CERIAS Tech Report 2002-43

AN EXTENSION OF THE DICKMAN FUNCTION AND ITS APPLICATION

by Chaogui Zhang

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

AN EXTENSION OF THE DICKMAN FUNCTION AND ITS APPLICATION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Chaogui Zhang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2002

To Qing and Celina

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the people who supported me during these years. In the past five years, there have been happy times and hard times, but I have always been lucky to have my family and my friends who showed their great confidence in me.

I am greatly indebted to my advisor, Professor Samuel S. Wagstaff, for his guidance and encouragement. I enjoyed discussing problems with him and I appreciate his support and encouragement.

I would also like to thank Professor Tzuong-Tsieng Moh, Professor Freydoon Shahidi and Professor David Goldberg for serving on my graduate committee. Their help and suggestions are greatly appreciated.

As always, my parents gave me their unconditional support. They deserve my special thanks for their endless care and love. My wife, Qing, and my daughter, Celina, gave me the joy of home and family. Their smiles kept me going at hard times.

TABLE OF CONTENTS

Page

ABSTRACT

Zhang, Chaogui. Ph.D., Purdue University, August, 2002. An extension of the Dickman function and its application. Major Professor: Samuel S. Wagstaff, Jr.

In this thesis, we study a generalization of the Dickman function and its applications.

We first generalize the concept of a smooth integer to make it suitable to analyze the Large-Prime Variations of the Quadratic Sieve(QS) and Number Field Sieve (NFS). Smooth integers are the ones whose largest prime divisors are bounded. In our generalization, we will bound the largest prime divisor and the $(k+1)st$ largest prime divisor and ignore the size of the other prime divisors in between. Such integers will be called k-semismooth.

A heuristic argument will first be given to derive recurrence formulae for the asymptotic distribution of k-semismooth integers. Using that, we define a generalization of the Dickman function. Then we give a rigorous proof for the recurrence formulae, and explore some properties of the generalized Dickman function. We also present a method for computing this function. Numerical results and applications to the MPQS and NFS will then be discussed.

At the end, we will investigate the smooth integer distribution in a short interval, and use results available in this area to get an estimate of the function $f(n)$ defined as the smallest positive integer z such that the interval $[n, n+z]$ contains a set of integers (including n and $n+z$) whose product is a perfect square. This problem was proposed by Selfridge and Meyerowitz.

1. Introduction

1.1 Integer Factorization Algorithms

1.1.1 Why factor integers?

The problem of factorizing large integers into prime divisors has been studied by many mathematicians for several hundred years. The problem is simple to state and understand, just like many other problems in number theory, yet the solution to the problem requires the understanding of very complicated theories. The theoretical importance of the problem itself deserves all the attention it can get from mathematicians of all time. One of the greatest mathematicians in history, Carl Friedrich Gauss, said the following more than two hundred years ago [10]:

> The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. It has engaged the industry and wisdom of ancient and modern geometers to such an extent that it would be superfluous to discuss the problem at length...Further, the dignity of the science itself seems to require solution of a problem so elegant and so celebrated.

After the RSA public-key cryptosystem came into existence in the 1970s [24], the effort to factor large integers got more attention than ever before. It is conjectured, and many people believe, that breaking the RSA cryptosystem is equivalent to factorizing the modulus n used in the system. At least half of the conjecture is true, that is, if one can factor n, then one can definitely break the RSA system, although the other half of the conjecture has not been proved or disproved yet.

For any serious application of the RSA system, it's essential to keep track of the size of the integers that people can factor using the best factorization algorithms available so that one can choose a large enough modulus to make the system secure.

When Gauss wrote those words quoted above, there were no efficient algorithms for primality testing or integer factorization. Since then, significant progress has been made on both problems, especially for primality testing. Today we have good algorithms that can test the primality of an integer in nearly polynomial time. Integer factorization, however, remains a hard problem, although there are much faster algorithms now than before, like ECM, MPQS and NFS, all with heuristic sub-exponential running time.

1.1.2 Algorithms for factorizing integers, QS and NFS

We will briefly present two of the most powerful integer factorization algorithms next. Both of them are based on the Legendre's Congruence Method, which we shall explain briefly first.

Given the number N to factor, if we can find a non-trivial solution to the congruence $x^2 \equiv y^2 \pmod{N}$, then we can find a factor of N by taking the GCD of $x - y$ and N. Here, non-trivial means that $x \not\equiv \pm y \pmod{N}$. In practice, we usually ignore the requirement of non-trivialness and just try to find a solution to the congruence above. If it turns out to be a trivial solution, we try again. The reason is that for the methods we use to construct the solutions, finding many such solutions only requires a little more effort than finding one solution. Furthermore, when N is composite, if we find the solutions randomly, then at least half of the solutions are non-trivial, and when N is the product of two primes, we have exactly half of the solutions being non-trivial. So if we find many solutions to the congruence, then chances are we will be able to factor N easily.

Therefore, the question now is how to construct solutions to the congruence, in other words, finding congruent squares mod N. The Quadratic Sieve and the Number

Field Sieve both try to find such congruent squares efficiently. We shall give a brief introduction here to the QS and NFS. See [23] for an excellent presentation of all the important factorizing methods, ancient or modern, including the QS and NFS. Also see [21], [22] and [26] for details about the QS. For the NFS, [15] provides a complete treatment.

The Quadratic Sieve constructs the congruent squares from a large set of congruences of the form $x^2 \equiv y \bmod N$, where y can be completely factored using only primes from a factor base $\mathcal{B}$ consisting of -1 and prime numbers $p \leq B$ such that $(\frac{N}{p}) = 1$, where $(\frac{N}{p})$ is the Legendre symbol. Such a congruence is called a relation $r(x, y)$, and can be written as

$$x^2 \equiv \prod_{p \in \mathcal{B}} p^{e(p)} \bmod N.$$

The value $p = -1$ is needed to allow negative y. For each relation, we can define a vector in $\mathbf{GF}(2)^{\#\mathcal{B}}$ where each component of the vector corresponds to the exponent $e(p)$ mod 2. It's clear that if the number of such relations is greater than $\#\mathcal{B}$, that is, the size of the factor base, then we can use Gaussian elimination or some other method like block Lanczos over $\mathbf{GF}(2)$ to find a dependency among those vectors, which corresponds to a subset, S, of all the relations, such that the product of the right hand sides of the relations in S is a perfect square. So we have

$$\prod_{r(x,y) \in S} x^2 \equiv \prod_{r(x,y) \in S} y = x'^2 \bmod N$$

Taking the GCD of $x \pm x'$ and N, we have at least a 50% chance of factorizing N.

The way to find lots of relations efficiently is by a sieving method. Let $m = \lfloor \sqrt{N} \rfloor$ and $Q(t) = (t - m)^2 - N$. Then for each t, we trivially have $(t - m)^2 \equiv Q(t) \bmod N$. Thus, $Q(t)$ is a quadratic residue modulo N for each t. Also, $Q(t) = t^2 - 2\,t\,m + m^2 - N$ is about as large as $2\,t\,\sqrt{N}$, which is small compared to N when t is not too large. So $Q(t)$ is a relatively small quadratic residue modulo N. If $Q(t)$ is B-smooth, that is, can be factored completely using primes in $\mathcal{B}$, then we have a relation $r(x, y)$ with $x = t - m$ and $y = Q(t)$. To identify such parameters t efficiently, notice that $p \in \mathcal{B}$

divides $Q(t)$ if and only if p divides $Q(t \pm p)$. So we can locate all the parameters t where p divides $Q(t)$ by solving just one quadratic congruence $(t-m)^2 \equiv N \bmod p$. Remember that we have chosen p such that $(\frac{N}{p}) = 1$, so it has exactly two solutions. When $p = 4k+3$, the solutions are given by $t \equiv -m \pm N^{(p+1)/4} \bmod p$. When $p = 4k+1$, the solutions are more difficult to find, but there are efficient (but more complicated) algorithms to do that. The divisibility of $Q(t)$ by powers of 2 is more complicated and we do not discuss that here. The sign of $Q(t)$ is just the sign of $(t-m)^2 - N$.

The multiple polynomial variation of the QS, called MPQS, uses many quadratic polynomials $Q(t)$ so that we only need to sieve for small values of t for each polynomial. $Q(t)$ must be chosen such that it is a quadratic residue modulo N for each t just like before. Care must also be taken to make sure that $Q(t)$ takes small values (relatively speaking). For details on MPQS, see [26].

The Number Field Sieve is the most efficient factoring algorithm available today. There are two variations of NFS, the Special NFS and the General NFS. They differ only in the first step, polynomial selection. The special NFS can only factor numbers of the form $r^e + s$ since this form gives rise to very simple and good polynomials that can be used for the sieving. For the general NFS, one has to search for a polynomial that is as "good" as possible.

Two polynomials $f_1(x)$ and $f_2(x)$ must be chosen such that $f_i(m) \equiv 0 \bmod N$, $i = 1, 2$. Usually $f_2(x) = x - m$, but it need not be. Also the polynomials must be irreducible, but that is not a problem because if we find $f_1(x)$ to be reducible, then we have just factored N nontrivially.

We will assume $f_1(x) \in \mathbf{Z}[x]$ is monic and irreducible of degree $d > 1$ and assume $f_2(x) = x - m$. The restriction on f_1 being monic and f_2 being linear is to make our discussion easier. To see how to remove these restrictions and other details about NFS, please refer to [3].

In NFS, we work with the ring $\mathbf{Z}[\alpha]$ generated by a root α of f_1. One can either consider $\mathbf{Z}[\alpha]$ as a subring of the field of complex numbers or as a subring of

$\mathbf{Z}[X]/f_1\mathbf{Z}[X]$, with $\alpha = (X \bmod f_1)$. Every element of $\mathbf{Z}[\alpha]$ can be written uniquely in the form $\sum_{i=0}^{d-1} a_i\alpha^i$, with $a_0, a_1, \cdots, a_{d-1} \in \mathbf{Z}$. Since we have $m \in \mathbf{Z}$ satisfying $f(m) \equiv 0 \bmod N$, there is a natural ring homomorphism $\phi : \mathbf{Z}[\alpha] \to \mathbf{Z}/N\mathbf{Z}$ induced by $\phi(\alpha) = (m \bmod N)$. Now, suppose we can find a non-empty set S of pairs (a, b) of relatively prime integers such that the following are true:

$$\prod_{(a,b)\in S} (a + bm) \text{ is a square in } \mathbf{Z} \tag{1.1}$$

$$\prod_{(a,b)\in S} (a + b\alpha) \text{ is a square in } \mathbf{Z}[\alpha] \tag{1.2}$$

Then let $x \in \mathbf{Z}$ be a square root of $\prod_{(a,b)\in S}(a+bm)$, and $\beta \in \mathbf{Z}[\alpha]$ be a square root of $\prod_{(a,b)\in S}(a + b\alpha)$. Since $\phi(a + b\alpha) = (a + bm \bmod N)$, we have $\phi(\beta^2) = (x^2 \bmod N)$. Let $y \in \mathbf{Z}$ such that $\phi(\beta) = (y \bmod N)$. Then we have $x^2 \equiv y^2 \bmod N$, and we have our congruent squares to factor N.

Several important issues need to be addressed before this can be put to work:

1. How are the polynomials f_1 and f_2 to be constructed?

2. How do we find the set S?

3. How do we find β?

4. How much time is needed?

Paper [3] answers all the above questions carefully. Here we only touch upon the second question since that's the most relevant one for us and it's the most important and time consuming step of the NFS. As with the QS, the construction of the set S is done in two steps. First, sieving procedures are used to find a set T of pairs (a, b) such that both $a + bm$ and $a + b\alpha$ are smooth (smoothness for $a + b\alpha$ is defined below in a similar sense as that for an integer). Next, one uses linear algebra over the field $\mathbf{GF}(2)$ to find a subset $S \subset T$ such that (1.1) and (1.2) are satisfied.

Let $F_i(x,y) = y^{\deg(f_i)} f_i(x/y)$ be the homogeneous polynomial corresponding to $f_i(x)$, $i = 1, 2$. We say that $a + b\alpha$ is B-smooth if its norm $N(a + b\alpha) = F_1(a, -b)$ is B-smooth. For fixed $u > 0$, let

$$U = \{(a,b) | a, b \in \mathbf{Z}, \gcd(a,b) = 1, |a| \leq u, 0 < b \leq u\}.$$

We will look for $T \subset U$ with the properties mentioned above. U is called the sieving region.

In NFS, we have two sieves. The linear sieve is simple. Assume B is the smoothness bound. Then for each fixed integer b with $0 < b \leq u$, we first initialize an array of integers $a + bm$ for $-u \leq a \leq u$. For each prime $p \leq B$, the entries in the array corresponding to $a \equiv -bm \bmod p$ are retrieved one by one, and divided by the highest power of p that divides them. These entries are then replaced by their corresponding quotient after dividing out the powers of p. After doing this for all the primes $p \leq B$, we check the array and those entries containing 1 or -1 correspond to B-smooth $a + bm$.

In practice, however, we do not sieve as described above because that's too time consuming, especially the division by prime powers. Instead, we initialize the array with approximate logarithms of $a + bm$ to some base. We subtract the logarithms of the prime powers from the entries, saving time by not doing divisions. Then at the end, we look for those entries with values close to 0. These are called candidates. Because of all the approximations used, we need to factor the candidates by trial division to find the ones that are smooth.

The algebraic sieve is more complicated. Again, let's say B is the smoothness bound. For each prime $p \leq B$, let $R(p)$ denote the set of roots of $f_1(r) \equiv 0 \bmod p$, that is,

$$R(p) = \{r \in \{0, 1, \ldots, p-1\} \mid f_1(r) \equiv 0 \bmod p\}.$$

Then for any fixed integer b with $0 < b \leq u$ and $b \not\equiv 0 \bmod p$, the integers a with $N(a + b\alpha) \equiv 0 \bmod p$ are those with $a \equiv -br \bmod p$ for some $r \in R(p)$. Note that if

$b \equiv 0 \bmod p$, then there is no integer a with $(a, b) \in U$ and $N(a + b\alpha) \equiv 0 \bmod p$. Now for each fixed b initialize an array of integers $N(a + b\alpha)$ for $-u \leq a \leq u$. For each pair (p, r) such that $r \in R(p)$, the entries corresponding to $a \equiv -br \bmod p$ are identified and divided by the highest power of p that divides them. Then these entries are replaced by the quotients. After this is done for all $p \leq B$, we find those entries containing 1 or -1, which correspond to the B-smooth values of $a + b\alpha$. Again, for efficiency, we use the approximate logarithm technique mentioned above in practice.

Taking the entries (a, b) such that $\gcd(a, b) = 1$ and both $a + bm$ and $a + b\alpha$ are B-smooth, we get our set T. Apply linear algebra over $\mathbf{GF}(2)$ now to find the subset S. Here more complication arises. The problem comes from the fact that the norm of $\beta \in \mathbf{Z}[\alpha]$ being a square does not necessarily mean β itself is a square, although when β is a square, its norm is definitely a square. Fortunately, this can be solved if we remember, for each prime p dividing $N(a + b\alpha)$, the value $r \in R(p)$ which is "responsible for it". For more details, see [3].

