sum_{k=1}^{\infty}\frac{k}{p^k}\left(1-\frac{1}{p}\right)=\frac{1}{p-1}$.

Calculation of $cont_p(F)$ 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 Hensel'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 $\mathrm{disc} f$ we have $n_{p^k}=n_p$ for every $k\ge 1$ and we can easily sum the exponents of p to give $\mathrm{cont}_p(f) = \sum_{i=1}^{\infty} \frac{kn_p}{p^k}(1-\frac{1}{p}) = \frac{n_p}{p-1}$. For the primes diving disc$f$ we need to calculate $\mathrm{cont}_p(f)$ manually.

Next, we look into the case of a homogeneous polynomial in two variables $F(a,b) = f(a/b)b^{\deg(f)}$. We want to estimate the probability that p^k divides $F(a,b)$ restricted to the (a,b) pairs with $\gcd(a,b) = 1$. We know that if $p^k \mid F(a,b)$, then $p^k \mid F(a+lp^k, b+mp^k)$ for l and m integers. So we can restrict our analysis to $\mathbb{Z}/p^k\mathbb{Z} \times \mathbb{Z}/p^k\mathbb{Z}$. If we take a random pair (a,b) with $\gcd(a,b) = 1$, we can reduce both a and b modulo p^k and obtain one of the $p^{2k} - p^{2k-2}$ pairs (x,y) in $\mathbb{Z}/p^k\mathbb{Z} \times \mathbb{Z}/p^k\mathbb{Z}$ with $p \nmid \gcd(x,y)$. The $p^{2(k-1)}(p^2-1)$ pairs are (almost) equally likely, because in each case $\gcd(a,b) = 1$ with probability $\frac{6}{\pi^2(1-\frac{1}{p^2})}$ if we would consider an infinite (in both dimensions) sieving region.

Consider a root a/b modulo p^k, namely $f(a/b)b^{\deg(f)} \equiv 0 \bmod p^k$. We distinguish

1. $a/b \equiv s \bmod p^k$ and $p \nmid s$. There are $\phi(p^k)$ possible s between 0 and p^k and each s has $\phi(p^k)$ different pairs $(x,y) \in \mathbb{Z}/p^k\mathbb{Z} \times \mathbb{Z}/p^k\mathbb{Z}$ with $x/y \equiv s \bmod p^k$ and $p \nmid \gcd(x,y)$.

2. $a/b \equiv s \bmod p^k$ and $p \mid s$. We have p^{k-1} possible s from 0 to $p^k - 1$ and each s has $\phi(p^k)$ different pairs $(x,y) \in \mathbb{Z}/p^k\mathbb{Z} \times \mathbb{Z}/p^k\mathbb{Z}$ with $x/y \equiv s \bmod p^k$ and $p \nmid \gcd(x,y)$.

3. $b/a \equiv s \bmod p^k$ and $p \mid s$. This case is like 2, after exchanging x and y. These are called projective roots.

We see that whichever is the nature of a root with respect to p, we have $\phi(p^k)$ corresponding (x,y) pairs in $\mathbb{Z}/p^k\mathbb{Z} \times \mathbb{Z}/p^k\mathbb{Z}$ with $p \nmid \gcd(x,y)$. Thus, given n_{p^k} distinct roots modulo p^k (counting also possible projective roots), the probability for $F(a,b)$ with $\gcd(a,b) = 1$ to be divisible by p^k is equal to

$$\frac{n_{p^k}\phi(p^k)}{p^{2(k-1)}(p^2-1)} = \frac{n_{p^k}}{p^{k-1}(p+1)}. \tag{4.3}$$

Like in the univariate case we easily get the total $\log p$ contribution for primes not dividing disc$f$: we have n_p non-multiple roots of f modulo p (counting also possible projective roots), thus we find an average exponent of p equal to

$$\mathrm{cont}_p(F) = \sum_{k=1}^{\infty} \frac{kn_p}{p^{k-1}(p+1)} \left(1 - \frac{1}{p}\right) = \frac{n_p p}{p^2-1}. \tag{4.4}$$

For primes dividing disc$f$ we need to individually study the division behaviour and calculate $\mathrm{cont}_p(F)$. See Section 8 for information about the primes dividing the discriminant in our example polynomials.

5. Approximating Ψ_i

The canonical way to get an approximation for Ψ is by using Dickman's ρ function defined by

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

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

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

We will add the condition $\log x > \frac{1}{\alpha^2}$ 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 $0 < \alpha < \beta < 1$

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

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

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

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

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

In fact, in Section 5.1 we will prove

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

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

we have

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

The error bound is uniform in i, α and β.

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

$$H_i(x,z,y) =$$
$$G_i(\log_x z, \log_x y) + \frac{1-\gamma}{\log x}\frac{1}{i!}\underbrace{\int_z^y \cdots \int_z^y}_{i \text{ times}} \rho\left(\frac{\log x - \log(t_1\cdots t_i)}{\log z} - 1\right)\frac{dt_1}{t_1\log t_1}\cdots\frac{dt_i}{t_i\log t_i}. \tag{5.4}$$

with $\log x \geq \log z + i \log y$.[2]

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

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

where $1 \geq \alpha_1 \geq \cdots \geq \alpha_i \geq 0$, 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

$$\pi(t) = \mathrm{li}(t) + \epsilon(t) \tag{5.5}$$

with $\mathrm{li}(t) = \int_0^t dx/\log x = \lim_{\varepsilon\to+0}\left(\int_0^{1-\varepsilon} \frac{dx}{\log x} + \int_{1+\varepsilon}^t \frac{dx}{\log x}\right)$ and

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

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

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

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

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

Proof. For (5.7) we have

$$\begin{aligned}\int_{x^\alpha}^s \frac{dt}{t \log t} &= \left.\log\log t\right]_{x^\alpha}^s \leq \log\log x^\beta - \log\log x^\alpha = \log\beta - \log\alpha \\ &< \log\left(\frac{1}{i}\right) - \log\alpha = \log\left((i\alpha)^{-1}\right).\end{aligned}$$

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,

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

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. □

[2] or, equivalently, $\alpha + i\beta \leq 1$.

Proof of Theorem 1.

$$\begin{aligned}\Psi_i(x,x^\beta,x^\alpha) &= \sum_{x^\alpha<p_i\le x^\beta}\cdots\sum_{x^\alpha<p_1\le p_2}\Psi\left(\frac{x}{p_1\cdots p_i},x^\alpha\right)\\ &= \sum_{x^\alpha<p_i\le x^\beta}\cdots\sum_{x^\alpha<p_1\le p_2}\Psi\left(\frac{x}{p_1\cdots p_i},\left(\frac{x}{p_1\cdots p_i}\right)^{\frac{\alpha}{1-\log(p_1\cdots p_i)/\log x}}\right) \qquad (5.10)\end{aligned}$$

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

$$\begin{aligned}\frac{\Psi_i(x,x^\beta,x^\alpha)}{x} &= \sum_{x^\alpha<p_i\le x^\beta}\cdots\sum_{x^\alpha<p_1\le p_2}\frac{1}{p_1\cdots p_i}\rho\left(\frac{1-\log(p_1\cdots p_i)/\log x}{\alpha}\right)+\\ & O\left(\frac{1}{\alpha}\sum_{x^\alpha<p_i\le x^\beta}\cdots\sum_{x^\alpha<p_1\le p_2}\frac{\frac{1}{p_1\cdots p_i}}{\log(\frac{x}{p_1\cdots p_i})}\right). \qquad (5.11)\end{aligned}$$

