SQUARE FORM FACTORIZATION, II

CLINTON BRADFORD AND SAMUEL S. WAGSTAFF, JR.

Abstract. We propose a new subexponential time integer factoring algorithm called SQUFOF2, based on ideas of D. Shanks and R. de Vogelaere. It begins by using a sieve like that in the multiple polynomial Quadratic Sieve to construct a square value of a binary quadratic form. It uses this value to produce a square form. Then it factors the integer N as the original SQUFOF does by taking an inverse square root and following a nonprincipal cycle to a symmetry point. This marriage with the Quadratic Sieve transforms SQUFOF from a $O(N^{1/4})$ algorithm into one with subexponential time. On the way we prove new facts about infrastructure distance, which is used in the time complexity analysis.

1. Introduction

About forty-five years ago, Daniel Shanks invented an integer factoring algorithm he called SQUFOF for SQUare FOrms Factoring. The method factors N in expected time $O(N^{1/4})$ with a short, simple algorithm. He explained the algorithm to a few people but published nothing about it. A manuscript [20] was found in his office after his death. A paper [10] completed the heuristic argument Shanks began in [20], but did not explore all of the ideas in that work. The present work investigates another idea from [20], one that de Vogelaere raised when Shanks lectured on SQUFOF in [19]. Starting from de Vogelaere's idea, we were led to a variation of SQUFOF, called SQUFOF2, that factors N in expected subexponential time $O(\exp(1.02\sqrt{\log N \log\log N}))$.

Both SQUFOF and SQUFOF2 require the theory of real quadratic fields, including the concept of the infrastructure of such fields, to explain the running time of the algorithms.

The next section introduces the parts of the theory of binary quadratic forms, developed by Gauss [9], that we need. The following section treats the infrastructure distance and requires a bit of theory of ideals in a quadratic number field. After we recall how SQUFOF works, we present our new algorithm and give several examples. Finally we give the proof of the expected running time for SQUFOF2. Let $L(N) = \exp(\sqrt{\log N \log\log N})$. We will show that the expected running time is $L(N)^{1.02+o(1)}$.

The authors thank D Buell, MJ Jacobson Jr, HW Lenstra Jr, H Montgomery and C Pomerance for valuable correspondence related to this work.

2. Binary quadratic forms

We follow Buell [3] in this treatment of quadratic forms. Let $F(X, Y) = aX^2 + bXY + cY^2$ be a binary quadratic form in the variables X, Y. The coefficients a, b, c will always be integers. The *discriminant* of F is $\Delta = b^2 - 4ac$. We are concerned only with *indefinite* forms, those with positive discriminant. Sometimes we will write (a, b, c) for the form F.

2010 *Mathematics Subject Classification.* Primary 11A51; Secondary 11E16, 11R11, 11Y05.

Key words and phrases. Integer factorization, binary quadratic form, infrastructure distance.

This work is based on the Ph.D. thesis of the first author, supervised by the second author.

SSW's work was supported by the CERIAS Center at Purdue University.

2.1. **Equivalent forms.** A form F *represents* an integer m if there are integers x and y so that $F(x, y) = m$. The representation is *primitive* if $\gcd(x, y) = 1$.

The classical modular group $\mathrm{SL}_2(\mathbb{Z})$ acts on the set of binary quadratic forms by

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} F(X, Y) = F(\alpha X + \beta Y, \gamma X + \delta Y).$$

Two forms are *equivalent* if they are in the same orbit of $\mathrm{SL}_2(\mathbb{Z})$. Equivalent forms have the same discriminant and represent the same integers. The number of equivalence classes of forms with given discriminant is a finite number h.

Each equivalence class of forms contains a set of canonical representatives called reduced forms. The (indefinite) form (a, b, c) is *reduced* if $\left|\sqrt{\Delta} - 2|c|\right| < b < \sqrt{\Delta}$. The set $\mathcal{R}$ of reduced forms (a, b, c) with given discriminant Δ is finite because $|a| < \sqrt{\Delta}$.