1.1.3 Large-Prime Variations of the MPQS and NFS

The QS and NFS algorithm both run much faster if we allow in our relations not only those that can be factored completely using primes in the factor base, but also those that may have a few large prime divisors outside the factor base but below another slightly larger bound. Relations with large primes involved are usually called partial relations and the ones with no large primes are called full relations. Partial relations need to be combined before they can be used in the linear algebra step. A lot of experiments have been done with the QS and NFS using large primes. For the QS, as many as 3 large primes were used. See [16], [4] and [17] for details. For the NFS, because we can distribute the large primes over two sieves, for example, three for one polynomial and two for the other, as many as 5 large primes were used. See, for example, [7] and [5].

First we explain how partial relations can be efficiently collected during the sieving process. When sieving for full relations, only those candidates that have complete factorization in the factor base are saved. Suppose that we want to find those relations that have exactly 1 large prime (1-LP) between B and L ($B < L < B^2$). Then after the trial division, we check the remaining cofactor c. If $c < L < B^2$, then we have a 1-LP partial relation because c must be prime. As we can see, 1-LP relations are found at almost no extra cost. That's why QS and NFS with 1-LP always perform better than the no large prime versions. To collect relations with 2-LP, we check for those cofactors c with $B^2 < c < L^2 < B^3$. If c satisfies this condition, then it's either the product of two primes between B and L, that is, we have a 2-LP relation, or a single prime $> B^2 > L$, that is, we have a false report (useless relation). We distinguish these two cases by applying a compositeness test on c, and in case of a 2-LP relation, we then factor c to find the two large primes. Similarly we can do k-LP with $k \geq 3$, but the factorization pattern of c gets more complicated and we may have more and more false reports. In fact, people have thought that letting k go beyond 2 would make the cost of identifying those partial relations outweigh the benefit we get from the partial relations, thus making it slower than the 2-LP version. See, for example, [7]. However, recent experiments have shown that the 3-LP version of MPQS (called TMPQS) is indeed about 1.75 times faster than PPMPQS, the 2-LP version of MPQS. We will explain why that is the case next.

It's necessary to combine the partial relations first to find the so-called fundamental cycles before they are used in the linear algebra step. Essentially what one does is to combine partial relations together to remove the large primes and hence obtain full relations. The number of cycles as a function of the number of partial relations was observed to behave as $c_1\, m^2$ and $c_2\, m^4$ respectively for 1-LP and 2-LP partials, where m is the number of partials and c_1, c_2 are small constants [16, 1]. However, with more large primes allowed, the behavior of the number of cycles as a function of the number of partials have shown some interesting sudden growth after initially behaving according to a power law like before. This was first observed with

NFS [7], because there relations with more than 2 large primes can be found at very little extra cost. Just recently, an experiment with the MPQS using 3 large primes confirmed this interesting behavior again [17]. It is because of this sudden growth of cycle numbers that we can overcome the extra cost in identifying partial relations with more than 2 large primes. There is no theoretical explanation yet to this sudden change of behavior of cycle numbers.

1.2 Smoothness

1.2.1 Smooth integers and the Dickman function

An integer n is said to be smooth with respect to y (or y-smooth) if all the prime factors of n are $\leq y$. Smooth integers are important to factorization algorithms like the Quadratic Sieve and the Number field Sieve because they are exactly what we seek in the sieving process (the most time consuming part) in both algorithms. So estimating the number of smooth integers available under certain conditions is very important for the running time analysis of such factorization algorithms.

There are two excellent survey papers on this subject. K. Norton [20] gave a comprehensive survey of the literature up to 1970. A. Hildebrand [12] covered all the important work on the subject from 1970 to early 1990s.

Let $\Psi(x,y) = \#\{n \leq x : n \text{ has no prime divisor } > y\}$. K. Dickman [6] was the first to obtain an asymptotic formula for $\Psi(x,y)$. He showed that for any $u > 0$,

$$\lim_{x\to\infty} \frac{\Psi(x, x^{1/u})}{x} = \rho(u),$$

where $\rho(u)$ is defined by the differential-difference equation $u\rho'(u) = -\rho(u-1)$ for $u > 1$ and $\rho(u) = 1$ for $0 \leq u \leq 1$. It can be shown that $\rho(u) = \frac{1}{u}\int_{u-1}^{u} \rho(t)\,dt$.

1.2.2 Computing the Dickman ρ function accurately

Using the fact that $\rho(u) = \frac{1}{u}\int_{u-1}^{u} \rho(t)\,dt$, one can compute $\rho(u)$ simply using numerical integration methods. However, the results obtained this way are not very

good, especially when u gets bigger, because we need to do numerical integration repeatedly and errors accumulate in the process.

We introduce Bach and Peralta's method for effective calculation of the Dickman ρ function [2] here. Notice that ρ is analytic on $[m-1, m]$ for integer $m \geq 1$, that is, there is an analytic function $\rho_m(x)$ that is equal to $\rho(x)$ on $[m-1, m]$. So we have a Taylor expansion for $\rho(x) = \rho_m(x) = \rho_m(m-\xi)$ on $[m-1, m]$,

$$\rho_m(m-\xi) = \sum_{i=0}^{\infty} c_i^{(m)} \xi^i,$$

Since $\rho_1(x) = 1$ and $\rho_2(x) = 1 - \ln(x)$, the coefficients $c_i^{(1)}$ and $c_i^{(2)}$ are completely determined.

$$c_0^{(1)} = 1, \quad c_i^{(1)} = 0 \text{ for } i > 0,$$

$$c_0^{(2)} = 1 - \ln(2), \quad c_i^{(2)} = \frac{1}{i\, 2^i} \text{ for } i > 0.$$

For $m > 2$, the coefficients $c_i^{(m)}$ can be computed by the following formulae [2]. For $i > 0, m > 2$,

$$c_i^{(m)} = \sum_{j=0}^{i-1} \frac{c_j^{(m-1)}}{i m^{i-j}}.$$

For $i = 0, m > 2$,

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

The coefficients decrease exponentially: $c_i^{(m)} \leq (\frac{1}{2})^i$ for $m \geq 2$.

A more accurate method (based on the same idea of piecewise analytic function) exists. See Marsaglia, Zaman and Marsaglia [18] for details. Instead of expanding $\rho_m(x)$ at $x = m$, the method of [18] expands $\rho_m(x)$ at $m - \frac{1}{2}$. For our purpose, Bach and Peralta's method suffices.

1.3 Outline of the Thesis

In chapter 2, we will give an extension of the Dickman ρ function and discuss some properties of the new function. Numerical results and applications to the QS and NFS will be discussed in Chapter 3. Chapter 4 will deal with a problem posed by J. L. Selfridge and A. Meyerowitz [25], the growth rate of a function connected with a perfect square product in a short interval. It turns out to depend on the smooth integer distribution in short intervals.

2. k-Semismooth Integers

2.1 Background

An integer is called smooth with respect to y if all of its divisors are $\leq y$, that is, its largest prime divisor is $\leq y$, and it's called semismooth with respect to y and z if its second largest prime divisor is $\leq y$ and its largest prime divisor is $\leq z$.

The well known Dickman function $\rho(u)$ describes the asymptotic probability of an integer being smooth. Knuth and Trabb Pardo generalized the Dickman function to analyze the size of the k-th largest prime divisor of n in [13]. Bach and Peralta gave another nice generalization of the Dickman function in [2] to study the semismooth integers. These functions play an important role in analyzing factorization algorithms like the MPQS and NFS, and their one large prime variations, because the running times of these algorithms heavily depend on the number of smooth (semismooth) integers available in a certain range.

Since in current implementations of the MPQS and NFS, relations with several large primes are being used, it is desirable to generalize the Dickman function further to analyze integers with at most k large prime divisors in a certain range and all other divisors below that. We will call such integers k-semismooth integers, and would like to find a function that estimates the asymptotic probability of an integer being k-semismooth given the smoothness bounds y and z (see the definition below).The work done by Bach and Peralta solved this problem for $k = 1$ using Stieltjes integration. Lambert [14] used the same method to solve the problem for $k = 2$. In principle, their method should work for $k > 2$, but it gets very complicated and it's not practical to carry out the analysis. We will use a different method here to solve this problem for any $k > 0$. Our method also gives a stronger result in terms of the error estimate.

Definition 2.1.1 *An integer n is called k-semismooth with respect to y and z if the largest prime divisor of n is $\leq y$ and its $(k+1)$st largest prime divisor (counting multiplicity) is $\leq z$, that is, n has at most k prime divisors between y and z.*

Following the notation in [13], we write an integer n as $n = n_1 n_2 \cdots n_r, n_1 \geq n_2 \geq \cdots \geq n_r$, where the n_i's are prime divisors of n. So n_k is the k-th largest prime divisor of n (counting with multiplicity). If n has fewer than k prime divisors, then let $n_k = 1$. Also, let $n_0 = \infty$ for convenience.

Let $\Psi_k(x, y, z)$ be the number of k-semismooth integers $\leq x$ with smoothness bounds y and z, that is,

$$\Psi_k(x, y, z) = \#\{n \leq x : n_1 \leq y, n_{k+1} \leq z\}, \quad k \geq 0. \tag{2.1}$$

This generalizes the following functions defined in [6], [2] and [13]:

$$\Psi(x, y, z) = \#\{n \leq x : n_1 \leq y, n_2 \leq z\} = \Psi_1(x, y, z);$$

$$\Psi(x, y) = \#\{n \leq x : n_1 \leq y\} = \Psi_0(x, y, y);$$

$$\Psi_k(x, y) = \#\{n \leq x : n_k \leq y\} = \Psi_{k-1}(x, x, y).$$

We will prove that

$$\lim_{x \to +\infty} \frac{\Psi_k(x, x^t, x^s)}{x} = G_k(s, t)$$

exists, for $k \geq 0$, and the functions $G_k(s, t)$ satisfy some interesting formulae. Before we do that, we will briefly explain the generalizations of the Dickman function done by Knuth, Trabb Pardo [13] and Bach, Peralta [2].

Knuth and Trabb Pardo defined a generalization of the Dickman function as follows:

$$\rho_k(u) = 1 - \int_1^u (\rho_k(t-1) - \rho_{k-1}(t-1)) \, \tfrac{dt}{t}, \quad \text{for } u > 1, \ k \geq 1; \tag{2.2}$$

$$\rho_k(u) = 1, \quad \text{for } 0 < u \leq 1, \ k \geq 1; \tag{2.3}$$

$$\rho_k(u) = 0, \quad \text{for } u \leq 0 \text{ or } k = 0. \tag{2.4}$$

and proved that for $k \geq 1$,

$$\lim_{x \to +\infty} \frac{\Psi_k(x, x^{1/u})}{x} = \rho_k(u)$$

by proving the following theorem:

Theorem 2.1.1 *Let $\Psi_k(x, y)$ and $\rho_k(u)$ be defined as above, $k \geq 1$, then*

$$\Psi_k(x, x^{1/u}) = \rho_k(u)x + \sigma_k(u)x/\ln(x) + O(u^2 x/(\ln x)^2), \tag{2.5}$$

where

$$\sigma_k(u) = (1-\gamma)(\rho_k(u-1) - \rho_{k-1}(u-1))$$

and γ is Euler's constant.

Clearly, $\rho_1(u) = \rho(u)$.

In [2], Bach and Peralta proved that if we define

$$G(s,t) = F(s) + \int_s^t F(\frac{s}{1-\lambda})\frac{d\lambda}{\lambda},$$

where $F(s) = \rho(1/s)$, then we have

Theorem 2.1.2 *If $0 < s < t < 1$, then,*

$$\Psi(x, x^t, x^s) = x\, G(s,t) + O\left(\frac{\ln(s^{-1})}{s(1-t)}\frac{x}{\ln x}\right). \tag{2.6}$$

Therefore,

$$G(s,t) = \lim_{x \to \infty} \frac{\Psi(x, x^t, x^s)}{x}.$$

In [2], it was also proved that

$$G(s,t) = \int_0^s G(\frac{s}{1-\lambda}, \frac{t}{1-\lambda})d\lambda + \int_s^t F(\frac{s}{1-\lambda})d\lambda.$$

An effective method for computing $G(s,t)$ was also discussed in [2]. Notice that this $G(s,t)$ is equal to our $G_k(s,t)$ with $k = 1$.

Lambert defined a function $G_2(s,t)$ [14] such that

$$\lim_{x \to \infty} \frac{\Psi_2(x, x^t, x^s) - \Psi_1(x, x^t, x^s)}{x} = G_2(s,t).$$

He also gave an effective method for computing his $G_2(s,t)$. Notice that this function $G_2(s,t)$ in [14] is different from our $G_k(s,t)$ with $k = 2$. More precisely, his $G_2(s,t)$ is actually our $G_2(s,t) - G_1(s,t)$.

2.2 Heuristic argument

In this section we heuristically derive properties that $G_k(s,t)$ should have if we assume at the moment that $G_k(s,t) = \lim_{x\to+\infty} \frac{\Psi_k(x,x^t,x^s)}{x}$ does exist. This argument is similar to what Knuth and Trabb Pardo did in [13].

First we will (heuristically) show that

$$\frac{\partial G_k(s,t)}{\partial t} = G_{k-1}(\frac{s}{1-t}, \frac{t}{1-t})\frac{1}{t}. \tag{2.7}$$

This will lead to our definition for $G_k(s,t)$ in the next section.

Consider $\Psi_k(x, x^{t+\Delta t}, x^s) - \Psi_k(x, x^t, x^s)$, the number of integers $n \le x$ such that n_1 is between x^t and $x^{t+\Delta t}$ and $n_{k+1} \le x^s$, where Δt is a very small positive real number and $s < t$ are between 0 and 1. Any such integer n can be obtained by multiplying a prime p between x^t and $x^{t+\Delta t}$ by an integer $m \le \frac{x}{p} \le x^{1-t}$ with $m_1 \le p \approx x^t$, $m_k \le x^s$. The number of such m is approximately $\Psi_{k-1}(x^{1-t}, x^t, x^s)$. So we have

$$\Psi_k(x, x^{t+\Delta t}, x^s) - \Psi_k(x, x^t, x^s) \approx (\pi(x^{t+\Delta t}) - \pi(x^t))\Psi_{k-1}(x^{1-t}, x^t, x^s). \tag{2.8}$$

Using the Prime Number Theorem, we know that $\pi(x^{t+\Delta t}) - \pi(x^t) \approx \frac{x^t}{t}\Delta t$, so plugging this into (2.8) and dividing by $x\Delta t$ gives us

$$\frac{\Psi_k(x, x^{t+\Delta t}, x^s) - \Psi_k(x, x^t, x^s)}{x\Delta t} \approx \frac{\Psi_{k-1}(x^{1-t}, x^t, x^s)}{x^{1-t}t}.$$

Notice that

$$\frac{\Psi_{k-1}(x^{1-t}, x^t, x^s)}{x^{1-t}t} = \frac{\Psi_{k-1}(x^{1-t}, (x^{1-t})^{\frac{t}{1-t}}, (x^{1-t})^{\frac{s}{1-t}})}{x^{1-t}\,t}.$$

So letting $x \to \infty$ gives us (assuming $\lim_{x\to\infty} \frac{\Psi_k(x,x^t,x^s)}{x}$ exists and is equal to $G_k(s,t)$)

$$\frac{G_k(s, t+\Delta t) - G_k(s,t)}{\Delta t} \approx \frac{G_{k-1}(\frac{s}{1-t}, \frac{t}{1-t})}{t}.$$

Letting $\Delta t \to 0$ gives us (2.7).