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

$$O\left(\frac{1}{\alpha\log x}\sum_{x^\alpha<p_i\le x^\beta}\cdots\sum_{x^\alpha<p_1\le p_2}\frac{1}{p_1\cdots p_i(1-\log(p_1\cdots p_i)/\log x)}\right)=$$
$$O\left(\frac{1}{\alpha(1-i\beta)\log x}\left(\sum_{x^\alpha<p\le x^\beta}\frac{1}{p}\right)^i\right)=O\left(\frac{\log^i((i\alpha)^{-1})}{\alpha(1-i\beta)}\frac{1}{\log x}\right)$$

and by using Stieltjes integration (5.11) transforms into

$$\frac{\Psi_i(x,x^\beta,x^\alpha)}{x}=$$
$$\int_{x^\alpha}^{x^\beta}\cdots\int_{x^\alpha}^{t_2}\rho\left(\frac{1-\log(t_1\cdots t_i)/\log(x)}{\alpha}\right)\frac{d\pi(t_1)}{t_1}\cdots\frac{d\pi(t_i)}{t_i}+O\left(\frac{\log^i((i\alpha)^{-1})}{\alpha(1-i\beta)}\frac{1}{\log x}\right).(5.12)$$

We will prove, for $x^\beta\ge\cdots\ge t_{k+1}\ge t_k\ge\cdots\ge x^\alpha$, that

$$\int_{x^\alpha}^{t_{k+1}}\cdots\int_{x^\alpha}^{t_2}\rho\left(\frac{1-\log(t_1\cdots t_i)/\log x}{\alpha}\right)\frac{d\pi(t_1)}{t_1}\cdots\frac{d\pi(t_k)}{t_k}=$$
$$\int_{x^\alpha}^{t_{k+1}}\cdots\int_{x^\alpha}^{t_2}\rho\left(\frac{1-\log(t_1\cdots t_i)/\log x}{\alpha}\right)\frac{dt_1}{t_1\log t_1}\cdots\frac{dt_k}{t_k\log t_k}+O\left(\frac{\log^{k-1}((i\alpha)^{-1})}{\alpha\log x}\right) \qquad (5.13)$$

for $k\le i$ 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 $\rho(f(t_1,\ldots,t_i))\le 1$ and get

$$\begin{aligned}\int_{x^\alpha}^{t_2}\rho(f(t_1,\ldots,t_i))\frac{d\pi(t_1)}{t_1} &= \int_{x^\alpha}^{t_2}\rho(f(t_1,\ldots,t_i))\frac{dt_1}{t_1\log t_1}+\int_{x^\alpha}^{t_2}\rho(f(t_1,\ldots,t_i))\frac{d\epsilon(t_1)}{t_1}\\ &= \int_{x^\alpha}^{t_2}\rho(f(t_1,\ldots,t_i))\frac{dt_1}{t_1\log t_1}+O(1)\int_{x^\alpha}^{t_2}\frac{d\epsilon(t_1)}{t_1}.\end{aligned}$$

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 $\rho(f(t_1,\ldots,t_i))\leq 1$,

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

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^β 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

$$\int_{x^\alpha}^{x^\beta}\cdots\int_{x^\alpha}^{t_2}\rho\left(\frac{1-\log(t_1\cdots t_i)/\log x}{\alpha}\right)\frac{dt_1}{t_1\log t_1}\cdots\frac{dt_k}{t_i\log t_i}+O\left(\frac{\log^{i-1}((i\alpha)^{-1})}{\alpha}\frac{1}{\log x}\right). \quad (5.14)$$

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

$$\log((i\alpha)^{-1})>\log((i\beta)^{-1})>1-i\beta.$$

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

6. Analysis of the G_i approximations

With the help of Theorem 2 we will measure how well $G_i(\alpha,\beta)$ approximates $\Psi_i(x,x^\beta,x^\alpha)/x$ 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 $0<\alpha<\beta<\frac{1}{i}$ and i be a positive integer. Assume the Riemann hypothesis. Choose c_1 and c_2 so that*

$$c_1\leq\frac{\Psi(t,t^\gamma)}{t\rho(1/\gamma)}\leq c_2 \quad (6.1)$$

whenever $\frac{\alpha}{1-i\alpha}\leq\gamma\leq\frac{\alpha}{1-i\beta}$ and $x^{1-i\beta}\leq t\leq x^{1-i\alpha}$. Then, if $x^\alpha\geq 2\,657$, we have

$$c_1(1-\Delta_{1,i})\leq\frac{\Psi_i(x,x^\beta,x^\alpha)}{xG_i(\alpha,\beta)}\leq c_2(1+\Delta_{2,i}),$$

where $\Delta_{j,i} \le \Lambda_i(\alpha,\beta,x)$ *for* $j = 1,2$ *and*

$$\Lambda_1(\alpha,\beta,x) = \frac{\rho\left(\frac{1-\beta}{\alpha}\right)}{G_1(\alpha,\beta)} \frac{3\beta\log x}{8\pi x^{\alpha/2}} \tag{6.2}$$

and, for $i > 1$,

$$\begin{aligned}\Lambda_i(\alpha,\beta,x) \quad = \quad & \frac{\rho\left(\frac{1-i\beta}{\alpha}\right)}{G_i(\alpha,\beta)} \left(\frac{1}{i!}\left(\left(\log\left(\frac{\beta}{\alpha}\right) + \frac{3\beta\log x}{8\pi x^{\alpha/2}}\right)^i - \log^i\left(\frac{\beta}{\alpha}\right)\right)\right. \\ & \left. + \left(\frac{1}{\alpha x^\alpha \log x} + \frac{\alpha\log x}{2\pi x^{3\alpha/2}}\right)\left(\log\left(\frac{\beta}{\alpha}\right) + \frac{3\beta\log x}{8\pi x^{\alpha/2}}\right)^{i-2}\right). \end{aligned} \tag{6.3}$$

This theorem is a generalisation of Bach and Peralta's [3] Theorem 6.1 handling $i = 1$. We give a slightly better Λ_1. Table 1 contains values of Λ_i for $i \le 5$ 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 Λ_2 to Λ_5 are too large to be useful in contrast with the remaining values which are all below 4%.

In Table 1, $x'_k = 2\overline{F}_k e^{\alpha_k(F_k,B_k)}$ with $\overline{F}_k$ 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

$$|\epsilon(x)| < \frac{\sqrt{x}\log x}{8\pi} \tag{6.4}$$

which is valid under the Riemann hypothesis for $x \ge 2\,657$.