2.2. **Reduction of forms.** For any indefinite form (a, b, c) with $ac \neq 0$, define the *standard reduction operator* ρ by

$$\rho((a, b, c)) = \left(c, r(-b, c), (r(-b, c)^2 - \Delta)/(4c)\right),$$

where $r(-b, c)$ is the unique integer r with $r + b \equiv 0 \pmod{2c}$ and

$$\begin{aligned} -|c| < r \leq |c| \quad &\text{if} \quad \sqrt{\Delta} < |c|, \\ \sqrt{\Delta} - 2|c| < r < \sqrt{\Delta} \quad &\text{if} \quad |c| < \sqrt{\Delta}. \end{aligned}$$

Write $\rho^n(F)$ for the result of n applications of ρ to F. Note that if F has discriminant Δ, then $\rho(F)$ also has discriminant Δ. If F is reduced, then so is $\rho(F)$. If F is not reduced, then $\rho^n(F)$ is reduced for some

$$n \leq 2 + \left\lceil \frac{\log |c|}{\sqrt{\Delta}} \right\rceil \tag{1}$$

according to Proposition 5.6.6 of Cohen [5].

The unique reduced form $F_0 = (1, b, c)$ is the *principal form*. It has $b > 0$ and $c < 0$.

One can prove that ρ is a permutation of $\mathcal{R}$. The inverse of ρ is $\rho^{-1} = \tau\rho\tau$, where $\tau((a, b, c)) = (c, b, a)$. A *cycle* of $\mathcal{R}$ is an orbit of $\mathcal{R}$ under the action of powers of ρ. Since the leading coefficients alternate in sign as ρ is applied, every cycle contains an even number of reduced forms. The *principal cycle* $\mathcal{P}$ is the one containing the principal form.

2.3. **Composition of forms.** There is a multiplication operation called *composition* defined on forms of a fixed discriminant. We do not define it here because it is complicated and we do not actually need it. See [3], [10], [14], [18] or [21] for the definition. Write FG for the composition of forms F and G. Composition is commutative and the principal form F_0 is a neutral element. Every form has an inverse. However, composition is not associative. Gauss [9] proved that if forms F and F' are equivalent and if G and G' are equivalent, then FG and $F'G'$ are equivalent. Hence one can define composition of equivalence classes of forms of a given discriminant. This operation makes the set $\mathcal{C}$ of equivalence classes into a group called the *class group*.

The only composition of forms we need for the new algorithm is F_0 with itself, and $F_0F_0 = F_0 = F_0^{-1}$.

Suppose F and G are forms and that $H = FG$ is their composition. Gauss proved that if x_1, y_1, x_2 and y_2 are integers, then there exist integers x_3, y_3 so that $H(x_3, y_3) = F(x_1, y_1) \cdot G(x_2, y_2)$ and gave formulas for x_3 and y_3 in terms of x_1, y_1, x_2, y_2 and the coefficients of

F and G. The modern version of these formulas appears on page 57 of [3] or as van der Poorten's [21] "magic matrix." We need only the formulas for $F = G = F_0 = (1, b, c)$; they are

$$x_3 = x_1x_2 - cy_1y_2 \tag{2}$$

$$y_3 = x_1y_2 + y_1x_2 + by_1y_2 \tag{3}$$

which the reader may check using high school algebra.

2.4. **Form with specified value.** The only other computation involving forms that we need is this: Given a form $F = (a, b, c)$ and a pair of relatively prime integers x, y at which F has the value $F(x, y) = r$, find a form (r, s, t) equivalent to F. The solution is simple and was known to Gauss. See page 49 of [3]. Use the Euclidean algorithm to find integers w and z with $xw - yz = 1$. Then

$$r = ax^2 + bxy + cy^2 \tag{4}$$

$$s = b(xw + zy) + 2(axz + cyw) \tag{5}$$

$$t = az^2 + bzw + cw^2 \tag{6}$$

works, as the reader may verify using high school algebra. The form (r, s, t) is equivalent to (a, b, c) because the transformation matrix

$$\begin{pmatrix} x & z \\ y & w \end{pmatrix}$$

from F to (r, s, t) has determinant $xw - yz = 1$, so it is in $\mathrm{SL}_2(\mathbb{Z})$.

2.5. **Ambiguous forms.** Both SQUFOF and SQUFOF2 work by finding an *ambiguous* form, a form (a, b, c) with $a \mid b$. Since $a \mid b^2 - 4ac = \Delta$, a must divide Δ. Conversely, if $a \mid \Delta$ and $a < \sqrt{\Delta}$, there is a reduced ambiguous form $(\pm a, b, c)$.

Ambiguous forms occur in reduced cycles at *symmetry points*, where $\rho((c, b, a)) = (a, b, c)$. Every symmetry point must have an ambiguous form by the definition of ρ, as $2b \equiv 0 \bmod 2a$, so $a \mid b$.