Now let's show (again heuristically) that

$$\frac{\partial G_k(s,t)}{\partial s} = \left(G_k(\frac{s}{1-s}, \frac{t}{1-s}) - G_{k-1}(\frac{s}{1-s}, \frac{t}{1-s})\right)\frac{1}{s}. \tag{2.9}$$

This is a generalization of (3.7) in [13], and it will lead us to a generalization of (4.2) in [13]. The argument is similar to what we just did for (2.7).

Consider $\Psi_k(x, x^t, x^{s+\Delta s}) - \Psi_k(x, x^t, x^s)$, the number of integers $n \le x$ such that $n_1 \le x^t$ and n_{k+1} is between x^s and $x^{s+\Delta s}$, where Δs is a very small positive real number and $s < t$ are between 0 and 1. Any such integer n can be obtained by multiplying a prime p between x^s and $x^{s+\Delta s}$ by an integer $m \le \frac{x}{p} \le x^{1-s}$ with $m_1 \le x^t$, $m_{k+1} \le p \approx x^s$ and $m_k \ge p \approx x^s$. The number of such m is approximately $\Psi_k(x^{1-s}, x^t, x^s) - \Psi_{k-1}(x^{1-s}, x^t, x^s)$. So we have

$$\begin{aligned} & \Psi_k(x, x^t, x^{s+\Delta s}) - \Psi_k(x, x^t, x^s) \\ \approx \quad & (\pi(x^{s+\Delta s}) - \pi(x^s))\Psi_k(x^{1-s}, x^t, x^s) - \Psi_{k-1}(x^{1-s}, x^t, x^s). \end{aligned}$$

The rest of the argument is similar to that of (2.7). First use the Prime Number Theorem to approximate $\pi(x^{s+\Delta s}) - \pi(x^s)$, and divide both sides by $x\Delta s$. Then letting $x \to \infty$, $\Delta s \to 0$ gives us (2.9).

2.3 Rigorous proof

The integral form of (2.7) gives us the following

Definition 2.3.1 *Let $0 < s < t$, $G_0(s,t) = F(s) = \rho(1/s)$, then for $k > 0$ we define*

$$G_k(s,t) = F(s) + \int_s^t G_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau}) \frac{d\tau}{\tau}.$$

For the error term estimate that we are going to get, we introduce a function $\lambda_k(s,t)$:

Definition 2.3.2 *Let $0 < s < t$, $\lambda_0(s,t) = (1-\gamma)F(\frac{s}{1-s})$, and for $k > 0$,*

$$\lambda_k(s,t) = (1-\gamma)F(\frac{s}{1-s}) \ + \ \int_s^t \lambda_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau}) \frac{d\tau}{\tau(1-\tau)}.$$

We will prove that

$$G_k(s,t) = \lim_{x \to +\infty} \frac{\Psi_k(x, x^t, x^s)}{x}.$$

In fact, we will prove an even stronger result that gives us the main error term, but we need to prove the following lemma first:

Lemma 2.3.1 *For $0 < s < t < 1$, $k > 0$ and p prime in the summation below,*

$$\sum_{x^s < p \le x^t} \Psi_k(\frac{x}{p}, p, x^s) = \int_{x^s}^{x^t} \Psi_k(\frac{x}{y}, y, x^s)\frac{dy}{\ln y} + O\left(\frac{x}{(\ln x)^2}\right).$$

Proof We will use the Prime Number Theorem in the form $\pi(x) = \mathrm{li}(x) + O(\frac{x}{\ln^c(x)})$ for any $c > 0$, where

$$\mathrm{li}(x) = \int_0^x \frac{dt}{\ln(t)} = \lim_{\epsilon \to 0}(\int_0^{1-\epsilon} \frac{dt}{\ln(t)} + \int_{1+\epsilon}^x \frac{dt}{\ln(t)}).$$

Our proof is based on the ideas of Knuth and Trabb Pardo [13].

Let $S_k(x, y, z) = \{n \le x : n_1 \le y, n_{k+1} \le z\}$. Then

$$\begin{aligned}
&\sum_{x^s < p \le x^t} \Psi_k(\frac{x}{p}, p, x^s) - \int_{x^s}^{x^t} \Psi_k(\frac{x}{y}, y, x^s)\frac{dy}{\ln y} \\
= &\sum_{x^s < p \le x^t} (\sum_{n \in S_k(\frac{x}{p}, p, x^s)} 1) - \int_{x^s}^{x^t} (\sum_{n \in S_k(\frac{x}{y}, y, x^s)} 1)\frac{dy}{\ln y} \\
= &\sum_{x^s < p \le x^t} (\sum_{\substack{n \le \frac{x}{p} \\ n_1 \le p \\ n_{k+1} \le x^s}} 1) - \int_{x^s}^{x^t} (\sum_{\substack{n \le \frac{x}{y} \\ n_1 \le y \\ n_{k+1} \le x^s}} 1)\frac{dy}{\ln y} \\
= &\sum_{\substack{1 \le n < x^{1-s} \\ n_1 \le \min(\frac{x}{n}, x^t) \\ n_{k+1} \le x^s}} \left((\sum_{\substack{n_1 \le p \le \frac{x}{n} \\ x^s < p \le x^t}} 1) - \int_{\max(n_1, x^s)}^{\min(\frac{x}{n}, x^t)} \frac{dy}{\ln y} \right) \\
= &\sum_{\substack{1 \le n < x^{1-s} \\ n_1 \le \min(\frac{x}{n}, x^t) \\ n_{k+1} \le x^s}} \Big(\pi(\min(\frac{x}{n}, x^t)) - \pi(\max(n_1, x^s)) + O(1) \\
&\qquad\qquad -\mathrm{li}(\min(\frac{x}{n}, x^t)) + \mathrm{li}(\max(n_1, x^s)) \Big) \\
= &\sum_{\substack{1 \le n < x^{1-s} \\ n_1 \le \min(\frac{x}{n}, x^t) \\ n_{k+1} \le x^s}} O(\frac{x/n}{\ln^c(x/n)}) \\
= &\sum_{\substack{1 \le n < x^{1-s} \\ n_1 \le \min(\frac{x}{n}, x^t) \\ n_{k+1} \le x^s}} O(\frac{x/n}{\ln^c(x^s)}) \quad = \quad O(\frac{x}{\ln^c(x^s)}) \sum_{1 \le n \le x^{1-s}} \frac{1}{n} \\
= &\ O(\frac{x}{\ln^c(x^s)} \ln(x^{1-s})) \quad = \quad O(\frac{1-s}{s^c}\frac{x}{\ln^{c-1} x}).
\end{aligned}$$

Letting c = 3 gives what we need. ■

Now, we are ready to prove our main theorem:

Theorem 2.3.2 *If* $0 < s < t < 1$, $\Psi_k(x, y, z)$, $G_k(s, t)$ *and* $\lambda_k(s, t)$ *are defined as above and* $k \geq 0$, *then we have*

$$\Psi_k(x, x^t, x^s) = xG_k(s, t) + \lambda_k(s, t)\frac{x}{\ln(x)} + O(\frac{x}{(\ln x)^2}).$$

Proof By induction on k.

For $k = 0$, the statement is simply a corollary of Theorem 2.1.1.

When $k > 0$, we have

$$\Psi_k(x, x^t, x^s) = \sum_{p \leq x^s} \#\{n \leq x : n_1 = p\} + \sum_{x^s < p \leq x^t} \#\{n \leq x : n_1 = p, n_{k+1} \leq x^s\}$$

For the first sum, we have

$$\begin{aligned} & \sum_{p \leq x^s} \#\{n \leq x : n_1 = p\} \\ = \ & \#\{n \leq x : n_1 \leq x^s\} \\ = \ & \Psi(x, x^s) \\ = \ & xF(s) + \sigma_1(\frac{1}{s})\frac{x}{\ln x} + O(\frac{x}{s^2(\ln x)^2}). \end{aligned}$$

For the second sum, we have

$$\begin{aligned} & \sum_{x^s < p \leq x^t} \#\{n \leq x : n_1 = p, n_{k+1} \leq x^s\} \\ = \ & \sum_{x^s < p \leq x^t} \#\{m \leq \frac{x}{p} : m_1 \leq p, m_k \leq x^s\} \\ = \ & \sum_{x^s < p \leq x^t} \Psi_{k-1}(\frac{x}{p}, p, x^s) \\ = \ & \int_{x^s}^{x^t} \Psi_{k-1}(\frac{x}{y}, y, x^s)\frac{dy}{\ln y} + O\left(\frac{x}{(\ln x)^2}\right) \\ = \ & \int_{x^s}^{x^t} \Psi_{k-1}(\frac{x}{y}, \left(\frac{x}{y}\right)^{\frac{\ln y}{\ln x - \ln y}}, \left(\frac{x}{y}\right)^{\frac{s \ln x}{\ln x - \ln y}})\frac{dy}{\ln y} + O\left(\frac{x}{(\ln x)^2}\right). \end{aligned}$$

By induction, we have

$$\begin{aligned}\Psi_{k-1}(\frac{x}{y}, \left(\frac{x}{y}\right)^{\frac{\ln y}{\ln x-\ln y}}, \left(\frac{x}{y}\right)^{\frac{s\ln x}{\ln x-\ln y}}) &= \frac{x}{y}G_{k-1}(\frac{s\ln x}{\ln x-\ln y}, \frac{\ln y}{\ln x-\ln y}) \\ &+ \frac{x/y}{\ln(x/y)}\lambda_{k-1}(\frac{s\ln x}{\ln x-\ln y}, \frac{\ln y}{\ln x-\ln y}) \\ &+ O\left(\frac{x/y}{\ln(x/y)^2}\right).\end{aligned}$$

So,

$$\begin{aligned}&\int_{x^s}^{x^t} \Psi_{k-1}(\frac{x}{y}, \left(\frac{x}{y}\right)^{\frac{\ln y}{\ln x-\ln y}}, \left(\frac{x}{y}\right)^{\frac{s\ln x}{\ln x-\ln y}})\frac{dy}{\ln y} \\ = &\int_{x^s}^{x^t} \frac{x}{y}G_{k-1}(\frac{s\ln x}{\ln x-\ln y}, \frac{\ln y}{\ln x-\ln y})\frac{dy}{\ln y} \\ &+ \int_{x^s}^{x^t} \frac{x/y}{\ln(x/y)}\lambda_{k-1}(\frac{s\ln x}{\ln x-\ln y}, \frac{\ln y}{\ln x-\ln y})\frac{dy}{\ln y} \\ &+ \int_{x^s}^{x^t} O\left(\frac{x/y}{\ln(x/y)^2}\right)\frac{dy}{\ln y}.\end{aligned}$$

Making a change of variable, $\tau = \frac{\ln y}{\ln x}$ in the above integrals, we get

$$\begin{aligned}&\int_{x^s}^{x^t} \Psi_{k-1}(\frac{x}{y}, \left(\frac{x}{y}\right)^{\frac{\ln y}{\ln x-\ln y}}, \left(\frac{x}{y}\right)^{\frac{s\ln x}{\ln x-\ln y}})\frac{dy}{\ln y} \\ = &\; x\int_s^t G_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\tau}{\tau} \\ &+ \frac{x}{\ln x}\int_s^t \lambda_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\tau}{\tau(1-\tau)} \\ &+ O(\frac{x}{(\ln x)^2}).\end{aligned}$$

So we have

$$\begin{aligned}&\Psi_k(x, x^t, x^s) \\ = &\; x(F(s) + \int_s^t G_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\tau}{\tau}) \\ &+ \frac{x}{\ln x}(\sigma_1(\frac{1}{s}) + \int_s^t \lambda_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\tau}{\tau(1-\tau)}) \\ &+ O(\frac{x}{(\ln x)^2}) \\ = &\; xG_k(s,t) + \lambda_k(s,t)\frac{x}{\ln(x)} + O(\frac{x}{(\ln x)^2}).\end{aligned}$$

This proves the theorem. ■

2.4 Some properties of $G_k(s,t)$

First of all, let's prove that $G_k(s,t)$ does satisfy (2.9). Instead of proving (2.9) directly, we will give a proof of its equivalent integral form.

The following lemma can be proved just like Lemma 2.3.1.

Lemma 2.4.1 *For $0 < s < t < 1$, $k > 0$ and p prime in the summation below,*

$$\sum_{x^s < p \le x^t} \Psi_k(\frac{x}{p}, x^t, p) = \int_{x^s}^{x^t} \Psi_k(\frac{x}{y}, x^t, y) \frac{dy}{\ln y} + O\left(\frac{x}{(\ln x)^2}\right).$$

Theorem 2.4.2 *Given $G_k(s,t)$ as defined in the previous section, we have*

$$G_k(s,t) = F(t) - \int_s^t (G_k(\frac{\xi}{1-\xi}, \frac{t}{1-\xi}) - G_{k-1}(\frac{\xi}{1-\xi}, \frac{t}{1-\xi}))\frac{d\xi}{\xi}. \tag{2.10}$$

Proof For $0 < s < t < 1$,

$$\begin{aligned}
& \Psi(x, x^t) - \Psi_k(x, x^t, x^s) \\
= \; & \#\{n \le x : n_1 \le x^t, n_{k+1} > x^s\} \\
= \; & \sum_{x^s < p \le x^t} \#\{n \le x : n_1 \le x^t, n_{k+1} = p\} \\
= \; & \sum_{x^s < p \le x^t} \#\{m \le \frac{x}{p} : m_1 \le x^t, m_{k+1} \le p, m_k \ge p\} \\
= \; & \sum_{x^s < p \le x^t} (\Psi_k(\frac{x}{p}, x^t, p) - \Psi_{k-1}(\frac{x}{p}, x^t, p)) \\
= \; & \int_{x^s}^{x^t} (\Psi_k(\frac{x}{y}, x^t, y) - \Psi_{k-1}(\frac{x}{y}, x^t, y))\frac{dy}{\ln y} + O(\frac{x}{(\ln x)^2}) \\
= \; & \int_{x^s}^{x^t} (\Psi_k(\frac{x}{y}, \left(\frac{x}{y}\right)^{\frac{t\ln x}{\ln x - \ln y}}, \left(\frac{x}{y}\right)^{\frac{\ln y}{\ln x - \ln y}}) - \Psi_{k-1}(\frac{x}{y}, \left(\frac{x}{y}\right)^{\frac{t\ln x}{\ln x - \ln y}}, \left(\frac{x}{y}\right)^{\frac{\ln y}{\ln x - \ln y}}))\frac{dy}{\ln y} \\
& + O(\frac{x}{(\ln x)^2}).
\end{aligned}$$

Making a substitution $\xi = \frac{\ln y}{\ln x}$ in the integral, we have

$$\begin{aligned}
& \Psi(x, x^t) - \Psi_k(x, x^t, x^s) \\
= \; & \int_s^t (\Psi_k(x^{1-\xi}, (x^{1-\xi})^{\frac{t}{1-\xi}}, (x^{1-\xi})^{\frac{\xi}{1-\xi}}) - \Psi_{k-1}(x^{1-\xi}, (x^{1-\xi})^{\frac{t}{1-\xi}}, (x^{1-\xi})^{\frac{\xi}{1-\xi}}))\frac{x^\xi\, d\xi}{\xi} \\
& + O(\frac{x}{(\ln x)^2}).
\end{aligned}$$

Dividing by x and letting x go to ∞, we get

$$F(t) - G_k(s,t) = \int_s^t (G_k(\frac{\xi}{1-\xi}, \frac{t}{1-\xi}) - G_{k-1}(\frac{\xi}{1-\xi}, \frac{t}{1-\xi}))\,\frac{d\xi}{\xi},$$

that is,

$$G_k(s,t) = F(t) - \int_s^t (G_k(\frac{\xi}{1-\xi}, \frac{t}{1-\xi}) - G_{k-1}(\frac{\xi}{1-\xi}, \frac{t}{1-\xi}))\,\frac{d\xi}{\xi}$$

and this concludes the proof. ■

Remark: (2.10) is a generalization of (2.2). Notice that $G_k(s,t) = \rho_{k+1}(\frac{1}{s})$ if $t \geq 1$. Then when we let $t = 1$, $u = \frac{1}{s}$and set the index k to $k-1$ in (2.10), we get exactly (2.2).