Lemma 2. *Assume the Riemann hypothesis. For* $0 < \alpha < \beta$ *and* $x^\alpha \ge 2\,657$, *we have*

$$\left|\int_{x^\alpha}^{x^\beta} \frac{d\epsilon(t)}{t}\right| \le \frac{3\beta\log x}{8\pi x^{\alpha/2}} \tag{6.5}$$

$$\int_{x^\alpha}^{x^\beta} \frac{d\pi(t)}{t} \le \log\left(\frac{\beta}{\alpha}\right) + \frac{3\beta\log x}{8\pi x^{\alpha/2}} \tag{6.6}$$

$$\int_{x^\alpha}^{x^\beta} \frac{d\pi(t)}{t^2} \le \frac{1}{\alpha x^\alpha \log x} + \frac{\alpha\log x}{2\pi x^{3\alpha/2}} \tag{6.7}$$

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

$$\begin{aligned}\left|\int_{x^\alpha}^{x^\beta} \frac{d\epsilon(t)}{t}\right| \quad & \le \quad \left|\frac{\epsilon(t)}{t}\right]_{x^\alpha}^{x^\beta}\right| + \left|\int_{x^\alpha}^{x^\beta} \frac{\epsilon(t)}{t^2}dt\right| \\ & \le \quad \frac{|\epsilon(x^\beta)|}{x^\beta} + \frac{|\epsilon(x^\alpha)|}{x^\alpha} + \frac{1}{8\pi}\int_{x^\alpha}^{x^\beta} \frac{\log t}{t^{3/2}}dt \\ & \le \quad \frac{\beta\log x}{8\pi x^{\beta/2}} + \frac{\alpha\log x}{8\pi x^{\alpha/2}} + \frac{\beta\log x}{8\pi}\int_{x^\alpha}^{x^\beta} \frac{dt}{t^{3/2}} \\ & \le \quad \frac{3\beta\log x}{8\pi x^{\alpha/2}}. \end{aligned}$$

Number	Polyn. (k)	α'_k	β'_k	x'_k	Λ_1	Λ_2	Λ_3	Λ_4	Λ_5
3,993M	1	0.186	0.205	$8.2 \cdot 10^{37}$	0.007	0.016	0.025	0.031	n.a.
3,993M	2	0.203	0.238	$5.2 \cdot 10^{32}$	0.008	0.018	0.026	0.026	n.a.
3,999L	1	0.211	0.240	$7.8 \cdot 10^{32}$	0.007	0.015	0.022	0.023	n.a.
3,999L	2	0.181	0.205	$4.3 \cdot 10^{38}$	0.007	0.015	0.024	0.030	n.a.
3,407+	1	0.169	0.190	$1.1 \cdot 10^{42}$	0.006	0.014	0.022	0.028	0.026
3,407+	2	0.211	0.241	$1.6 \cdot 10^{33}$	0.006	0.014	0.020	0.021	n.a.
3,413+	1	0.202	0.230	$6.0 \cdot 10^{34}$	0.006	0.014	0.020	0.021	n.a.
3,413+	2	0.179	0.201	$6.6 \cdot 10^{39}$	0.006	0.014	0.022	0.027	n.a.
3,427+	1	0.200	0.223	$7.2 \cdot 10^{35}$	0.006	0.013	0.020	0.021	n.a.
3,427+	2	0.179	0.198	$2.7 \cdot 10^{40}$	0.006	0.013	0.021	0.027	0.027
3,516+	1	0.204	0.234	$1.1 \cdot 10^{34}$	0.006	0.015	0.021	0.022	n.a.
3,516+	2	0.177	0.201	$3.5 \cdot 10^{39}$	0.006	0.015	0.023	0.029	n.a.
F857	1	0.166	0.188	$3.2 \cdot 10^{42}$	0.006	0.014	0.023	0.029	0.026
F857	2	0.208	0.234	$1.7 \cdot 10^{34}$	0.006	0.013	0.019	0.021	n.a.
F949	1	0.188	0.214	$2.5 \cdot 10^{37}$	0.006	0.014	0.022	–	n.a.
F949	2	0.169	0.190	$1.3 \cdot 10^{42}$	0.006	0.014	0.022	0.028	0.026
3,433+	1	0.252	0.305	$1.8 \cdot 10^{26}$	0.007	0.015	0.017	n.a.	n.a.
3,433+	2	0.154	0.170	$8.5 \cdot 10^{46}$	0.006	0.014	0.023	0.030	0.034
2,2130M	1	0.244	0.295	$1.3 \cdot 10^{27}$	0.007	0.015	0.017	n.a.	n.a.
2,2130M	2	0.152	0.168	$4.5 \cdot 10^{47}$	0.006	0.014	0.023	0.031	0.035
2,773+	1	0.161	0.198	$2.4 \cdot 10^{45}$	0.004	0.009	0.016	0.018	0.013
2,773+	2	0.162	0.200	$1.1 \cdot 10^{45}$	0.004	0.009	0.016	0.018	0.013
$7, 211-$	1	0.169	0.190	$3.0 \cdot 10^{42}$	0.006	0.013	0.021	0.026	0.024
$7, 211-$	2	0.200	0.239	$6.6 \cdot 10^{33}$	0.007	0.015	0.022	0.021	n.a.
7,211+	1	0.160	0.190	$3.0 \cdot 10^{42}$	0.007	0.017	0.028	0.034	–
7,211+	2	0.200	0.239	$6.6 \cdot 10^{33}$	0.007	0.015	0.022	0.021	n.a.
B&P	1	0.083	0.133	$1.0 \cdot 10^{60}$	0.040	0.203	0.687	1.769	3.378

Table 1: Λ_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 $\frac{\log t}{t^{1/2}}$ is decreasing in the interval $[x^\alpha, x^\beta]$. We find

$$\begin{aligned}
\int_{x^\alpha}^{x^\beta} \frac{d\pi(t)}{t^2} &= \int_{x^\alpha}^{x^\beta} \frac{dt}{t^2 \log t} + \frac{\epsilon(t)}{t^2}\Bigg]_{x^\alpha}^{x^\beta} + 2\int_{x^\alpha}^{x^\beta} \frac{\epsilon(t)}{t^3} dt \\
&\leq \frac{1}{\alpha \log x} \int_{x^\alpha}^{x^\beta} \frac{dt}{t^2} + \frac{|\epsilon(x^\beta)|}{x^{2\beta}} + \frac{|\epsilon(x^\alpha)|}{x^{2\alpha}} + \frac{1}{4\pi} \int_{x^\alpha}^{x^\beta} \frac{\log t}{t^{5/2}} dt \\
&\leq \frac{1}{\alpha \log x}\left(\frac{1}{x^\alpha} - \frac{1}{x^\beta}\right) + \frac{\log x^\beta}{8\pi x^{3\beta/2}} + \frac{\log x^\alpha}{8\pi x^{3\alpha/2}} + \frac{\alpha \log x}{4\pi x^{\alpha/2}} \int_{x^\alpha}^{x^\beta} \frac{dt}{t^2} \\
&\leq \frac{1}{\alpha x^\alpha \log x} + \frac{\alpha \log x}{2\pi x^{3\alpha/2}}.
\end{aligned}$$

□

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 $p_1, \dots, p_i$ are equal. The second term is a large upper bound for the correction.

$$\begin{aligned}
\Psi_i(x, x^\beta, x^\alpha) \leq \frac{1}{i!} \sum_{x^\alpha < p_i \leq x^\beta} \cdots \sum_{x^\alpha < p_1 \leq x^\beta} \Psi\left(\frac{x}{p_1 \cdots p_i}, \left(\frac{x}{p_1 \cdots p_i}\right)^{\frac{\alpha}{1 - \log(p_1 \cdots p_i)/\log x}}\right) + \\
\sum_{x^\alpha < p_{i-1} \leq x^\beta} \cdots \sum_{x^\alpha < p_1 \leq x^\beta} \Psi\left(\frac{x}{p_1 \cdots p_{i-2} p_{i-1}^2}, \left(\frac{x}{p_1 \cdots p_{i-2} p_{i-1}^2}\right)^{\frac{\alpha}{1 - \log(p_1 \cdots p_{i-2} p_{i-1}^2)/\log x}}\right) \qquad (6.8)
\end{aligned}$$