The class of an ambiguous form is also called ambiguous. The ambiguous classes are exactly the classes of order 2 in the class group by Buell [3] Corollary 4.9. Both SQUFOF and SQUFOF2 factor N by finding a square form (under composition) in the principal period of forms of discriminant $\Delta = N$ or $4N$. They take its square root, which is a form in an ambiguous class. Then they traverse this class to an ambiguous form, whose end coefficient is a factor of Δ and hopefully of N.

3. The infrastructure distance

We need the concept of infrastructure distance for the time complexity of SQUFOF2 (and also for that of the original SQUFOF).

3.1. **Fundamental discriminants.** Let N be an odd integer to be factored. Since squares are easy to detect and factor, we may assume that N is not a square. We also assume N is square free. If $N \equiv 1 \pmod 4$, let $\Delta = N$. If $N \equiv 3 \pmod 4$, let $\Delta = 4N > 0$. Then Δ is a *fundamental discriminant*, that is, $\Delta \equiv 1 \pmod 4$ or $\Delta \equiv 0 \pmod 4$ and $\Delta/4 \equiv 2$ or $3 \pmod 4$. In fact, both SQUFOF and SQUFOF2 appear to work fine even when N is not square free (but not a square) and $\Delta = 4N$ is not a fundamental discriminant (when $N \equiv 1 \pmod 4$), but we do not know how to analyze either algorithm without the assumption that Δ is a fundamental discriminant.

3.2. **Number fields and ideals.** Let $K = \mathbb{Q}(\sqrt{\Delta})$ be a number field. Let σ denote the nontrivial field automorphism of K: $\sigma(a + b\sqrt{\Delta}) = a - b\sqrt{\Delta}$. The norm of $\alpha \in K$ is $N(\alpha) = \alpha\sigma(\alpha) = a^2 - b^2\Delta$.

An *order* in K is a subring of the ring of algebraic integers in K containing 1 and with field of fractions K. In the following we focus on one order, namely, $A = \mathbb{Z}[(\Delta + \sqrt{\Delta})/2]$. The product $M_1 \cdot M_2$ of two subsets M_1, M_2 of K is the additive subgroup of K generated by the set of all products xy with $x \in M_1$, $y \in M_2$. An A-ideal is a subset M of K with $A \cdot M = M$. An invertible A-ideal is an A-ideal for which there exists M' with $M \cdot M' = A$. The inverse of M is $A \cdot M'$. The set $\mathcal{I}$ of all invertible A-ideals is a commutative group with respect to multiplication.

One can show (see [14]) that the invertible A-ideals are the subgroups of K of the form

$$M = \left(\mathbb{Z} + \left(\frac{b + \sqrt{\Delta}}{2a}\right)\mathbb{Z}\right) \cdot \alpha\,,$$

where $\alpha \in K^*$, a, $b \in \mathbb{Z}$ satisfy $c = (b^2 - \Delta)/(4a) \in \mathbb{Z}$, $\gcd(a, b, c) = 1$ and $N(\alpha)/a > 0$.

Formulas for multiplying invertible A-ideals are given on page 127 of [14]. They are the same ones used for composition of forms.

A *principal* A-ideal is an additive subgroup of K of the form $A\alpha$ with $\alpha \in K^*$ and $N(\alpha) > 0$. The principal A-ideals form a subgroup $\mathcal{P}$ of $\mathcal{I}$. The *class group* of A is the quotient $\mathcal{C} = \mathcal{I}/\mathcal{P}$. It is a finite group. Its order h is the *class number* of A. We use the same symbols $\mathcal{C}$, h because $\mathcal{C}$ is the same group as the class group of forms with discriminant Δ.

There is a correspondence between the binary quadratic forms of discriminant Δ and invertible A-ideals in $K = \mathbb{Q}(\sqrt{\Delta})$ defined by

$$(a, b, c) \leftrightarrow \left(\mathbb{Z} + \left(\frac{b + \sqrt{\Delta}}{2a}\right)\mathbb{Z}\right)\alpha,$$

where α is any element of K^*. If we write $\beta = ((b + \sqrt{\Delta})/(2a))\alpha$, then (a, b, c) corresponds to $\mathbb{Z}\alpha + \mathbb{Z}\beta$. Principal A-ideals correspond to forms equivalent to those in the principal cycle, that is, to F_0.