If we start with the definition of $G_k(s,t)$ and plug in $G_{k-1}(s,t)$ recursively until we reach $G_0(s,t) = F(s)$, we get

$$\begin{aligned}
& G_k(s,t) \\
= \;& F(s) + \int_s^t G_{k-1}(\frac{s}{1-\tau_1}, \frac{\tau_1}{1-\tau_1})\,\frac{d\tau_1}{\tau_1} \\
= \;& F(s) + \int_s^t \left(F(\frac{s}{1-\tau_1}) + \int_{\frac{s}{1-\tau_1}}^{\frac{\tau_1}{1-\tau_1}} G_{k-2}(\frac{s}{(1-\tau_1)(1-\tau_2)}, \frac{\tau_2}{1-\tau_2})\,\frac{d\tau_2}{\tau_2} \right) \frac{d\tau_1}{\tau_1} \\
= \;& F(s) + \int_s^t F(\frac{s}{1-\tau_1})\,\frac{d\tau_1}{\tau_1} + \int_s^t \int_{\frac{s}{1-\tau_1}}^{\frac{\tau_1}{1-\tau_1}} G_{k-2}(\frac{s}{(1-\tau_1)(1-\tau_2)}, \frac{\tau_2}{1-\tau_2})\,\frac{d\tau_2\,d\tau_1}{\tau_2\,\tau_1} \\
= \;& \cdots \\
= \;& F(s) + \int_s^t F(\frac{s}{1-\tau_1})\,\frac{d\tau_1}{\tau_1} + \int_s^t \int_{\frac{s}{1-\tau_1}}^{\frac{\tau_1}{1-\tau_1}} F\left(\frac{s}{(1-\tau_1)(1-\tau_2)}\right) \frac{d\tau_2\,d\tau_1}{\tau_2\,\tau_1} \\
& + \cdots + \\
& \int_s^t \int_{\frac{s}{1-\tau_1}}^{\frac{\tau_1}{1-\tau_1}} \cdots \int_{\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_{k-1})}}^{\frac{\tau_{k-1}}{1-\tau_{k-1}}} F\left(\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_k)}\right) \frac{d\tau_k \cdots d\tau_2 d\tau_1}{\tau_k \cdots \tau_2 \tau_1}.
\end{aligned}$$

Now if we define $I_0(s,t) = F(s)$ and $I_k(s,t) = \int_s^t I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\,\frac{d\tau}{\tau}$ for $k > 0$, then it's easy to see that $I_k(s,t) =$

$$\int_s^t \int_{\frac{s}{1-\tau_1}}^{\frac{\tau_1}{1-\tau_1}} \cdots \int_{\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_{k-1})}}^{\frac{\tau_{k-1}}{1-\tau_{k-1}}} F\left(\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_k)}\right) \frac{d\tau_k \cdots d\tau_2 d\tau_1}{\tau_k \cdots \tau_2 \tau_1},$$

and we have just proved the following:

$$G_k(s,t) = \sum_{j=0}^{k} I_j(s,t). \tag{2.11}$$

Remark: $I_k(s,t)$ is in fact the asymptotic probability for an integer $n \le x$ to have *exactly* k large prime divisors between x^s and x^t, and all other prime divisors $\le x^s$. In other words, $I_k(s,t) = \lim_{x\to\infty} \frac{\Psi_k(x,x^t,x^s)-\Psi_{k-1}(x,x^t,x^s)}{x}$. Notice that the function $G_2(s,t)$ defined by Lambert [14] is in fact our $I_2(s,t)$.

Because of the above interpretation of $I_k(s,t)$, we expect it to have the following properties: Assume $0 < s \le t \le 1$, $k \ge 1$.

1. If $s \ge \frac{1}{k}$, then $I_k(s,t) = 0$, therefore, $G_k(s,t) = G_{k-1}(s,t)$.

2. If $t \ge 1-(k-1)\,s$, then $I_k(s,t) = I_k(s, 1-(k-1)\,s)$, in other words, if we fix s, then $I_k(s,t)$ as a function of t is constant on $[1-(k-1)\,s, 1]$.

Before we prove them, we first give a natural explanation of the properties given the meaning of $I_k(s,t)$. The first property simply says that if $n \le x$, then n can't have k factors $> x^s$ if $s \ge 1/k$ because that would imply $n > x$, a contradiction! The second property says that if $n \le x$ has k prime factors $> x^s$, then the largest one must be $\le x^{1-(k-1)s}$, because if not, then $n > (x^s)^{k-1}\, x^{1-(k-1)s} = x$, a contradiction again!

To prove the properties, we just use the definition of $I_k(s,t)$ and induction.

Proof For $k = 1$, the first property is true because $F(\frac{s}{1-\tau}) = \rho(\frac{1-\tau}{s}) = 0$ when $\tau \ge s \ge 1$.

When $k \ge 2$, $s \ge \frac{1}{k}$ implies that when $\tau \ge s$, $\frac{s}{1-\tau} \ge \frac{s}{1-s} \ge \frac{1}{k-1}$, so by induction $I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau}) = 0$, therefore $I_k(s,t) = \int_s^t I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\,\tau}{\tau} = 0$.

The second property follows from the first one because, when $\tau \geq 1-(k-1)\,s$, we have $\frac{s}{1-\tau} \geq \frac{1}{k-1}$, so $I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau}) = 0$ and

$$\begin{aligned} I_k(s,t) &= \int_s^t I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\,\tau}{\tau} \\ &= \int_s^{1-(k-1)\,s} I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\,\tau}{\tau} + \int_{1-(k-1)\,s}^t I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\,\tau}{\tau} \\ &= \int_s^{1-(k-1)\,s} I_{k-1}(\frac{s}{1-\tau}, \frac{\tau}{1-\tau})\frac{d\,\tau}{\tau} \\ &= I_k(s, 1-(k-1)\,s)\,. \end{aligned}$$

■

3. Numerical results and applications

3.1 Preparation

To compute $G_k(s,t)$, the simplest way is to use the definition and numerical integration methods. If we have a very precise table of $G_{k-1}(s,t)$ available, then we can start with this table and use numerical integration methods to calculate $G_k(s,t)$. However, when k gets bigger, the assumption of having a precise table of $G_{k-1}(s,t)$ available is hard to satisfy because as k gets higher, the results obtained using numerical integration become less accurate.

We will present a better method for computing $G_k(s,t)$. Just like Bach and Peralta's method for computing their function $\sigma(u,v) = G(\frac{1}{u}, \frac{1}{v})$, which is equivalent to our $G_1(s,t)$, and Lambert's method for computing his function $G_2(s,t)$, which is equivalent to our $I_2(s,t)$, we will utilize the fact that the Dickman function $\rho(x)$ is piecewise analytic on the interval $[k, k+1]$ for every integer k.

Since we have $G_k(s,t) = \sum_{j=0}^{k} I_j(s,t)$, we will concentrate on computing $I_k(s,t)$ instead from now on. Recall that we have

$$\begin{aligned} & I_k(s,t) \\ = & \int_s^t \int_{\frac{s}{1-\tau_1}}^{\frac{\tau_1}{1-\tau_1}} \cdots \int_{\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_{k-1})}}^{\frac{\tau_{k-1}}{1-\tau_{k-1}}} F\left(\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_k)}\right) \frac{d\tau_k \cdots d\tau_2 d\tau_1}{\tau_k \cdots \tau_2 \tau_1} \\ = & \int \cdots \int_{(\tau_1,\tau_2,\cdots,\tau_k)\in D} F\left(\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_k)}\right) \frac{d\tau_k \cdots d\tau_2 d\tau_1}{\tau_k \cdots \tau_2 \tau_1}, \end{aligned}$$

where the domain of integration D is defined by

$$s \leq \tau_1 \leq t,$$

$$\begin{aligned}
\frac{s}{1-\tau_1} &\le \tau_2 \le \frac{\tau_1}{1-\tau_1},\\
\frac{s}{(1-\tau_1)(1-\tau_2)} &\le \tau_3 \le \frac{\tau_2}{1-\tau_2},\\
&\cdots\\
\frac{s}{(1-\tau_1)(1-\tau_2)\cdots(1-\tau_{k-1})} &\le \tau_k \le \frac{\tau_{k-1}}{1-\tau_{k-1}}.
\end{aligned}$$

Let $w_1 = \tau_1$ and $w_i = \tau_i(1-\tau_1)(1-\tau_2)\cdots(1-\tau_{i-1})$ for $2 \le i \le k$. Then under this map D is transformed to D':

$$\begin{aligned}
s &\le w_1 \le t,\\
s &\le w_2 \le w_1,\\
&\cdots\\
s &\le w_k \le w_{k-1}.
\end{aligned}$$

So making such a change of variables in the above integral gives us

$$\begin{aligned}
I_k(s,t) &= \int\cdots\int_{(w_1,w_2,\cdots,w_k)\in D'} F\left(\frac{s}{1-w_1-w_2-\cdots-w_k}\right)\frac{dw_k\cdots dw_2 dw_1}{w_k\cdots w_2 w_1}\\
&= \int_s^t\int_s^{w_1}\cdots\int_s^{w_{k-1}} \rho\left(\frac{1-w_1-w_2-\cdots-w_k}{s}\right)\frac{dw_k}{w_k}\cdots\frac{dw_2}{w_2}\frac{dw_1}{w_1}.
\end{aligned}$$

When $k=2$, we have $I_2(s,t) = \int_s^t\int_s^{w_1}\rho(\frac{1-w_1-w_2}{s})\frac{dw_2}{w_2}\frac{dw_1}{w_1}$. Noticing the symmetry in the integrand, we can change the integration limits to get

$$I_2(s,t) = \frac{1}{2}\int_s^t\int_s^t \rho\left(\frac{1-w_1-w_2}{s}\right)\frac{dw_2}{w_2}\frac{dw_1}{w_1},$$

which is exactly the formula proved by Lambert in [14].

Recall that $\rho(x)$ is analytic for $x \in [m-1, m]$ for an integer m. So if we divide the above integral into regions where $\frac{1-w_1-w_2-\cdots-w_k}{s} \in [m-1,m]$ for some m, then we can replace ρ with its Taylor expansion, and then be able to integrate term by term. To make these regions where ρ is analytic more amenable, we make another change of variable in the above integral.

Let $u_i = w_i + w_{i+1} + \cdots + w_k$ for $i = 1, 2, \cdots, k$, and $u_{k+1} = 0$ for convenience. This defines a map $\phi : \mathbf{R}^k \to \mathbf{R}^k$ by $\phi((w_1, w_2, \cdots, w_k)) = (u_1, u_2, \cdots, u_k)$, and we have

$$I_k(s,t) = \int_{\phi(D')} \cdots \int \rho(\frac{1-u_1}{s}) \frac{du_k \cdots du_2 du_1}{(u_k - u_{k+1}) \cdots (u_2 - u_3)(u_1 - u_2)}.$$

This formula can be used to calculate $I_k(s,t)$. We shall give an example for $k = 3$ in the next chapter. The same principle works for higher (and lower) k, while the first step, figuring out $\phi(D')$ explicitly using inequalities, becomes quite complicated when $k \geq 4$, at least for hand calculation.

3.2 Calculating $I_3(s,t)$

First of all, we need to express $\phi(D')$ using inequalities which can give us explicit integration limits. In other words, if we integrate over u_3 first, then u_2 and u_1, we should have u_1 bounded by constants, u_2 bounded by expressions involving only u_1, and u_3 bounded by expressions involving u_1 and u_2.

Since $s \leq w_1 \leq t$, $s \leq w_i \leq w_{i-1}, i = 2, 3$, we have

$$s \leq \quad u_1 - u_2 \quad \leq t, \tag{3.1}$$

$$s \leq \quad u_2 - u_3 \quad \leq u_1 - u_2, \tag{3.2}$$

$$s \leq \qquad u_3 \qquad \leq u_2 - u_3. \tag{3.3}$$

We can easily derive from the above inequalities that

$$3s \leq \quad u_1 \quad \leq 3t, \tag{3.4}$$

$$2s \leq \quad u_2 \quad \leq \frac{2}{3}u_1, \tag{3.5}$$

$$s \leq \quad u_3 \quad \leq \frac{1}{2}u_2. \tag{3.6}$$

(3.2) implies that $2u_2 - u_1 \leq u_3 \leq u_2 - s$, together with (3.6), we have

$$\max\{s, 2u_2 - u_1\} \leq u_3 \leq \min\{\frac{1}{2}u_2, u_2 - s\}.$$

Since $\frac{1}{2}u_2 \leq u_2 - s$, $\min\{\frac{1}{2}u_2, u_2 - s\} = \frac{1}{2}u_2$, which means

$$\max\{s, 2u_2 - u_1\} \leq u_3 \leq \frac{1}{2}u_2.$$

Therefore,

- If $2s \leq u_2 \leq \frac{u_1+s}{2}$ then $s \leq u_3 \leq \frac{1}{2}u_2$.
- If $\frac{u_1+s}{2} \leq u_2 \leq \frac{2}{3}u_1$ then $2u_2 - u_1 \leq u_3 \leq \frac{1}{2}u_2$.