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

$$\begin{aligned}
\frac{\Psi_i(x, x^\beta, x^\alpha)}{x} \leq c_2 \Bigg(\frac{1}{i!} \underbrace{\int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta}}_{i \text{ times}} \rho\left(\frac{1 - \log(t_1 \dots t_i)/\log x}{\alpha}\right) \frac{d\pi(t_1)}{t_1} \cdots \frac{d\pi(t_i)}{t_i} + \\
\underbrace{\int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta}}_{i-1 \text{ times}} \rho\left(\frac{1 - \log(t_1 \dots t_{i-2} t_{i-1}^2)/\log x}{\alpha}\right) \frac{d\pi(t_1)}{t_1} \cdots \frac{d\pi(t_{i-2})}{t_{i-2}} \frac{d\pi(t_{i-1})}{t_{i-1}^2} \Bigg). \qquad (6.9)
\end{aligned}$$

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

$$\rho\left(\frac{1 - \log(t_1 \dots t_{i-2} t_{i-1}^2)/\log x}{\alpha}\right)$$

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

$$\begin{aligned}
\frac{\Psi_i(x, x^\beta, x^\alpha)}{x} \leq c_2 \Bigg(G_i(\alpha, \beta) + \frac{1}{i!} \sum_{j=1}^{i} \binom{i}{j} E_{i-j,j} + \\
\rho\left(\frac{1 - i\beta}{\alpha}\right) \left(\frac{1}{\alpha x^\alpha \log x} + \frac{\alpha \log x}{2\pi x^{3\alpha/2}}\right) \cdot \sum_{j=0}^{i-2} \binom{i-2}{j} \log^{i-2-j}\left(\frac{\beta}{\alpha}\right) \left(\frac{3\beta \log x}{8\pi x^{\alpha/2}}\right)^j \Bigg) \qquad (6.10)
\end{aligned}$$

where

$$E_{i-j,j} = \int_{x^\alpha}^{x^\beta} \cdots \int_{x^\alpha}^{x^\beta} \rho\left(\frac{1-\log(t_1\dots t_i)/\log x}{\alpha}\right) \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_{i-j}}{t_{i-j}\log t_{i-j}} \frac{d\epsilon(t_{i-j+1})}{t_{i-j+1}} \cdots \frac{d\epsilon(t_i)}{t_i}. \tag{6.11}$$

In order to find an upper bound for $E_{i-j,j}$ we bound ρ in (6.11) by $\rho\left(\frac{1-i\beta}{\alpha}\right)$, apply $i-j$ times $\int_{x^\alpha}^{x^\beta} \frac{dt}{t\log t} = \log\left(\frac{\beta}{\alpha}\right)$ and j times (6.5) and obtain

$$E_{i-j,j} \le \rho\left(\frac{1-i\beta}{\alpha}\right) \log^{i-j}\left(\frac{\beta}{\alpha}\right) \left(\frac{3\beta\log x}{8\pi x^{\alpha/2}}\right)^j. \tag{6.12}$$

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

$$\frac{\Psi_i(x, x^\beta, x^\beta)}{xG_i(\alpha,\beta)} \le c_2\,(1+\Lambda_i)$$

with Λ_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 ρ

We want to calculate Dickman's ρ function to high precision, as our estimates of smooth numbers rely on values of the ρ function.

The ρ function is the solution of a so-called differential-difference equation. This implies that it is piecewise analytic. For example

$$\rho(x) = \begin{cases} 1 & \text{if } 0 \le x \le 1 \\ 1-\log x & \text{if } 1 \le x \le 2 \end{cases}$$

For all the other intervals of length 1 we can write ρ as a Taylor series where the coefficients depend on the Taylor series coefficients of ρ 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 Rumsey; it expands the series on the right end of the intervals. The other method is by Marsaglia, Zaman and Marsaglia [16] and expands the series around the midpoints of the intervals.[3] 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 $0 \le x \le 20$ 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 ρ in the range $0 \le x \le 20$ 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's on an SGI O2 MIPS R5000 180 MHz.

[3] One might also think of expanding the series on the left end of the intervals, i.e. writing $\rho(k+\xi) = \sum_i^\infty c_i^{(k)}\xi^i$ for $0 \le \xi \le 1$. The formulae can 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, $c_i^{(1)} = \frac{(-1)^i}{i}$.

Patterson & Rumsey	Marsaglia & Zaman & Marsaglia
$0 \le \xi \le 1$ $\rho(k-\xi) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i$ $k = 1, 2, \ldots$	$-1 \le \xi \le 1$ $\rho\left(k + \frac{1}{2} + \frac{1}{2}\xi\right) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^i$ $k = 0, 1, \ldots$
$c_0^{(1)} = 1$ $c_i^{(1)} = 0 \quad \text{for} \quad i > 0$	$c_0^{(0)} = 1$ $c_i^{(0)} = 0 \quad \text{for} \quad i > 0$
$c_0^{(2)} = 1 - \log 2$ $c_i^{(2)} = \frac{1}{i2^i} \quad \text{for} \quad i > 0$	$c_0^{(1)} = 1 - \log(\frac{3}{2})$ $c_i^{(1)} = \frac{(-1)^i}{i3^i} \quad \text{for} \quad i > 0$
$c_i^{(k)} = \frac{1}{i} \sum_{j=0}^{i-1} \frac{c_j^{(k-1)}}{k^{i-j}}$ for $i > 0$ $c_0^{(k)} = \frac{1}{k-1} \sum_{j=1}^{\infty} \frac{c_j^{(k)}}{j+1}$ for $k > 1$	$c_i^{(k)} = -\frac{c_{i-1}^{(k-1)} + (i-1)c_{i-1}^{(k)}}{i(2k+1)}$ for $i > 0$ $c_0^{(k)} = \sum_{j=0}^{\infty} \left(c_j^{(k-1)} + \frac{(-1)^{j+1} m_j^{(k-1)}}{j+1} \right)$ $m_0^{(k)} = \frac{c_0^{(k)}}{2k+3}$ $m_i^{(k)} = \frac{c_i^{(k)} - m_{i-1}^{(k)}}{2k+3} \quad \text{for} \quad i > 0$ for $k > 0$

Table 2: Taylor coefficients for ρ - 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.

	Patterson & Rumsey			Marsaglia & Zaman & Marsaglia		
order rel. error	number of terms	working precision	time in s	number of terms	working precision	time in s
10^{-17}	42	17	13	87	45	6
10^{-33}	91	32	70	120	61	9
10^{-65}	195	64	400	188	93	15

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

$\frac{\rho(k-1/2+\xi/2)}{2k+1+\xi}$ 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 $1/(1-\xi/k)$ 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 also easily be derived from

Patterson & Rumsey	Marsaglia & Zaman & Marsaglia
$c_i^{(k)} = \frac{c_{i-1}^{(k-1)}+(i-1)c_{i-1}^{(k)}}{ik} \quad i > 0$	$c_i^{(k)} = -\frac{c_{i-1}^{(k-1)}+(i-1)c_{i-1}^{(k)}}{i(2k+1)} \quad i > 0$
$c_0^{(k)} = \frac{1}{k-1}\sum_{j=1}^{\infty}\frac{c_j^{(k)}}{j+1}$	$c_0^{(k)} = \frac{1}{2k}\left(c_0^{(k-1)} + \sum_{j=1}^{\infty}\frac{c_j^{(k-1)}+(-1)^j c_j^{(k)}}{j+1}\right)$
for $k > 1$	for $k > 0$

Table 4: New form of Taylor coefficients for ρ compared to the corresponding term in the other method

$$\begin{aligned} c_i^{(k)} &= \frac{1}{i}\sum_{j=0}^{i-1}\frac{c_j^{(k-1)}}{k^{i-j}} = \frac{1}{ik}\sum_{j=0}^{i-1}\frac{c_j^{(k-1)}}{k^{i-1-j}} = \frac{1}{ik}\left(\sum_{j=0}^{i-2}\frac{c_j^{(k-1)}}{k^{i-1-j}} + c_{i-1}^{(k-1)}\right) \\ &= \frac{1}{ik}\left((i-1)c_{i-1}^{(k)} + c_{i-1}^{(k-1)}\right). \end{aligned}$$

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.