3.3. **Infrastructure distance defined.** Let η be the smallest element (a unit) of A for which $\eta > 1$ and $N(\eta) = 1$. The *regulator* of A is $R = \log \eta$.

Let Γ denote the subgroup

$$\Gamma = \left\{ \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} : m \in \mathbb{Z} \right\}$$

of $\mathrm{SL}_2(\mathbb{Z})$. Two forms (a_1, b_1, c_1) and (a_2, b_2, c_2) are in the same orbit under Γ if and only if $a_1 = a_2$ and $b_1 \equiv b_2 \pmod{2a_1}$. Let $\mathcal{F}$ denote the orbit space $\{$forms of discriminant $\Delta\}/\Gamma$.

Since Γ is a subgroup of $\mathrm{SL}_2(\mathbb{Z})$, there is a natural surjection from $\mathcal{F}$ onto the orbit space of equivalence classes of forms of discriminant Δ under the action of $\mathrm{SL}_2(\mathbb{Z})$, that is, onto the class group $\mathcal{C}$. Lenstra [14] defines a group structure on $\mathcal{F}$ that makes this map into a group homomorphism. Let $\mathcal{G}$ denote the kernel of this map.

Using the correspondence between invertible A-ideals and forms, Lenstra [14] defines a map d: $\mathcal{G} \to \mathbb{R}/R\mathbb{Z}$ with the property that if the form (a, b, c) corresponds to the principal ideal $A\gamma$, then

$$d((a,b,c)) = \frac{1}{2}\log\left|\frac{\sigma(\gamma)}{\gamma}\right| \mod R.$$

This map d is a small modification of the "distance" defined by Shanks [18]. The map d is a group homomorphism: $d(FG) = d(F) + d(G) \mod R$.

This distance, as a unary operator, is defined only on the principal cycle. To apply it within other cycles, Lenstra defines the infrastructure distance as the binary operator:

$$d(F,G) = d(GF^{-1})$$

Note that this is only defined for F, G in the same cycle, as that is precisely when GF^{-1} will be in the principal cycle.

3.4. **Formulas for infrastructure distance.** Lenstra [14] gives an explicit formula for the distance when reducing forms:

Theorem 1. *For an indefinite integral binary quadratic form $F(X,Y) = aX^2 + bXY + cY^2$ of discriminant Δ the infrastructure distance between F and $\rho(F)$ is given by*

$$d(F,\rho(F)) = \frac{1}{2}\log\left|\frac{b+\sqrt{\Delta}}{b-\sqrt{\Delta}}\right|.$$

For a proof, see section 11 of [14].

Lenstra [14] shows that the average value, taken over all forms in a cycle, of this distance is Lévy's constant

$$\ell = \pi^2/(12\log 2) \approx 1.19. \tag{7}$$

Thus, the distance d_n from the first form in a cycle to the n-th form is roughly proportional to n, with ℓ being the proportionality factor.

This explicit formula for the infrastructure distance leads to an interesting corollary, particularly in the context of the original SQUFOF:

Corollary 1. *For all reduced indefinite integral binary quadratic forms $F = (a',b',c')$, with principal form of the same discriminant $F_0 = (1,b,c)$, the distance $d(F,\rho(F)) \leq d(F_0,\rho(F_0))$, with equality if and only if $b' = b$.*

Proof. By the discriminant formula, $\Delta = b'^2 - 4a'c'$, and so $b' \equiv \Delta \mod 2$. The principal form is constructed as a reduced form with $a = 1$, and thus $|\sqrt{\Delta} - 2| < b < \sqrt{\Delta}$, and so b is uniquely determined. For any reduced F, b' has the same parity requirement and upper bound, and so we have $b' \leq b$, and $b' - \sqrt{\Delta} \leq b - \sqrt{\Delta} < 0$, giving

$$d(F,\rho(F)) = \frac{1}{2}\log\left|\frac{b'+\sqrt{\Delta}}{b'-\sqrt{\Delta}}\right| \leq \frac{1}{2}\log\left|\frac{b'+\sqrt{\Delta}}{b-\sqrt{\Delta}}\right| \leq \frac{1}{2}\log\left|\frac{b+\sqrt{\Delta}}{b-\sqrt{\Delta}}\right| = d(F_0,\rho(F_0)). \qquad \square$$

In the algorithm, we need the distance from the identity form F_0 to a square form F. For that, we find the distance of a matrix action:

Theorem 2. *For an indefinite integral binary quadratic form $F(X,Y) = aX^2 + bXY + cY^2$ of discriminant Δ and matrix*

$$S = \begin{pmatrix} x & z \\ y & w \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}),$$

the infrastructure distance between F and $S \cdot F$ is

$$d(F, S \cdot F) = \frac{1}{2} \log \left| \frac{2ax + y(b + \sqrt{\Delta})}{2ax + y(b - \sqrt{\Delta})} \right| \mod R,$$

where R is the regulator of $\mathbb{Q}[\sqrt{\Delta}]$.