In the case $2s \leq u_2 \leq \frac{u_1+s}{2}$, notice that (3.1) implies $u_1 - t \leq u_2 \leq u_1 - s$, and we have

$$\max(u_1 - t, 2s) \leq u_2 \leq \frac{u_1 + s}{2}.$$

Using this together with $3s \leq u_1 \leq 3t$, we get

- If $3s \leq u_1 \leq 2s + t$ and $2s \leq u_2 \leq \frac{u_1+s}{2}$, then $s \leq u_3 \leq \frac{1}{2}u_2$.
- If $2s + t \leq u_1 \leq s + 2t$ and $u_1 - t \leq u_2 \leq \frac{u_1+s}{2}$, then $s \leq u_3 \leq \frac{1}{2}u_2$.

In the case $\frac{u_1+s}{2} \leq u_2 \leq \frac{2}{3}u_1$, similar reasoning gives us

- If $3s \leq u_1 \leq s + 2t$ and $\frac{u_1+s}{2} \leq u_2 \leq \frac{2}{3}u_1$, then $2u_2 - u_1 \leq u_3 \leq \frac{1}{2}u_2$.
- If $s + 2t \leq u_1 \leq 3t$ and $u_1 - t \leq u_2 \leq \frac{2}{3}u_1$, then $2u_2 - u_1 \leq u_3 \leq \frac{1}{2}u_2$.

So we finally have that $\phi(D') = D_1 \cup D_2 \cup D_3 \cup D_4$, where

$$\begin{aligned}
D_1 &= \{(u_1, u_2, u_2) | 3s \leq u_1 \leq 2s + t, 2s \leq u_2 \leq \frac{u_1 + s}{2}, s \leq u_3 \leq \frac{1}{2}u_2\}, \\
D_2 &= \{(u_1, u_2, u_2) | 2s + t \leq u_1 \leq s + 2t, u_1 - t \leq u_2 \leq \frac{u_1 + s}{2}, s \leq u_3 \leq \frac{1}{2}u_2\}, \\
D_3 &= \{(u_1, u_2, u_2) | 3s \leq u_1 \leq s + 2t, \frac{u_1 + s}{2} \leq u_2 \leq \frac{2}{3}u_1, 2u_2 - u_1 \leq u_3 \leq \frac{1}{2}u_2\}, \\
D_4 &= \{(u_1, u_2, u_2) | s + 2t \leq u_1 \leq 3t, u_1 - t \leq u_2 \leq \frac{2}{3}u_1, 2u_2 - u_1 \leq u_3 \leq \frac{1}{2}u_2\},
\end{aligned}$$

and $D_i,\ i = 1, 2, 3, 4$ do not overlap with each other except on the boundaries. Therefore,

$$\begin{aligned} I_3(s,t) &= \iiint_{D_1} \rho(\frac{1-u_1}{s}) \frac{du_3 du_2 du_1}{u_3(u_2-u_3)(u_1-u_2)} \\ &+ \iiint_{D_2} \rho(\frac{1-u_1}{s}) \frac{du_3 du_2 du_1}{u_3(u_2-u_3)(u_1-u_2)} \\ &+ \iiint_{D_3} \rho(\frac{1-u_1}{s}) \frac{du_3 du_2 du_1}{u_3(u_2-u_3)(u_1-u_2)} \\ &+ \iiint_{D_4} \rho(\frac{1-u_1}{s}) \frac{du_3 du_2 du_1}{u_3(u_2-u_3)(u_1-u_2)}. \end{aligned}$$

Since we have $D_i, i = 1, 2, 3, 4$ defined above explicitly by inequalities, we can rewrite these integrals with explicit integration limits and compute them. We will give a detailed analysis for the integral over D_1 as an example. For D_2, D_3, D_4, the process is similar.

Given the integral

$$\begin{aligned} &\iiint_{D_1} \rho(\frac{1-u_1}{s}) \frac{du_3 du_2 du_1}{u_3(u_2-u_3)(u_1-u_2)} \\ = &\int_{3s}^{2s+t} \int_{2s}^{\frac{u_1+s}{2}} \int_{s}^{\frac{1}{2}u_2} \rho(\frac{1-u_1}{s}) \frac{du_3 du_2 du_1}{u_3(u_2-u_3)(u_1-u_2)}, \end{aligned}$$

we approximate $\rho(\frac{1-u_1}{s})$ in the integrand using the Taylor expansion of ρ mentioned in Chapter 1. Let $m = \lceil \frac{1-u_1}{s} \rceil$, and $\xi(u_1) = m - \frac{1-u_1}{s}$, on an interval $[a, b] \subseteq [2s, 2s+t]$ of u_1 where m is a constant. We have

$$\rho(\frac{1-u_1}{s}) = \rho(m - \xi(u_1)) = \sum_{i=0}^{\infty} c_i^{(m)} \xi(u_1)^i = \sum_{i=0}^{N} c_i^{(m)} \xi(u_1)^i + E,$$

where

$$E = \sum_{i=N+1}^{\infty} c_i^{(m)} \xi(u_1)^i.$$

So we have

$$\begin{aligned}
&\int_a^b \int_{2s}^{\frac{u_1+s}{2}} \int_s^{\frac{1}{2}u_2} \rho(\frac{1-u_1}{s}) \frac{du_3 du_2 du_1}{u_3(u_2-u_3)(u_1-u_2)} \\
= &\sum_{i=0}^{\infty} c_i^{(m)} \int_a^b \int_{2s}^{\frac{u_1+s}{2}} \int_s^{\frac{1}{2}u_2} \frac{\xi(u_1)^i}{u_3(u_2-u_3)(u_1-u_2)} du_3 du_2 du_1 \\
\approx &\sum_{i=0}^{N} c_i^{(m)} \int_a^b \int_{2s}^{\frac{u_1+s}{2}} \int_s^{\frac{1}{2}u_2} \frac{(m-\frac{1-u_1}{s})^i}{u_3(u_2-u_3)(u_1-u_2)} du_3 du_2 du_1 \\
= &\sum_{i=0}^{N} c_i^{(m)} \int_a^b \int_{2s}^{\frac{u_1+s}{2}} (m-\frac{1-u_1}{s})^i \frac{\ln(\frac{u_2-s}{s})}{u_2(u_1-u_2)} du_2 du_1 \\
= &\sum_{i=0}^{N} c_i^{(m)} \int_a^b (m-\frac{1-u_1}{s})^i \left[\frac{(\ln 2)^2}{2} + \ln(\frac{u_1+s}{s}) \ln(\frac{u_1-s}{2\,s}) \right. \\
&\qquad \left. + \mathrm{Li}_2(-\frac{u_1-s}{2\,s}) + \mathrm{Li}_2(\frac{s}{u_1-s}) \right] \frac{du_1}{u_1},
\end{aligned}$$

where

$$\mathrm{Li}_2(z) = \int_z^0 \frac{\ln(1-t)}{t}\, dt$$

is the second order polylogarithm function, also called the dilogarithm [1].

3.3 Results and implications for parameter choices of MPQS and NFS

Mathematica programs to compute $I_3(s,t)$ are given in the Appendix. Figure 3.1 shows the shape of this function in the region $[0,1/3] \times [0,1/3]$. Figures 3.2 to 3.6 give us closer looks at the behavior of the function. Table 3.3 gives values of $\sigma_3(u,v) = I_3(1/u, 1/v)$ at integral points with $4 \le u \le 20$, $1 \le v \le 10$.

It's clear from the graphs that when s is fixed, $I_3(s,t)$ increases as a function of t, but for t fixed, it's not monotonic in s. If t is fixed, $I_3(s,t)$ as a function of s will increase initially with s, but after it reaches the peak, it starts decreasing.

The shape of $I_3(s,t)$ closely resembles that of $I_2(s,t)$ (see [14]), we expect that to be the case for $I_k(s,t)$ with $k \ge 4$. So although the discussion below concerning the parameter choice for MPQS and NFS is for the 3-LP variations, we expect to

[1] Dilogarithm sometimes also refers to $\mathrm{Li}_2(1-z)$

have the same kind of results for $k \geq 4$ provided that we have good numerical values available for $I_k(s,t)$ with $k \geq 4$.

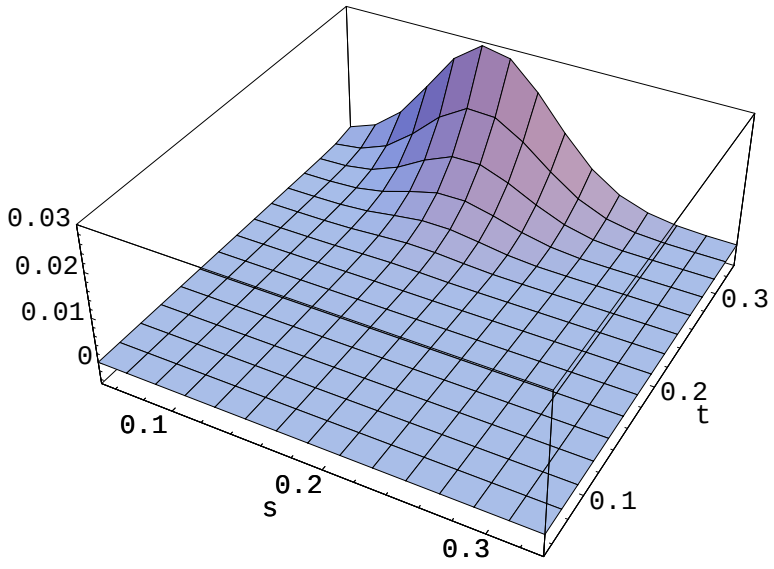

Figure 3.1. $I_3(s,t)$ with $0 \leq s \leq 1/3$, $0 \leq t \leq 1/3$

We will analyze the parameter choice problem for MPQS here, using the latest record-breaking (for MPQS) factorization of $n = 2,1606L.c135$ as an example [17]. We first give the actual parameters used in the factorization (from [17]), then examine whether these choices are optimal. The factor base size was chosen as 555 000, so B = 17 157 953 was the largest prime in the factor base. The large prime bound $L = 2^{30}$ and the sieving range was $-17158000/2 \leq t \leq 17158000/2$. These parameters were determined experimentally. Initially, a set of possible values were chosen (using past experience with the algorithm), and sieving experiments were conducted with these different parameter values. The best performer among these were used in the actual sieving. It was noted in [17] that the sieving range only slightly affected the yield of full and partial relations, with shorter sieve intervals producing somewhat higher yield.

If we fix the sieving range for the moment, we have two important parameters that may affect the algorithm, B and L. In the above setting, most of the numbers

Table 3.1
$\sigma_3(u,v) = I_3(1/u, 1/v)$ for $4 \le u \le 20$, $1 \le v \le 10$

	v									
u	1	2	3	4	5	6	7	8	9	10
4	1.48863e-2	1.48863e-2	3.96814e-3	-	-	-	-	-	-	-
5	7.12659e-2	6.80954e-2	1.88411e-2	9.42377e-4	-	-	-	-	-	-
6	1.40822e-1	1.23824e-1	2.87082e-2	2.22512e-3	9.01503e-5	-	-	-	-	-
7	1.98437e-1	1.57745e-1	2.79713e-2	1.91224e-3	1.31608e-4	5.53028e-6	-	-	-	-
8	2.40073e-1	1.70494e-1	2.23729e-2	1.08903e-3	7.2448e-5	5.59516e-6	2.5458e-7	-	-	-
9	2.69038e-1	1.70185e-1	1.64852e-2	5.08494e-4	2.63569e-5	2.15762e-6	1.90899e-7	9.42544e-9	-	-
10	2.88994e-1	1.63447e-1	1.18528e-2	2.15634e-4	7.72096e-6	5.48512e-7	5.50327e-8	5.49599e-9	2.92057e-10	-
11	3.02644e-1	1.54188e-1	8.55239e-3	8.76134e-5	2.00327e-6	1.10884e-7	1.04466e-8	1.24077e-9	1.37267e-10	7.77771e-12
12	3.11838e-1	1.44382e-1	6.27255e-3	3.50492e-5	4.84851e-7	1.95161e-8	1.56555e-9	1.84548e-10	2.51265e-11	3.02864e-12
13	3.1784e-1	1.34916e-1	4.69872e-3	1.39809e-5	1.12649e-7	3.14533e-9	2.02085e-10	2.15865e-11	3.0345e-12	4.6167e-13
14	3.21523e-1	1.26135e-1	3.59738e-3	5.58863e-6	2.55185e-8	4.78114e-10	2.35982e-11	2.16063e-12	2.87691e-13	4.65182e-14
15	3.23502e-1	1.18135e-1	2.81133e-3	2.24109e-6	5.68072e-9	6.97564e-11	2.56806e-12	1.94103e-13	2.3249e-14	3.6777e-15
16	3.24215e-1	1.10898e-1	2.23805e-3	9.00848e-7	1.24724e-9	9.8679e-12	2.65309e-13	1.61185e-14	1.67775e-15	2.47298e-16
17	3.23981e-1	1.04367e-1	1.81097e-3	3.62491e-7	2.7045e-10	1.36125e-12	2.63237e-14	1.26089e-15	1.11296e-16	1.48017e-17
18	3.23036e-1	9.84702e-2	1.48648e-3	1.45822e-7	5.7937e-11	1.83686e-13	2.52645e-15	9.4074e-17	6.91693e-18	8.11335e-19
19	3.21555e-1	9.31364e-2	1.23551e-3	5.85742e-8	1.22605e-11	2.42862e-14	2.35594e-16	6.7491e-18	4.07936e-19	4.15042e-20
20	3.1967e-1	8.82996e-2	1.03829e-3	2.34698e-8	2.56242e-12	3.14902e-15	2.1404e-17	4.68096e-19	2.3031e-20	2.00733e-21