	Patterson & Rumsey			Marsaglia & Zaman & Marsaglia		
order rel. error	number of terms	working precision	time in s	number of terms	working precision	time in s
10^{-17}	42	17	2	31	25	2
10^{-33}	91	32	4	64	41	3
10^{-65}	195	64	9	130	73	7

Table 5: Improved methods

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 $c_1^{(k)}$ $(k > 2)$ and M&Z&M's $c_1^{(k)}$ $(k > 1)$ 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 $2, 2h$M we denote the Aurifeuillian factor [5, III.C.2] $2^h+2^{\frac{h+1}{2}}+1$ of $2, 2h+$. Similarly, $3, 3h$M is short for $3^h + 3^{\frac{h+1}{2}} + 1$ and $3, 3h$L for $3^h - 3^{\frac{h+1}{2}} + 1$. The numbers Fx

3,993M	$f_1(x)$	$=$	$3^{55}x-1$	
	$f_2(x)$	$=$	x^6+3x^3+3	$\mathrm{cont}_3(F_2)=\frac{1}{4}$
3,999L	$f_1(x)$	$=$	$3^{55}x-1$	
	$f_2(x)$	$=$	x^6-9x^3+27	$\mathrm{cont}_3(F_2)=\frac{3}{4}$
3,407+	$f_1(x)$	$=$	$3^{37}x-3^{74}-1$	
	$f_2(x)$	$=$	$x^5-x^4-4x^3+3x^2+3x-1$	$\mathrm{cont}_{11}(F_2)=\frac{1}{12}$
3,413+	$f_1(x)$	$=$	$x-3^{59}$	
	$f_2(x)$	$=$	$x^6-x^5+x^4-x^3+x^2-x+1$	$\mathrm{cont}_7(F_2)=\frac{1}{8}$
3,427+	$f_1(x)$	$=$	$x-3^{61}$	
	$f_2(x)$	$=$	$x^6-x^5+x^4-x^3+x^2-x+1$	$\mathrm{cont}_7(F_2)=\frac{1}{8}$
3,516+	$f_1(x)$	$=$	$3^{57}x-1$	
	$f_2(x)$	$=$	x^6+3x^3+9	$\mathrm{cont}_3(F_2)=\frac{1}{2}$
F857	$f_1(x)$	$=$	$\mathrm{F171}x-\mathrm{F172}$	
	$f_2(x)$	$=$	$x^5+5x^4+10x^2-5x+2$	$\mathrm{cont}_5(F_2)=\frac{1}{3}$
F949	$f_1(x)$	$=$	$x-\mathrm{L146}$	
	$f_2(x)$	$=$	$x^6-x^5-5x^4+4x^3+6x^2-3x-1$	$\mathrm{cont}_{13}(F_2)=\frac{1}{14}$
3,433+	$f_1(x)$	$=$	$x-1018022109428884191058$	
	$f_2(x)$	$=$	$5821578000x^5$	$\mathrm{cont}_2(F_2)=3$
			$-13767381653260x^4$	$\mathrm{cont}_3(F_2)=\frac{9}{8}$
			$-3504111252981476x^3$	$\mathrm{cont}_5(F_2)=\frac{25}{24}$
			$+5033731003610092975x^2$	$\mathrm{cont}_7(F_2)=\frac{13}{48}$
			$+41414643218036780062x$	$\mathrm{cont}_{61}(F_2)=\frac{1}{62}$
			$-5635721308413922843666681$	$\mathrm{cont}_{881}(F_2)=\frac{1}{882}$
2,2130M	$f_1(x)$	$=$	$x-5310903123331135610192$	
	$f_2(x)$	$=$	$6590263680x^5$	$\mathrm{cont}_2(F_2)=\frac{8}{3}$
			$-71058983292296x^4$	$\mathrm{cont}_3(F_2)=\frac{3}{2}$
			$+1012675109422539 8x^3$	$\mathrm{cont}_5(F_2)=\frac{7}{12}$
			$+349867764197537945x^2$	$\mathrm{cont}_{19}(F_2)=\frac{37}{360}$
			$-5404582433335517396810x$	$\mathrm{cont}_{41}(F_2)=\frac{27}{560}$
			$+25814092623100339973 12415$	$\mathrm{cont}_{2003}(F_2)=\frac{8011}{4012008}$

Table 6: Detailed polynomials data

name	3,993M	3,999L	3,407+	3,413+	3,427+
SNFS difficulty	158	159	177	169	175
cofactor size	144	149	148	135	169
degree $f_1(x)$	1	1	1	1	1
degree $f_2(x)$	6	6	5	6	6
A	1680000	2520000	3600000	3360000	4200000
B	1560000	1250000	3000000	2400000	3200000
X	$3.18651 \cdot 10^{12}$	$3.82999 \cdot 10^{12}$	$1.31312 \cdot 10^{13}$	$9.80474 \cdot 10^{12}$	$1.63411 \cdot 10^{13}$
B_1	4400000	8500000	13000000	11000000	14500000
B_2	11000000	10000000	10000000	13000000	17000000
L	60000000	80000000	100000000	100000000	100000000
S_1	15	30	40	30	30
S_2	60	30	7	7	7
x_1	$2.93074 \cdot 10^{32}$	$4.39611 \cdot 10^{32}$	$6.08267 \cdot 10^{41}$	$3.39129 \cdot 10^{34}$	$4.06955 \cdot 10^{35}$
x_2	$1.87775 \cdot 10^{37}$	$1.02598 \cdot 10^{38}$	$1.53030 \cdot 10^{32}$	$6.13546 \cdot 10^{38}$	$2.54655 \cdot 10^{39}$
$\alpha(F_1, B_1)$	0.569915	0.569915	0.569915	0.569915	0.569915
$\alpha(F_2, B_2)$	1.468072	1.429203	2.319329	2.378699	2.377064
full relations	297961/0.56/0.65	412555/0.54/0.62	387672/0.63/0.73	502027/0.53/0.61	684987/0.55/0.63
0,1-partial rels.	481365/0.61/0.69	873553/0.58/0.66	737783/0.67/0.76	1047129/0.58/0.65	1205720/0.59/0.67
0,2-partial rels.	268380/0.68/0.76	633695/0.66/0.73	446398/0.74/0.81	759311/0.64/0.72	741788/0.66/0.73
1,0-partial rels.	769170/0.57/0.65	806649/0.55/0.62	944266/0.64/0.72	1008690/0.54/0.61	1194986/0.56/0.63
1,1-partial rels.	1248973/0.62/0.69	1711506/0.59/0.66	1799413/0.68/0.75	2116479/0.59/0.65	2107447/0.60/0.66
1,2-partial rels.	694993/0.70/0.76	1245009/0.67/0.73	1085377/0.74/0.81	1532260/0.66/0.71	1299863/0.67/0.72
2,0-partial rels.	627188/0.61/0.68	500656/0.58/0.64	819125/0.66/0.74	655488/0.58/0.64	686676/0.59/0.65
2,1-partial rels.	1018741/0.66/0.72	1065195/0.62/0.68	1565368/0.70/0.77	1374882/0.62/0.68	1217910/0.63/0.68
2,2-partial rels.	568849/0.74/0.79	780025/0.70/0.74	946628/0.77/0.82	1003843/0.69/0.74	752013/0.70/0.75
total relations	5975620/0.64/0.71	8028843/0.61/0.68	8732030/0.69/0.77	10000109/0.61/0.67	9891390/0.61/0.68
$1.6\mathrm{li}(L)$	5700294	7472145	9219535	9219535	9219535
sieving time (days)	148	131	98	59	112