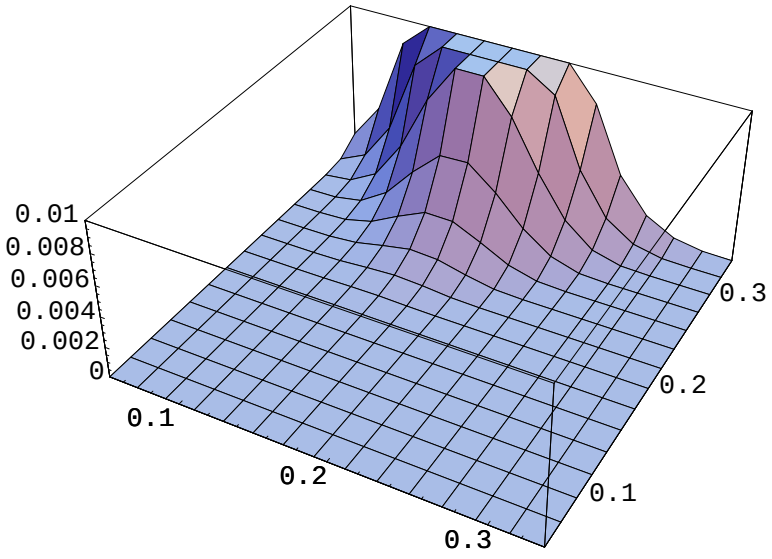

Figure 3.2. $I_3(s,t)$ with $0 \leq s \leq 1/3$, $0 \leq t \leq 1/3$

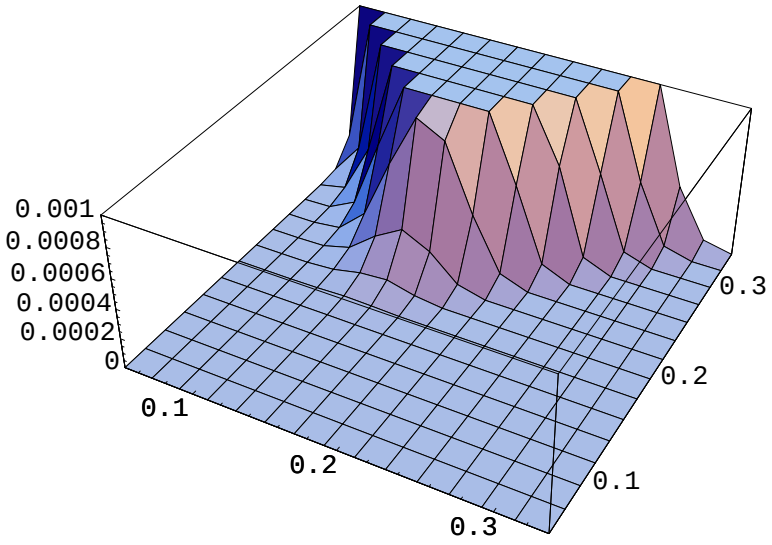

Figure 3.3. $I_3(s,t)$ with $0 \leq s \leq 1/3$, $0 \leq t \leq 1/3$

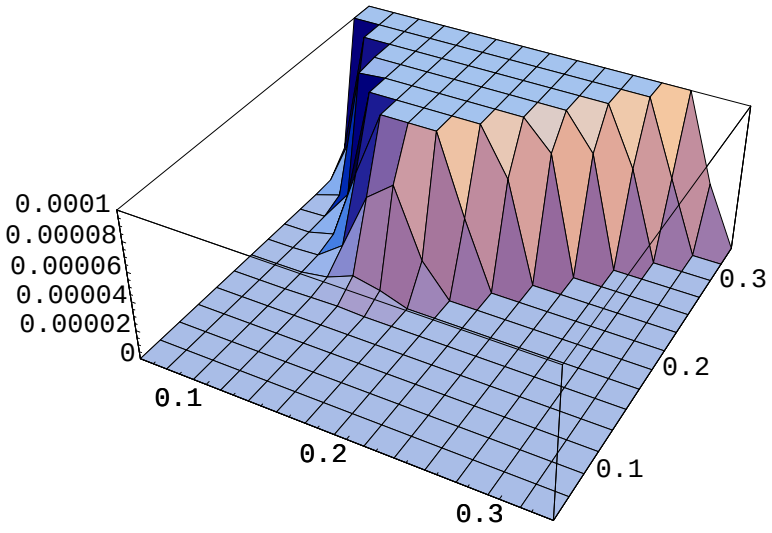

Figure 3.4. $I_3(s,t)$ with $0 \leq s \leq 1/3$, $0 \leq t \leq 1/3$

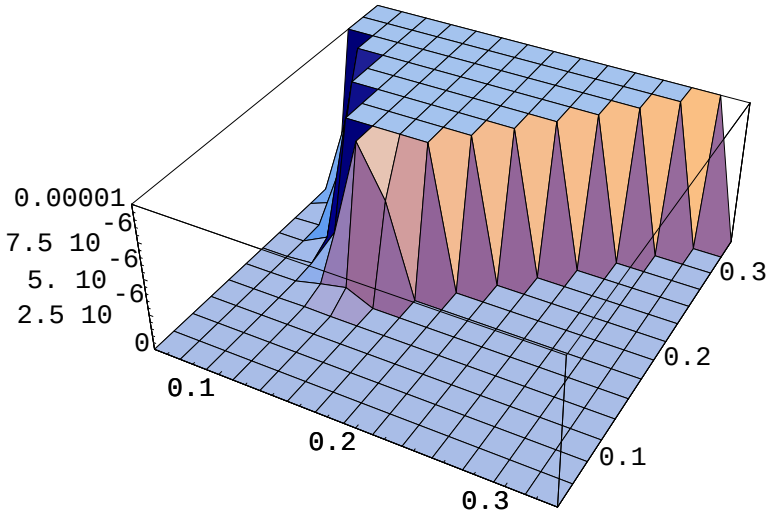

Figure 3.5. $I_3(s,t)$ with $0 \leq s \leq 1/3$, $0 \leq t \leq 1/3$

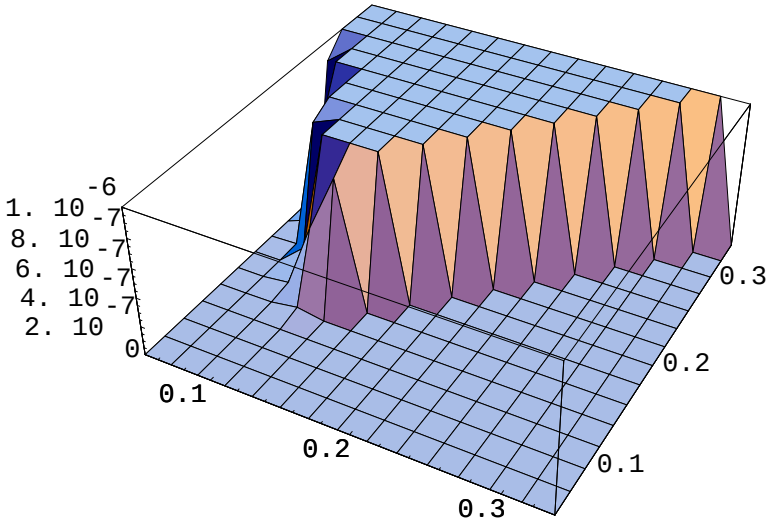

Figure 3.6. $I_3(s,t)$ with $0 \leq s \leq 1/3$, $0 \leq t \leq 1/3$

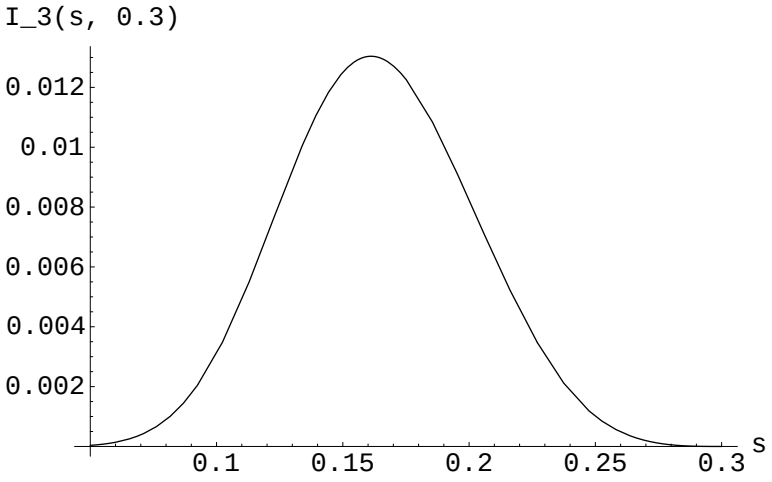

Figure 3.7. $I_3(s,0.3)$ with $0 \leq s \leq 0.3$

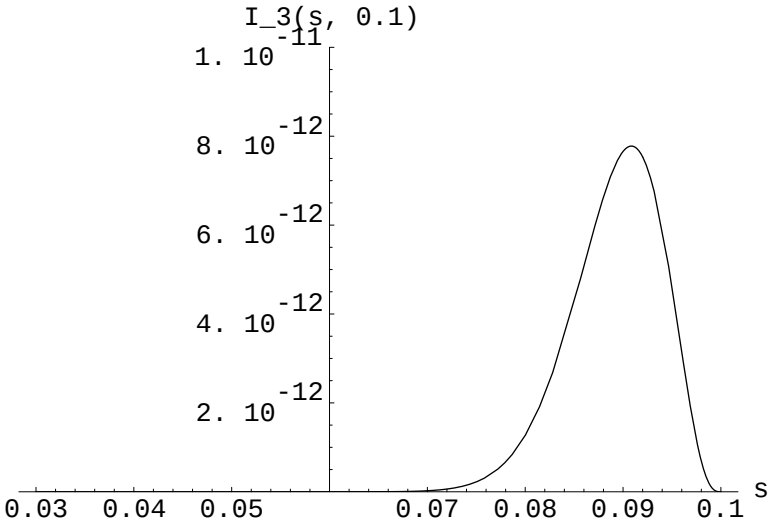

Figure 3.8. $I_3(s, 0.1)$ with $0.06 \leq s \leq 0.1$

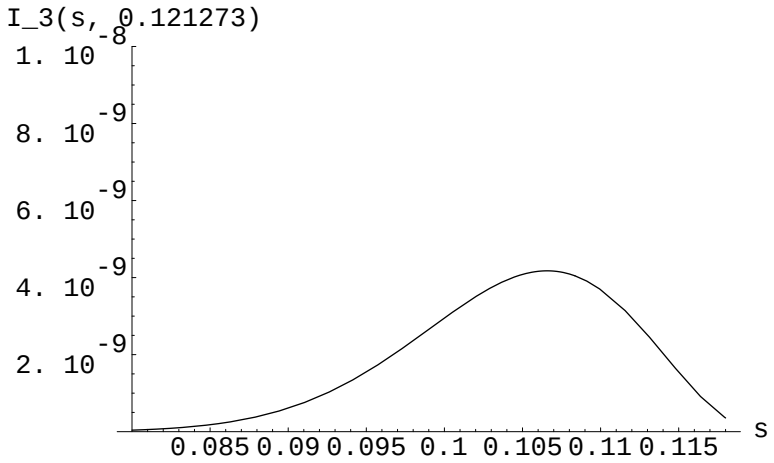

Figure 3.9. $I_3(s, 0.121273)$ with $0.08 \leq s \leq 0.12$

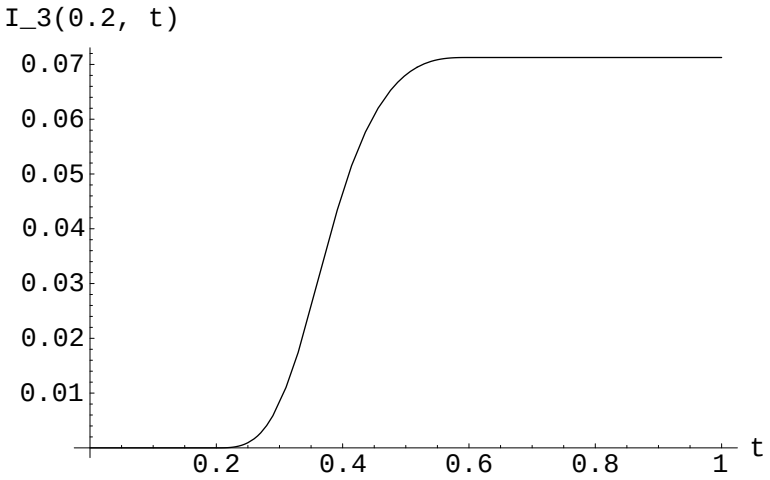

Figure 3.10. $I_3(0.2, t)$ with $0 \leq t \leq 1$

that we are sieving have roughly the same size as $x = M * \sqrt{n} = 2.93322 \times 10^{74}$, where $M = 17158000/2$ is half of the length of the sieve interval. The corresponding s is about 0.0971495, and t is about 0.121273. If we fix t at this value, and draw $I_3(s, t)$ as a function of s, we have Figure 3.9. It is quite clear from the figure that $s = 0.0971495$ is not the best choice, at least if we want to get a better yield of relations with 3 large primes. When s is about 0.107, we have the highest $I_3(s, t)$ value. One might argue that this could result in a loss in the 2-LP and 1-LP relations, (full relations are actually going to gain a lot from this increase of s), however, calculating $I_2(s, t)$ and $I_1(s, t)$ at these values, we see that we actually should have more 2-LP and 1-LP relations too. Another interesting reason that we might want to maximize the 3-LP yield is that, for 3-LP relations, we are expecting to see a sudden increase in the number of cycles as a function of the number of partial relations. Higher yield in the 3-LP relations will take us to this sudden increase quicker and thus reduce the time needed for sieving. Therefore, taking $s \approx 0.107$ should almost certainly produce more relations, full or partial, but that still does not mean a shorter running time, because one must also take into account the fact that, with a higher s, we have a larger factor base too, therefore, we will need more relations to begin with. A rough estimate

shows that the factor base would be about 4 to 5 times larger if we take s to around 0.107. The increase in the full relations are much more than that, in fact, we should have about 30 times the original number of full relations. For partial relations, we should be concerned about how many fundamental cycles they can produce instead of the actual number of relations. Even if we do not consider the sudden increase in the cycle number, because of the power law, the yield of cycles should still grow much faster than 4 or 5 times when the number of relations doubles, which is roughly the case for the 3-LP relation in this case. Therefore, we conclude that the actual parameters used in factoring 2,1606L.c135 are quite far from optimal.

From this, we propose the following parameter choice procedure for future factorizations with MPQS and NFS:

1. Determine the size of the numbers that we expect to be sieving. This will depend on the number we are trying to factor, which algorithm we use as well as how large the sieve region will be.

2. Determine the large prime bound L, that is, the parameter t. This should be chosen as large as possible given the available computing resources.

3. Choose s to maximize $I_k(s,t)$ for the moment, where k is the number of large primes that will be allowed in a partial relation. Then check the values of $I_j(s,t)$ for $j = 1, \ldots, k-1$ to make sure that this s is not too bad a choice for the fewer large prime partials.

4. With the above s, find the largest prime in the factor base and the size of the factor base. Determine whether the factor base is acceptable with available computing resources. If not, choose the maximum affordable s.

5. Finally, vary the parameters slightly and conduct sieving experiments to determine the best choice.

With the program we have for computing $I_3(s,t)$, we can do this for $k = 3$. We will need to have an effective method for computing $I_k(s,t)$ first before we can utilize this procedure in practice for $k \geq 4$.

3.4 How good are the approximations?

Since $G_k(s,t)$, $I_k(s,t)$ are defined as limit functions when integers tend to infinity, a natural question that one might ask is, how close are the approximations when used to predict the number of smooth, k-semismooth integers in a certain range? This section gives some evidence that we have reasonably good matches when the integers are relatively big.