Table 7: Examples

name	3,516+	F857	F949	3,433+	2,2130M
SNFS difficulty	165	179	184	n.a.	n.a.
cofactor size	161	179	157	115	118
degree $f_1(x)$	1	1	1	1	1
degree $f_2(x)$	6	5	6	5	6
A	3900000	6000000	8400000	70200000	97200000
B	1600000	3050000	4400000	100000	135000
X	$7.58694 \cdot 10^{12}$	$2.22502 \cdot 10^{13}$	$4.49380 \cdot 10^{13}$	$8.53532 \cdot 10^{12}$	$1.59545 \cdot 10^{13}$
B_1	8500000	11000000	11000000	4200000	4200000
B_2	10000000	13000000	13000000	16777215	16777215
L	90000000	100000000	100000000	100000000	100000000
S_1	20	30	30	20	40
S_2	30	25	1	100	2000
x_1	$6.12316 \cdot 10^{33}$	$1.79418 \cdot 10^{42}$	$1.43103 \cdot 10^{37}$	$1.01802 \cdot 10^{26}$	$7.16972 \cdot 10^{26}$
x_2	$1.04850 \cdot 10^{39}$	$6.25192 \cdot 10^{33}$	$5.73419 \cdot 10^{40}$	$4.87448 \cdot 10^{48}$	$1.65831 \cdot 10^{50}$
$\alpha(F_1, B_1)$	0.569915	0.569915	0.569915	0.569915	0.569915
$\alpha(F_2, B_2)$	1.193893	1.002230	3.153286	−4.046483	−5.915719
full relations	408537/0.47/0.54	393668/0.59/0.68	359222/0.46/0.53	446527/0.44/0.51	364736/0.54/0.62
0,1-partial rels.	935790/0.52/0.59	652752/0.64/0.72	802483/0.51/0.58	963530/0.49/0.56	812613/0.60/0.68
0,2-partial rels.	742778/0.60/0.66	336153/0.74/0.82	636660/0.58/0.65	593371/0.74/0.82	621128/0.74/0.82
1,0-partial rels.	889398/0.48/0.54	1095953/0.60/0.68	808649/0.47/0.53	1014837/0.44/0.50	865394/0.55/0.62
1,1-partial rels.	2049612/0.53/0.59	1817042/0.64/0.72	1817656/0.52/0.57	2189528/0.50/0.55	1930024/0.61/0.67
1,2-partial rels.	1628450/0.61/0.66	937005/0.74/0.81	1442146/0.59/0.65	1338927/0.75/0.81	1471358/0.75/0.82
2,0-partial rels.	623474/0.51/0.56	1071958/0.63/0.70	610361/0.50/0.55	622358/0.46/0.50	574197/0.57/0.63
2,1-partial rels.	1441725/0.56/0.61	1779998/0.68/0.74	1374875/0.55/0.60	1345966/0.51/0.55	1279510/0.63/0.68
2,2-partial rels.	1148798/0.64/0.68	916616/0.79/0.84	1094971/0.62/0.67	813549/0.78/0.82	972034/0.78/0.83
total relations	9868562/0.56/0.61	9001145/0.67/0.74	8947023/0.54/0.60	9328593/0.56/0.62	8890994/0.65/0.72
1.6li(L)	8348496	9219535	9219535	9219535	9219535
sieving time (days)	90	215	281	68	162

Table 8: Examples (continued)

are Fibonacci numbers, the numbers Lx 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

$$\begin{aligned} a &:= \text{number } r_{i,j} \text{ of } i,j\text{-partial relations,} \\ b &:= \frac{X \cdot G_i(\alpha_1', \beta_1') \cdot G_j(\alpha_2', \beta_2')}{r_{i,j}} \\ c &:= \frac{X \cdot H_i(x_1', B_1, L) \cdot H_j(x_2', B_2, L)}{r_{i,j}} \end{aligned}$$

with $x_k = 2\overline{F_k}$ (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 $1.6\mathrm{li}(L)$ 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 α 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 69% 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 $0.6 \cdot 1.6\mathrm{li}(L)$ or $0.7 \cdot 1.6\mathrm{li}(L)$ 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 $[B^2, L^2]$ 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 times 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

K_1	K_2	#factors found	f	$1-(1-f)^3$	#tri-composites found	%
500	10000	247	0.28	0.63	183	0.63
500	20000	299	0.34	0.71	205	0.70
1000	20000	332	0.38	0.76	221	0.76
1000	50000	404	0.46	0.84	244	0.84
1200	60000	429	0.49	0.87	252	0.86
2000	50000	434	0.50	0.87	255	0.87
2000	100000	488	0.56	0.91	265	0.91
10000	100000	509	0.58	0.93	267	0.91
2000	200000	540	0.62	0.94	277	0.95
5000	250000	568	0.65	0.96	280	0.96
10000	500000	621	0.71	0.98	281	0.96
12000	600000	630	0.72	0.98	281	0.96
20000	1000000	676	0.77	0.99	285	0.98
50000	1000000	676	0.77	0.99	285	0.98
20000	1500000	702	0.80	0.99	288	0.99
50000	1500000	702	0.80	0.99	288	0.99
50000	2500000	730	0.83	1.00	290	0.99
60000	3000000	737	0.84	1.00	291	1.00

Table 9: Percentages found tri-composites for some $P-1$ limits

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 10^9 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 2 000 and 50 000. For these values and for $B = 10^7$ and $L = 10^9$ 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 $(a, 9973 \cdot b)$ 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 $L = 10^9$.

A total of 1 765 748 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

K_1	K_2	not factored	factor too large	cofactor prime	three factors	step 1 successful	step 2 successful	relations found	time per relation (s)
1000	50000	43158	6669	9814	890	3991	13382	3699	6.73
1200	60000	41862	7408	10345	916	4549	14120	3725	6.70
2000	100000	38213	9583	11770	965	6402	15916	3774	6.71
5000	250000	31721	13791	14014	1005	10478	18332	3814	6.90
10000	500000	27037	17058	15410	1026	14008	19486	3835	7.25
12000	600000	25877	17892	15730	1032	15022	19632	3841	7.41
20000	1000000	22780	19994	16715	1042	17903	19848	3851	7.95
50000	2500000	17746	23535	18198	1052	23182	19603	3861	9.87
60000	3000000	16878	24129	18471	1053	24212	19441	3862	10.62

Table 10: Tri-composite factorizations for $2^{773}+1$

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 60 000. These are also the bounds used in the factorisation of $2^{773}+1$ (see Section 10).