We factored the integers in the interval $[10^{15} - 10^5, 10^{15} - 1]$ completely and compiled the tables below. Table 3.2 gives the total number of integers that have exactly 3 large prime factors between y and z for different choices of y, z. Table 3.3 gives $I_3(s,t) * 10^5$ with $s = \ln(y)/\ln(10^{15}), t = \ln(z)/\ln(10^{15})$, which is the expected number of such integers. As we can see, they are very close to each other as soon as the count gets reasonably big. The relative errors are mostly $< 10\%$.

Table 3.2
Number of integers in $[10^{15} - 10^5, 10^{15} - 1]$ with exactly 3 prime divisors between y and z and all other prime divisors $\leq y$

	z								
y	4000	6000	8000	10000	20000	40000	80000	100000	200000
2000	5	22	48	80	259	507	902	1066	1653
4000	-	1	8	18	92	230	469	566	926
6000	-	-	0	3	40	129	293	363	631
8000	-	-	-	0	17	73	189	244	447
10000	-	-	-	-	9	52	143	189	364
20000	-	-	-	-	-	8	34	57	139

Table 3.3
$I_3(s,t) * 10^5$ with $s = \ln(y)/\ln(10^{15}),\ t = \ln(z)/\ln(10^{15})$

	z								
y	4000	6000	8000	10000	20000	40000	80000	100000	200000
2000	7.28	29.31	59.34	93.41	272.50	571.2	991.24	1152.11	1714.64
4000	-	1.65	8.25	18.75	92.52	244.65	488.23	587.32	945.36
6000	-	-	0.57	3.1	36.3	127.95	295.07	366.66	631.7
8000	-	-	-	0.25	15.25	74.51	197.99	253.20	462.48
10000	-	-	-	-	6.37	45.94	140.60	185.18	356.51
20000	-	-	-	-	-	5.16	37.48	56.91	136.57

4. Smooth Integer Distribution in Short Intervals

4.1 Introduction

In this chapter, we will study a problem posed by Selfridge and Meyerowitz [25]. The problem is about a function f defined as follows:

Definition 4.1.1 *Given integers $k \geq 2$ and $n > 0$, let $f(n,k)$ be the smallest positive integer t so that there is a subset of k distinct integers in the interval $[n, n+t]$ whose product is a perfect square and both n and $n+t$ are included in the set. Define $f(n)$ to be the minimum of $f(n,2)$, $f(n,3)$, ...*

Problem: Find good estimates for $f(n)$.

For each $k \geq 2$ and $n > 0$, $f(n,k)$ is well-defined since there are k distinct integers $\geq n$, namely n, $4n$, $9n$, ..., $(k-1)^2 n$, $k^2 n$ if k is even, or n, $4n$, $9n$, ..., $(k-1)^2 n$, $k^2 n^2$ if k is odd, whose product is a square. So $f(n,k) \leq k^2 n$ if k is even and $f(n,k) \leq k^2 n^2$ if k is odd. Of course, this is a gross overestimate, $f(n,k)$ should have much smaller values most of the time. For example, $f(8,3) = 10$ since in the interval $[8,18]$, we have 3 integers, $8, 9, 18$, that form a square product and its easy to check that the length of the interval cannot be any shorter.

Selfridge and Meyerowitz have shown [25] that $\liminf_{n\to\infty} \frac{f(n,3)}{\sqrt[4]{n}}$, and therefore $\liminf_{n\to\infty} \frac{f(n)}{\sqrt[4]{n}}$ is bounded. We will briefly present their ideas here.

If we choose integer x such that $x^2 - 1$ is twice a square, then the following three integers have a square product:

$$\begin{aligned} 2(4x^4 - x^2 - 2)^2 &= 32x^8 - 16x^6 - 30x^4 + 8x^2 + 8, \\ 2(x^2-1)(x^2+1)(4x^2-1)^2 &= 32x^8 - 16x^6 - 30x^4 + 16x^2 - 2, \\ 2x^2(x^2+1)(4x^2-3)^2 &= 32x^8 - 16x^6 - 30x^4 + 18x^2. \end{aligned}$$

If we let $n_x = 2(4x^4 - x^2 - 2)^2 = 32x^8 - 16x^6 - 30x^4 + 8x^2 + 8$ and $z_x = 10x^2 - 8$, then we can see that $f(n_x, 3)$ must be $\leq z_x$ since in the interval $[n_x, n_x + z_x]$, we have those three integers above that have a perfect square product. Notice that on the right hand side, all three expressions agree on the high order terms and differ only on the x^2 and constant terms. So the difference between the last and the first integer $= z_x \sim c\sqrt[4]{n_x}$. So we have $f(n_x, 3) \leq z_x \sim c\sqrt[4]{n_x}$ and

$$c = \frac{10}{\sqrt[4]{32}} \approx 4.20\,.$$

Now to show that $\liminf_{n\to\infty} \frac{f(n,3)}{\sqrt[4]{n}}$ is bounded, all we need to prove is that infinitely many such n_x, or x with $x^2 - 1$ being twice a square, exist. So now the question is how many integer solutions $x^2 - 1 = 2y^2$ has. This is exactly the Pell equation $x^2 - D\,y^2 = 1$ with $D = 2$, and it is well known that infinitely many integer solutions exist. See, for example, [19].

This proof is based on explicit construction of 3 integers in an interval of length that is only the fourth root of the size of the integers. Since the definition of f places no restriction on the number of integers to form the product, it's reasonable to hope for a better estimate of f using possibly more integers.

To use the results on smooth integer distribution in short intervals, we modify the definition of f to remove the constraint that the beginning and the end of the interval must be included in the subset of integers. Also we extend the domain of the function to all real numbers $x > 0$. We have

Definition 4.1.2 *For any real number $x > 0$, let $g(x)$ be the least positive real number z so that there exists a subset of at least 2 distinct integers in $[x, x + z]$ whose product is a perfect square.*

Clearly, for integer n, $g(n) \leq f(n)$. We will first obtain good estimates for $g(x)$, and then show that similar results can be extended to $f(x)$.

In the next section, we will first give results on the smooth integer distribution in short intervals. The length of the "short interval" for which we can derive a good

approximation of the number of smooth integers greatly impacts the growth rate of the function g. The shorter such an interval is, the better estimate we can get for g. The first result we will prove is that for any $\alpha > 0$, $g(x) \leq x^{\alpha}$ for "most" large x, in a sense which will be defined later. Then, with a little more complication, we further improve this to get that $g(x) \leq \exp((\ln(2\,x))^{1/6} + (\ln(2\,x))^{5/6+\epsilon})$ for "most" large x.

4.2 Estimates for $g(x)$ and $f(x)$.

The reason that we can use smooth integer distribution to estimate $g(x)$ better is because we can view the process of finding an integer subset of $[x, x+z]$ to form a square as a sieving process, just like in the Quadratic Sieve and Number Field Sieve algorithms. If we can find enough y-smooth integers in the interval, then we can find dependencies modulo 2 among the exponent vectors of all the y-smooth numbers we have, and thus get a square. So we need to estimate the number of y-smooth integers available in the interval $[x, x+z]$. Hildebrand proved the following [11]:

Theorem 4.2.1 *If* $y \geq 2, \quad \exp((\ln\ln x)^{5/3+\epsilon}) \leq y \leq x$ *and* $xy^{-5/12} \leq z \leq x$, *then we have*

$$\Psi(x+z,y) - \Psi(x,y) = z\rho(u)\left\{1 + O\left(\frac{\ln(u+1)}{\ln y}\right)\right\},$$

where $u = (\ln x)/(\ln y)$.

In other words, the number of y-smooth integers in the interval $[x, x+z]$ can be approximated by $z\rho(u)$ asymptotically, provided that y, z are in the given range. However the restriction on y, z makes this theorem useless for our purpose here. We hope to find a relatively short interval, $[x, x+z]$, preferably of length subexponential in $\ln x$, with enough y-smooth integers, but in this theorem, z is $x^{7/12}$ at its best (smallest). To relax the condition on y, z, one must turn to weaker results. Instead of asking for estimates that hold for all x, we now ask for estimates that hold for "almost all" x. Then the ranges for y and z can be improved a lot. The first such result was obtained by Friedlander [8]. Later he and Lagarias improved his results to obtain the following theorems [9]:

Theorem 4.2.2 *For any fixed $\epsilon > 0$, $0 < \beta \leq \alpha \leq 1$, and for all sufficiently large X, the estimate*

$$\Psi(x + x^\beta, x^\alpha) - \Psi(x, x^\alpha) \geq \frac{1}{64}\beta\rho(1/\alpha)x^\beta \tag{4.1}$$

holds for all $x \in [1, X]$ with the exception of a set of measure (usual Lesbegue measure) bounded by $c_{\epsilon,\alpha,\beta} X \exp(-(\ln X)^{1/3-\epsilon})$, where $c_{\epsilon,\alpha,\beta}$ is a constant depending only on ϵ, α and β.

Theorem 4.2.3 *For any fixed $\epsilon > 0$, for all sufficiently large X, and for y and z satisfying*

$$\exp((\ln X)^{5/6+\epsilon}) \leq y \leq X, \quad y\exp((\ln X)^{1/6}) \leq z \leq X, \tag{4.2}$$

the estimate

$$\Psi(x + z, y) - \Psi(x, y) \geq \frac{1}{16}\rho(\frac{\ln X}{\ln y})z \tag{4.3}$$

holds for all $x \in [1, X]$ with the exception of a set of measure (usual Lesbegue measure) bounded by $c_\epsilon X \exp(-\frac{1}{2}(\ln X)^{1/6})$, where c_ϵ is a constant depending only on ϵ.

With the above results on the smooth integer distribution in a short interval, we can get much better results concerning the growth rate of the function $g(x)$ defined in the previous section. First, we have the following corollary of Theorem 4.2.2

Corollary 4.2.4 *For any fixed $\epsilon > 0$, $0 < \alpha \leq 1$, and all sufficiently large X,*

$$\Psi(x + x^\alpha, x^\alpha) - \Psi(x, x^\alpha) \geq \frac{1}{64}\alpha\rho(1/\alpha)x^\alpha \tag{4.4}$$

holds for all $x \in [X/2, X]$ with the exception of a set of measure (usual Lesbegue measure) bounded by $c_{\epsilon,\alpha} X \exp(-(\ln X)^{1/3-\epsilon})$, where $c_{\epsilon,\alpha}$ is a constant depending only on ϵ and α.

Remark: we want $x \in [X/2, X]$ because we need x to tend to ∞ with X. The $X/2$ can be replaced by any function of X that tends to ∞ when $X \to \infty$.

Proof Let $\alpha = \beta$ in Theorem 4.2.2. ■

Now we are ready to prove

Theorem 4.2.5 *For any fixed $\epsilon > 0$, $0 < \alpha \leq 1$, $g(x)$ as defined in the previous section, and all sufficiently large X, $g(x) \leq x^\alpha$ for all $x \in [X/2, X]$ with the exception of a set of measure (usual Lesbegue measure) bounded by $c_{\epsilon,\alpha} X \exp(-(\ln X)^{1/3-\epsilon})$, where $c_{\epsilon,\alpha}$ is a constant depending only on ϵ and α.*

Proof For those $x \in [X/2, X]$ satisfying (4.4), we have the number of x^α-smooth integers in the interval $[x, x + x^\alpha]$ is $> \pi(x^\alpha) \sim x^\alpha / \ln(x^\alpha)$. Using the same linear algebra technique used in QS and NFS, one can see that there is a subset of those x^α-smooth integers that have a perfect square product. Therefore $g(x) \leq x^\alpha$. So for large X, $g(x) \leq x^\alpha$ can only fail to hold when (4.4) fails to hold. This proves the theorem. ■

In particular, for any $0 < \alpha \leq 1$, we have infinitely many x such that $g(x) \leq x^\alpha$. Noticing that for any $x > 0$, $g(\lceil x \rceil) = g(x)$, we have

Corollary 4.2.6 *For any $0 < \alpha \leq 1$, g defined as in the previous section,*

$$\liminf_{n \to \infty, n \in \mathbf{Z}} \frac{g(n)}{n^\alpha} \leq 1$$

Theorem 4.2.3 shortens the interval to subexponential length and therefore enables us to get an even better estimate for g, with some complications. Since we want the interval to be as short as possible, we will take y and z to be at their minimum values in the range (4.2).

Corollary 4.2.7 *For any fixed $\epsilon > 0$, for all sufficiently large X, and*

$$\begin{aligned} y &= \exp((\ln X)^{5/6+\epsilon}), \\ z &= y \exp((\ln X)^{1/6}) = \exp((\ln X)^{1/6} + (\ln X)^{5/6+\epsilon}), \end{aligned}$$

the estimate (4.3) holds for all $x \in [X/2, X]$ with the exception of a set of measure bounded by $c_\epsilon X \exp(-\frac{1}{2}(\ln X)^{1/6})$, where c_ϵ is a constant depending only on ϵ.

Proof Let $y = \exp((\ln X)^{5/6+\epsilon})$, $z = y\exp((\ln X)^{1/6}) = \exp((\ln X)^{1/6} + (\ln X)^{5/6+\epsilon})$ in Theorem 4.2.3. ■

For those x satisfying (4.3), the number of y-smooth integers in the interval $[x, x+z]$ is big enough to enable us to construct a perfect square, so we have

Theorem 4.2.8 *For any fixed $\epsilon > 0$, $g(x)$ as defined in the previous section, and all sufficiently large X,*

$$g(x) \leq \exp((\ln(2\,x))^{1/6} + (\ln(2\,x))^{5/6+\epsilon})$$

for all $x \in [X/2, X]$ with the exception of a set of measure (usual Lesbegue measure) bounded by $c_\epsilon X \exp(-\frac{1}{2}(\ln X)^{1/6})$, where c_ϵ is a constant depending only on ϵ.

Proof Let y, z be defined as in Corollary 4.2.7. We only need to show that for all sufficiently large X and $x \in [X/2, X]$ satisfying (4.3),

$$g(x) \leq z \leq \exp((\ln(2\,x))^{1/6} + (\ln(2\,x))^{5/6+\epsilon}).$$

Using the same argument as Theorem 4.2.5, we see that it suffices to prove

$$\Psi(x+z, y) - \Psi(x, y) > \pi(y),$$

with y and z defined as in Corollary 4.2.7.

By (4.3),

$$\Psi(x+z, y) - \Psi(x, y) \geq \frac{1}{16}\rho(\frac{\ln X}{\ln y})z = \frac{1}{16}\rho((\ln X)^{1/6-\epsilon})z.$$

Taking logarithms on both sides and using the fact that $\ln\rho(u) = -(u + o(1))\ln u$ as $u \to \infty$ (see [12] for a proof of this), we have

$$
\begin{aligned}
& \ln(\Psi(x+z,y)-\Psi(x,y)) \\
\geq\ & \ln z-((\ln X)^{1/6-\epsilon}+o(1))\ln((\ln X)^{1/6-\epsilon})-\ln(16) \\
=\ & (\ln X)^{1/6}+(\ln X)^{5/6+\epsilon}-((\ln X)^{1/6-\epsilon}+o(1))\ln((\ln X)^{1/6-\epsilon})-\ln(16) \\
>\ & (\ln X)^{5/6+\epsilon}+(\ln X)^{1/6}-2\,(\ln X)^{1/6-\epsilon}\ln((\ln X)^{1/6-\epsilon}) \\
=\ & (\ln X)^{5/6+\epsilon}+(\ln X)^{1/6}\left(1-\frac{2\,\ln\ln X}{(1/6-\epsilon)\,(\ln X)^{\epsilon}}\right) \\
>\ & (\ln X)^{5/6+\epsilon} \\
=\ & \ln y \\
>\ & \ln(\pi(y)).
\end{aligned}
$$

This shows that $\Psi(x+z,y)-\Psi(x,y)>\pi(y)$, and concludes the proof. ■

Similar to Corollary 4.2.6, we have

Corollary 4.2.9 *For any $\epsilon>0$, and g defined as in the previous section,*

$$\liminf_{n\to\infty,n\in\mathbf{Z}}\frac{g(n)}{\exp((\ln(2\,n))^{1/6}+(\ln(2\,n))^{5/6+\epsilon})}\leq 1.$$

To extend the results to $f(n)$, we use the fact that for any integer $n>0$, $\exists n'\geq n$ such that $f(n')\leq g(n)$. To see this, let $g(n)=z$. By definition of g, we can find $k\geq 2$ integers $n_1<n_2<\cdots<n_k$ in the interval $[n,n+z]$ whose product is a square. Also notice that n_k must be equal to $n+g(n)=n+z$ (otherwise, we can find smaller values for z, contradicting with the definition of g). So letting $n'=n_1\geq n$, we have a subset of integers in $[n',n+z]$, including n' and $n+z$, whose product is a square. By the definition of f, $f(n')\leq n+z-n'\leq z=g(n)$.

Using this fact, we see that Corollaries 4.2.6 and 4.2.9 with g replaced by f hold too. Therefore, $f(n)$ takes subexponential values (in $\ln(n)$) for infinitely many n. This improves the result of Selfridge and Meyerowitz significantly.

5. Summary

We have generalized the Dickman function to get an asymptotic probability estimate for k-semismooth integers. Our result agrees with previous work on the problem for $k = 1$ and $k = 2$. Properties of the new function were discussed. We also gave a method for calculating this function at $k = 3$. It is still an interesting problem to find an effective method for computing the new function at $k \geq 4$. Numerical results (at $k = 3$) were given and applied to the parameter choice problem of integer factorization algorithms like MPQS and NFS with 3 large primes. Experiments are needed to check the effectiveness of our proposed method for parameter choice.

The smooth integer distribution in a short interval gave a nice improvement for the estimate of the square product function of Selfridge and Meyerowitz.

LIST OF REFERENCES

LIST OF REFERENCES

[1] D. Atkins, M. Graff, A. K. Lenstra, and P. C. Leyland. THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE. In *Advances in Cryptology - Asiacrypt'94*, Lecture Notes in Computer Science 917, pages 265–277. Springer-Verlag, 1995.

[2] E. Bach and R. Peralta. Asymptotic semi-smoothness probabilities. *Math. Comp.*, 65:1701–1715, 1996.

[3] J. P. Buhler, H. W. Lenstra, Jr., and C. Pomerance. Factoring integers with the number field sieve. In *The Development of the Number Field Sieve*, Lecture Notes in Mathematics 1554, pages 50–94. Springer-Verlag, 1993.

[4] B. Carrier and S. S. Wagstaff, Jr. Implementing the hypercube quadratic sieve with two large primes. Technical report, Purdue University, 2001. CERIAS TR 2001-45, https://www.cerias.purdue.edu/papers/archive/2001-45.ps.

[5] S. Cavallar, B. Dodson, A. Lenstra, P. Leyland, W. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, and P. Zimmermann. Factorization of RSA-140 using the number field sieve. In K. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology - ASIACRYPT '99*, Lecture Notes in Computer Science 1716. Springer-Verlag, 1999.

[6] K. Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Ark. Mat. Astr. Fys.*, 22:1–14, 1930.

[7] B. Dodson and A. K. Lenstra. NFS with four large primes: An explosive experiment. In D. Coppersmith, editor, *Advances in Cryptology, CRYPTO '95*, Lecture Notes in Computer Science, 963, pages 372–385. Springer-Verlag, 1995.

[8] J. B. Friedlander. Large prime factors in small intervals. In G. Halász, editor, *Topics in classical number theory*, Colloquia mathematica Societatis János Bolyai 34, pages 511–518. North-Holland, Amsterdam; New York, 1984.

[9] J. B. Friedlander and J. C. Lagarias. On the distribution in short intervals of integers having no large prime factor. *J. Number Theory*, 25:249–273, 1987.

[10] C. F. Gauss. *Disquisitiones Arithmeticae*, pages 396–397. Yale University Press, English edition, 1966. Translated by Arthur A. Clarke, S. J.

[11] A. Hildebrand. On the number of positive integers $\leq x$ and free of prime factors $> y$. *J. of Number Theory*, 22(3):289–307, 1986.

[12] A. Hildebrand. Integers without large prime factors. *Journal de Théorie des Nombres de Bordeaux*, 5:411–484, 1993.

[13] D. Knuth and L. Trabb Pardo. Analysis of a simple factorization algorithm. *Theoretical Computer Science*, 3:321–348, 1976.

[14] R. Lambert. *Computational Aspects of Discrete Logarithms.* PhD thesis, University of Waterloo, 1996.

[15] A. K. Lenstra and H. W. Lenstra, Jr., editors. *The development of the number field sieve*. Lecture Notes in Mathematics 1554. Springer-Verlag, 1993.

[16] A. K. Lenstra and M. S. Manasse. Factoring with two large primes. *Math. Comp.*, 63:785–798, 1994.

[17] P. Leyland, A. K. Lenstra, B. Dodson, A. Muffett, and S. Wagstaff. MPQS with three large primes. In *Proceedings ANTS 2002*, Lecture Notes in Computer Science 2369, pages 448–462, 2002.

[18] G. Marsaglia, A. Zaman, and J. C. W. Marsaglia. Numerical solution of some classical differential-difference equations. *Math. Comp.*, 53:191–201, 1989.

[19] T. Nagell. *Introduction to Number Theory*, pages 195–212. Wiley, New York, 1951.

[20] K. Norton. Numbers with small prime factors and the least kth power non-residue. *Memoirs of the Amer. Math. Soc.*, 106:9–27, 1971.

[21] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In H. W. Lenstra, Jr. and R. Tijdeman, editors, *Computational Methods in Number Theory, Part I*, pages 89–139. Mathematisch Centrum Tract 154, Amsterdam, 1982.

[22] C. Pomerance. The quadratic sieve factoring algorithm. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances of Cryptology, Proc. of Eurocrypt 84*, Lecture Notes in Computer Science 209, pages 169–182. Springer-Verlag, 1985.

[23] H. Riesel. *Prime Numbers and Computer Methods for Factorization*. Birkhäuser, second edition, 1994.

[24] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM*, 21:120–126, 1978.

[25] J. L. Selfridge and A. Meyerowitz. The $\sqrt[4]{n}$ construction. Manuscript.

[26] R. D. Silverman. The multiple polynomial quadratic sieve. *Math. Comp.*, 48:329–339, 1987.

APPENDIX

APPENDIX

MATHEMATICA PROGRAM TO COMPUTE $I_3(s,t)$

```
(* Clear all the symbols that we need *)
ClearAll[numcoeffrho, c, intd1, intd2, intd3, intd4, I3, sigma3];

(* number of coefficients to use in the Taylor expansion of rho
   on [m-1, m] *)
numcoeffrho = 22;

(* Function to compute c_i^{(m)} *)
c[m_, i_] := c[m,i] = Module[{j, value},

If[m==1,
        If[ i==0, value=1, value=0],
        If[ m ==2,
                If[i==0, value = N[1-Log[2]], value = N[1/(i*2^i)]],
                value = 0;
                If[ i == 0,
                      Do[value=value+N[c[m,j]/((m-1)(j+1))],
                                        {j, 1, numcoeffrho}],
                      Do[value=value+N[c[m-1,j]/(i*m^(i-j))],
                                        {j, 0, i-1}]
                  ]
         ]
  ];
```

```
value
] (* End Module for c[m,i] *)


(* Function to compute the integral over the region D_1 *)
intd1[s_, t_] := Module[ {F1, u1, u2, u3, tmp, low, high, a, b,
                          m, i, sum},
(* F1,  produced by Mathematica *)
F1 = (Log[2]^2 + 2*Log[2]*Log[(-s + u1)/(2*s)] + 2*Log[(-s + u1)/(2*s)]*
      Log[(s + u1)/(2*s)] + 2*PolyLog[2, (s - u1)/(2*s)] +
      2*PolyLog[2, s/(-s + u1)])/(2*u1);

low = 3*s;
high = 2*s + t;
If[high > 1, high = 1];
a = low;
m = Ceiling[(1-a)/s];
b = 1 - (m-1)*s;
If[ b > high,  b = high ];

sum = 0;
While[ a < high - 10^(-5),
        For[i = 0, i < numcoeffrho, i++,
                tmp = (m - 1/s + u1/s)^i * F1;
                sum = sum + c[m, i] * NIntegrate[tmp, {u1, a, b}];
        ];
        a = b;
        b = b + s;
        If[ b > high, b = high ];
```

```
        m = m - 1;
];  (* End While *)
sum
];  (* End Module for intd1[s,t] *)


(* Function to compute the integral over the region D_2 *)
intd2[s_, t_] := Module[ {F2, u1, u2, u3, tmp, low, high, a, b,
                          m, i, sum},
(* F2,  produced by Mathematica *)
F2 =
  (-Pi^2/12 + Log[2]^2/2 + Log[-((s+t-u1)/s)]*Log[t/(-s+u1)] +
  Log[2]*Log[(-s+u1)/(2*s)] + Log[(-s+u1)/(2*s)]*Log[(s+u1)/(2*s)] -
  Log[-((s+t-u1)/s)]*Log[(-t+u1)/s] + PolyLog[2, (s-u1)/(2*s)] -
  PolyLog[2, (s+t-u1)/s] + PolyLog[2, (s+t-u1)/(s-u1)])/u1 ;

low = 2*s+t;
high = s+ 2*t;  If[high > 1, high = 1];
a = low;
m = Ceiling[(1-a)/s];
b = 1 - (m-1)*s; If[ b > high,  b = high ];

sum = 0;
While[ a < high - 10^(-5),
        For[i = 0, i < numcoeffrho, i++,
                tmp = (m - 1/s + u1/s)^i * F2;
                sum = sum + c[m, i] * NIntegrate[tmp, {u1, a, b}];
        ];
        a = b;
```

```
        b = b + s;
        If[ b > high, b = high ];
        m = m - 1;
];  (* End While *)
sum
];  (* End Module for intd2[s,t] *)


(* Function to compute the integral over the region D_3 *)
intd3[s_, t_] := Module[ {ex, F3, u1, u2, u3, tmp, low, high, a, b,
                          m, i, sum},
(* F3,  produced by Mathematica  and further simplified by hand
   using property of the dilogarithm to remove the imaginary part,
   which should not appear in the first place *)
F3 = (Pi^2/6  - Log[4/3]^2/2 - Log[s]*Log[1 - s/u1] -
  Log[3/2]*Log[u1/3] - Log[u1/3]^2/2 +
  Log[(-s + u1)/2]^2/2 + Log[s]*Log[(s + u1)/2] -
  Log[(s/u1)]*Log[u1/2] - Log[(-s + u1)/2]*Log[(s + u1)/2] +
  Log[(-s + u1)/(2*u1)]*Log[(s + u1)/2] + PolyLog[2, 1/3] -
  PolyLog[2, 2/3] - PolyLog[2, 3/4] - PolyLog[2, s/u1] +
  PolyLog[2, (s + u1)/(2*u1)] + PolyLog[2, -s/u1])/u1 ;
low = 3*s;
high = s + 2*t;   If[high > 1, high = 1];
a = low;
m = Ceiling[(1-a)/s];
b = 1 - (m-1)*s;  If[ b > high,  b = high ];
sum = 0;
While[ a < high - 10^(-5),
        For[i = 0, i < numcoeffrho, i++,
```

```
                tmp = (m - 1/s + u1/s)^i * F3;
                sum = sum + c[m, i] * NIntegrate[tmp, {u1, a, b}];
        ];
        a = b; b = b + s;   If[ b > high, b = high ];
        m = m - 1;
]; (* End While *)
sum
];  (* End Module for intd3[s,t] *)


(* Function to compute the integral over the region D_4 *)
intd4[s_, t_] := Module[ { F4, u1, u2, u3, tmp, low, high, a, b,
                           m, i, sum},
(* F4,  produced by Mathematica  and further simplified by hand
   using property of the dilogarithm to remove the imaginary part,
   which should not appear in the first place *)
F4 = (Pi^2/6  - Log[4/3]^2/2 + Log[t]^2/2 - Log[3/2]*Log[u1/3] -
  Log[u1/3]^2/2 - Log[(2*t)/u1]*Log[-2*t + u1] -
  Log[t]*Log[-t + u1] + Log[1 - (2*t)/u1]*Log[2/u1] +
  Log[t/u1]*Log[-t + u1] + Log[-2*t + u1]*Log[-t + u1] +
  PolyLog[2, 1/3] - PolyLog[2, 2/3] - PolyLog[2, 3/4] -
  PolyLog[2, 1 - (2*t)/u1] + PolyLog[2, (2*t)/u1 - 1] +
  PolyLog[2, 1 - t/u1])/u1 ;
low = s + 2*t;
high = 3*t;  If[high > 1, high = 1];
a = low;
m = Ceiling[(1-a)/s];
b = 1 - (m-1)*s;  If[ b > high,  b = high ];
sum = 0;
```

```
While[ a < high - 10^(-5),
        For[i = 0, i < numcoeffrho, i++,
                tmp = (m - 1/s + u1/s)^i * F4;
                sum = sum + c[m, i] * NIntegrate[tmp, {u1, a, b}];
        ];
        a = b; b = b + s;  If[ b > high, b = high ];
        m = m - 1;
];      (* End While *)
sum
];  (* End Module for intd4[s,t] *)


(* Function to compute I_3(s,t) *)
I3[s_, t_] := Module[ {value},
If[s >= t,
        value = 0,
        value = intd1[s,t] + intd2[s,t] + intd3[s,t] + intd4[s,t];
];
value
];

(* Define sigma3(u,v) *)
sigma3[u_, v_] := I3[1/u, 1/v];
```

VITA

VITA

Chaogui Zhang, born on October 27, 1974 in Yongchuan, Chongqing, China.

Starting from his high school years, he found mathematics interesting, and went to University of Science and Technology of China in 1993, majoring in mathematics. He got his bachelor's degree in July of 1997, and started to pursue his Ph.D in mathematics at Purdue University the same year.

While at Purdue University studying for his Ph.D in mathematics, he earned a Master's degree in Computer Science (May, 2002).