We can see that for $P-1$ bounds 1 200 and 60 000, 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 $K_2 = 60\,000$, 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

v	tri-composite candidates	primes	not factored	factor too large	cofactor prime	three factors	step 1 successful	step 2 successful	relations found	time per relation (s)
3.0	499077	289805	10290	2344	3280	26	1305	4345	2851	8.5
2.5	635208	355779	14028	2619	3996	165	1645	5135	2990	8.1
2.0	788546	426673	18584	3194	4688	429	1972	6339	3254	7.5
1.5	876915	459383	21785	3922	4916	441	2290	6989	3266	7.5
1.0	628643	318788	15923	3927	2424	131	1592	4890	2956	8.4
0.5	57466	28589	1487	469	98	0	135	432	2825	8.6

Table 11: Tri-composite factorizations for cofactors in $[B_1^v L^{3-v}, B_1^{v-0.5} L^{3.5-v}]$

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 $K_1 = 1\,200$ and $K_2 = 60\,000$. In this table, every line gives data considering only candidate tri-composites in $[B^v L^{3-v}, B^{v-0.5} L^{3.5-v}]$ while keeping the whole interval for the bi-composites, $[B^2, L^2]$. The interval $[B^{2.2} L^{0.8}, B^{1.1} L^{1.9}]$ from (3.2) is contained in $[B^{2.5} L^{0.5}, B^{1.0} L^{2.0}]$, 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 R12000 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 $B_1 = B_2 = 2\cdot 10^7$ 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 $A = 28\,875\,000$ 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 $\mathrm{cont}_2(F_2)$ and $\mathrm{cont}_3(F_2)$ 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 $n_{2^2} = n_{2^3} = \cdots = 0$. It follows that $\mathrm{cont}_2(F_2)$ consists only of the term (4.3) with $k=1$, so $\mathrm{cont}_2(F_2) = \frac{n_2}{2+1} = \frac{1}{3}$. In an analogous way we find that $\mathrm{cont}_3(F_2) = \frac{n_3}{3+1} = \frac{1}{2}$. The correction values are $\alpha(F_1, B_1) = 0.569915$ and $\alpha(F_2, B_2) = 1.938592$.

For this example, we sieved over a small part of the sieving region,[4] namely all integer pairs

$$(a,b) \in [-A, A) \times [1000001, 1000100] \quad \text{with} \quad \gcd(a,b) = 1. \tag{10.1}$$

This corresponds to about $X = 2A\sum_{1000001}^{1000100} \frac{\phi(b)}{b} \approx 3.52772\cdot 10^9$ candidate pairs. The mean value of

[4] This is because we could not take the actual siever output as we were sieving the polynomials separately.

polynomial F_1 over this region is

$$\overline{F_1} = \frac{1}{2AB}\int_{-A}^{A-1}\int_{1000001}^{1000100} |F_1(a,b)|\, da\, db \approx 6.73793 \cdot 10^{44},$$

whereas $\overline{F_2} \approx 8.197267 \cdot 10^{43}$. We put $x_i = 2 \cdot \overline{F_i}$ for $i = 1, 2$. We assume (see Assumption 1) we have got to do with random numbers of maximal size $x_1' = x_1 e^{\alpha(F_1,B_1)} \approx 2.38 \cdot 10^{45}$ and $x_2' = x_2 e^{\alpha(F_2,B_2)} \approx 1.14 \cdot 10^{45}$.

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.

			x_1		x_1'	
i	R_i	real	XG_i/R_i	XH_i/R_i	XG_i/R_i	XH_i/R_i
0	36214	36214	1.11	1.19	1.00	1.07
1	201002	201002	1.09	1.16	1.00	1.05
2	400217	397374	1.08	1.12	1.00	1.03
3	347230	184375	1.07	1.09	1.00	1.02
4	122983	0	1.04	1.05	0.99	1.00
5	11820	0	1.00	n.a.	1.00	n.a.

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.

		x_2		x_2'	
i	R_i	XG_i/R_i	XH_i/R_i	XG_i/R_i	XH_i/R_i
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.

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 $2 \cdot 10^{36}$ 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 round. However, in both cases the estimates with x' are lower than the ones with x, due to the positive α.

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

$$[-X/2 + 2^{129} \cdot 1\,000\,100, X/2 + 2^{129} \cdot 1\,000\,100), \tag{10.2}$$

which treats the same number X of candidates as in the previous section. The numbers in this interval are larger than $x = 2\overline{F_1}$ but still of the same order of magnitude.

The approximations $xG_i(\log_x B_1, \log_x L)$ 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^{int} and H_i^{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^{int} and H_i^{int}. As an estimate for smooth values in an interval $[x_l, x_r]$ we approximate $\Psi_i(x_r, y, z) - \Psi_i(x_l, y, z)$ by the derivative of the approximation (5.3)

$$G_i^{\text{int}}(x, z, y) = \frac{d}{dx}\left(xG_i(\log_x z, \log_x y)\right) =$$
$$G_i(\log_x z, \log_x y) - \frac{1}{i!}\int_z^y \cdots \int_z^y \frac{\rho\left(\frac{\log x - \log(t_1 \cdots t_i)}{\log z} - 1\right)}{\log x - \log(t_1 \cdots t_i)} \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_i}{t_i \log t_i} \tag{10.3}$$

times $x_r - x_l$ [5] for $\log x \geq \log z + i \log y$, or the derivative of the approximation (5.4)

$$H_i^{\text{int}}(x, z, y) = \frac{d}{dx}\left(xH_i(x, z, y)\right) = H_i(x, z, y) -$$
$$\frac{1}{i!}\int_z^y \cdots \int_z^y \frac{\rho\left(\frac{\log x - \log(t_1 \cdots t_i)}{\log z} - 1\right)}{\log x - \log(t_1 \cdots t_i)} \frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_i}{t_i \log t_i} -$$
$$\frac{1-\gamma}{\log x}\frac{1}{i!}\int_z^y \cdots \int_z^y \left(\frac{\rho\left(\frac{\log x - \log(t_1 \cdots t_i)}{\log z} - 1\right)}{\log x} + \frac{\rho\left(\frac{\log x - \log(t_1 \cdots t_i)}{\log z} - 2\right)}{\log x - \log(t_1 \cdots t_i) - \log z}\right)$$
$$\frac{dt_1}{t_1 \log t_1} \cdots \frac{dt_i}{t_i \log t_i} \tag{10.4}$$

times $x_r - x_l$ 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 $7,211- = 7^{211} - 1$ and $7,211+ = 7^{211} + 1$ 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.

[5] In our examples the interval bounds $x_r - x_l \ll x_l$, so it does not matter which $x \in [x_l, x_r]$ we use.

i	R_i	XG_i/R_i	XH_i/R_i	XG_i^{int}/R_i	XH_i^{int}/R_i
0	40920	1.11	1.19	0.91	0.98
1	223495	1.10	1.16	0.92	0.97
2	439114	1.09	1.13	0.93	0.97
3	374335	1.07	1.10	0.95	0.97
4	128293	1.05	1.05	1.01	n.a.

Table 14: Numbers of smooth numbers in interval (10.2) with $x = 2^{129} \cdot 1\,000\,000$, $z = B_1$ and $y = L$

We did not take advantage of already known factors

$$\begin{aligned} 7^{211} - 1 &= 2 \cdot 3 \cdot 141793 \cdot c173 \\ 7^{211} + 1 &= 2^3 \cdot 255571219 \cdot \\ &\quad 986018415638331144897705149152887163110839 \cdot c128 \end{aligned}$$

by taking for 7, 211± 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 $\text{cont}_5(F_2) = \frac{3}{8}$ and $\text{cont}_7(F_2) = \frac{1}{8}$. 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 $x \equiv 1 \bmod 8$, 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 $K_1 = 2\,000$ and $K_2 = 50\,000$ 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 $X \cdot G_i(\log_{x_1'} B_1, \log_{x_1'} L) \cdot G_j(\log_{x_2'} B_2, \log_{x_2'} L)$ 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_k' 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 `mergelevel` 1, `filtmin` 1M and `keep` 200K and later with `mergelevel` 8, `filtmin` 500K, `maxrels` 13.0 and `maxdiscard` 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.

method (# large primes)	2+2	3+2
name	$7,211-$	7,211+
SNFS difficulty	179	179
cofactor size	173	128
degree $f_1(x)$	1	1
degree $f_2(x)$	5	5
A	5400000	5400000
B	3500000	3500000
X	$2.29796 \cdot 10^{13}$	$2.29796 \cdot 10^{13}$
B_1	15000000	6000000
B_2	6000000	6000000
L	120000000	120000000
S_1	300	1
S_2	100	100
x_1	$1.68466 \cdot 10^{42}$	$1.68466 \cdot 10^{42}$
x_2	$2.36785 \cdot 10^{33}$	$2.36785 \cdot 10^{33}$
$\alpha(F_1, B_1)$	0.569915	0.569915
$\alpha(F_2, B_2)$	1.027386	1.027386
full relations	-/210279/243333	119560/73402/85456
0,1-partial relations	-/646312/731630	342560/225609/256941
0,2-partial relations	-/654225/725637	295549/228372/254836
1,0-partial relations	-/524958/596629	525659/326026/372459
1,1-partial relations	-/1613508/1793884	1506575/1002072/1119873
1,2-partial relations	-/1633264/1779189	1303147/1014341/1110699
2,0-partial relations	-/480504/536882	831856/530064/594762
2,1-partial relations	-/1476872/1614244	2394438/1629201/1788273
2,2-partial relations	-/1494954/1601021	2077068/1649149/1773624
3,0-partial relations	n.a.	318335/385258/425221
3,1-partial relations	n.a.	917502/1184126/1278512
3,2-partial relations	n.a.	795323/1198624/1268039
total relations	12112998/0.72/0.79	11427572/0.83/0.90
$1.6\text{li}(L)$	10947914	10947914
sieving time (days)	776	707
estimated time per relation (simulation)	0.95s	1.04s
estimated # relations (simulation)	12.0M–12.5M	11.8M–12.2M
# relations with more than 2 linear primes >6M	2 601 059	2 026 232
#ideals of norm >1M	11 137 981	11 096 890
matrix size	1 163 421 × 1 252 099	1 135 638 × 1 224 487
# non-duplicate relations in matrix	3 529 432	3 509 616
# non-duplicate relations in matrix with more than 2 linear primes >6M	539 731	359 597
matrix weight	23 346 515	22 599 998

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. sectionConclusions 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 155-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 ρ.

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

$$c_i^{(k)} = \frac{c_{i-1}^{(k-1)} + (i-1)c_{i-1}^{(k)}}{ki} \quad \textit{for} \quad i > 0 \quad \textit{and} \quad k > 1. \tag{A.1}$$

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 ρ for the intervals $[k-1,k]$ and $[k-2,k-1]$, multiply by $k-\xi$ on both sides and get

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

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

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

On comparing coefficients we obtain

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

□

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*

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

Proof. Because of

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

(which is another way of defining Dickman's ρ 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,

$$c_0^{(k)} = \frac{1}{2k+1}\left(\int_0^1 \rho\left(k-\frac{1}{2}+\frac{1}{2}z\right)dz + \int_{-1}^0 \rho\left(k+\frac{1}{2}+\frac{1}{2}z\right)dz\right).$$

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

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

□

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

Proposition 3 (Marsaglia Zaman Marsaglia $c_i^{(k)}$). *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*

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

Proof. From (5.1) we know

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

We substitute the Taylor series of ρ 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

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

On comparing coefficients we obtain

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

□

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*

$$c_0^{(k)}=\frac{1}{k-1}\sum_{j=1}^{\infty}\frac{c_j^{(k)}}{j+1}\quad \textit{for}\quad k>1. \tag{A.5}$$

Proof. Because of (A.3) we have

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

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

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

□

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 $c_i^{(1)}=0$ 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

1. Fredrik Almgren, Gunnar Andersson, Torbjörn Granlund, Lars Ivansson, and Staffan Ulfberg. How we cracked the codebook ciphers. http://codebook.org/.
2. Eric Bach and René Peralta. Asymptotic semi-smoothness probabilities. Technical Report 1115, Computer Sciences Department, University of Wisconsin, Madison, 1992.
3. Eric Bach and René Peralta. Asymptotic semismoothness probabilities. *Math. Comp.*, 65(216):1701–1715, 1996.
4. Hendrik Boender. *Factoring Large Integers with the Quadratic Sieve.* PhD thesis, Rijksuniversiteit Leiden, 1997.
5. John Brillhart, D. H. Lehmer, J. L. Selfridge, Bryant Tuckerman, and S. S. Wagstaff, Jr. *Factorizations of* $b^n \pm 1$ $b = 2, 3, 5, 6, 7, 10, 11, 12$ *up to high powers*, volume 22 of *Contemporary Mathematics*. AMS, 2nd edition, 1988.
6. N. G. de Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Indag. Math.*, 13:50–60, 1951.
7. Stefania Cavallar. Strategies in filtering in the number field sieve. In Wieb Bosma, editor, *Algorithmic Number Theory - ANTS-IV*, volume 1838 of *Lecture Notes in Computer Science*, pages 209–231, Berlin, 2000. Springer.
8. Stefania Cavallar. *On the Number Field Sieve Integer Factorisation Algorithm.* PhD thesis, Universiteit Leiden, 2002.
9. Stefania Cavallar, Bruce Dodson, Arjen K. Lenstra, Walter Lioen, Peter L. Montgomery, Brian Murphy, Herman te Riele, Karen Aardal, Jeff Gilchrist, Gérard Guillerm, Paul Leyland, Joël Marchand, François Morain, Alec Muffett, Chris and Craig Putnam, and Paul Zimmermann. Factorization of a 512-bit RSA modulus. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807, pages 1–18. Springer, 2000.
10. Henri Cohen. *A Course in Computational Algebraic Number Theory.* Springer, 1996.
11. Bruce Dodson and Arjen K. Lenstra. NFS with four large primes: An explosive experiment. In Don Coppersmith, editor, *Advances in Cryptology – CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 372–385, Berlin, 1995. Springer.
12. Reina-Marije Elkenbracht-Huizing. An implementation of the Number Field Sieve. *Experiment. Math.*, 5(3):231–253, 1996.

13. Simon Hunter and Jonathan Sorenson. Approximating the number of integers free of large prime factors. *Math. Comp.*, 66(220):1729–1741, 1997.
14. Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming.* Addison-Wesley, third edition, 1998.
15. R. Lambert. *Computational aspects of discrete logarithms.* PhD thesis, University of Waterloo, 1996.
16. George Marsaglia, Arif Zaman, and John C. W. Marsaglia. Numerical solution of some classical differential-difference equations. *Math. Comp.*, 53(187):191–201, 1989.
17. Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Math. Comp.*, 48(177):243–264, 1987.
18. Peter L. Montgomery. Comments in rootfinder program, 1992.
19. Brian Antony Murphy. *Polynomial Selection for the Number Field Sieve Integer Factorisation Algorithm.* PhD thesis, The Australian National University, 1999.
20. J. M. Pollard. Factoring with cubic integers. In A. K. Lenstra and H. W. Lenstra, Jr., editors, *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*, pages 4–10. Springer, Berlin, 1993.
21. Herman te Riele. 233-digit SNFS factorization. Available from ftp://ftp.cwi.nl/pub/herman/SNFSrecords/SNFS-233, November 2000.
22. Lowell Schoenfeld. Sharper bounds for the Chebyshev functions $\theta(x)$ and $\psi(x)$. ii. *Math. Comp.*, 30(134):337–360, 1976.
23. A. M. Vershik. The asymptotic distribution of factorizations of natural numbers into prime divisors. *Soviet Math. Dokl.*, 34(1):57–61, 1987.