Proof. Write $F'(X,Y) = S \cdot F = a'X^2 + b'XY + c'Y^2$. Write representatives for the invertible ideals corresponding to F, F':

$$M = \left(\mathbb{Z} + \frac{b + \sqrt{\Delta}}{2a} \mathbb{Z} \right) \qquad\qquad M' = \left(\mathbb{Z} + \frac{b' + \sqrt{\Delta}}{2a'} \mathbb{Z} \right)$$

We can use the substitution $F(xX + zY, yX + wY) = F'(X,Y)$ to verify that $\gamma = (2ax + y(b - \sqrt{\Delta}))/(2a') \in \mathbb{Q}[\sqrt{\Delta}]$ satisfies $M' = \gamma M$. Letting X and Y stand in for arbitrary integers:

$$\begin{aligned}
\gamma M &= \frac{\left(2ax + y(b - \sqrt{\Delta})\right)}{2a'} \cdot M \\
&= \frac{\left(2ax + y(b - \sqrt{\Delta})\right) \left((xX + zY) + \frac{b + \sqrt{\Delta}}{2a}(yX + wY) \right)}{2a'} \\
&= \frac{2 \underbrace{(ax^2 + bxy + cy^2)}_{=F(x,y)=a'} X + \left(\underbrace{2(axz + cwy) + b(wx + yz)}_{=b'} + \underbrace{(xw - yz)}_{=1} \sqrt{\Delta} \right) Y}{2a'} \\
&= \left(X + \frac{b' + \sqrt{\Delta}}{2a'} Y \right) = M'
\end{aligned}$$

This allows calculation of the infrastructure distance using the definition:

$$\begin{aligned}
d(F, S \cdot F) &= d(M, M') = d(\gamma M M^{-1}) = d(\gamma A) \\
&= \frac{1}{2} \log \left| \frac{\sigma(\gamma)}{\gamma} \right| = \frac{1}{2} \log \left| \frac{2ax + y(b + \sqrt{\Delta})}{2ax + y(b - \sqrt{\Delta})} \right|.
\end{aligned}$$

□

Corollary 2. *For an indefinite integral binary quadratic form $F(X,Y) = aX^2 + bXY + cY^2$ of discriminant Δ and primitive representation $F(x,y) = r$, the infrastructure distance between F and the form G with leading coefficient r, constructed as in Section 2.4, is*

$$d(F, G) = \frac{1}{2} \log \left| \frac{2ax + y(b + \sqrt{\Delta})}{2ax + y(b - \sqrt{\Delta})} \right| \mod R,$$

where R is the regulator of $\mathbb{Q}[\sqrt{\Delta}]$.

Proof. The matrix for this transformation is given as $\begin{pmatrix} x & z \\ y & w \end{pmatrix}$, for some w, z with $xw - yz = 1$, which exist as $F(x,y)$ is a primitive representation of v. The distance follows from Theorem 2. □

The form produced by the matrix action probably is not reduced, so we also need a bound on the distance to the nearby reduced form $\rho(G)$. A suitable bound is discussed in section 12 of [14], which states that the reduction of a form is one of the two forms closest in infrastructure distance above or below it with the same a sign, or the form with opposite a sign between them. This means the reduction adds at most two steps along the cycle over what would be expected from infrastructure distance before reduction.

During the operation of the original SQUFOF, we will also need the distance $d(\tau(F_n), F_0)$ for a form F_n on the principal cycle. We can construct this distance using a well known fact about τ.

Theorem 3. *For an indefinite integral binary quadratic form $F(X, Y) = aX^2 + bXY + cY^2$ of discriminant Δ, $\rho(\tau(F))$ is an inverse of F under composition in $\mathcal{F}$.*

Proof. τ is given by the matrix action by $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. ρ is an action by $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, followed by a matrix in Γ. The action by the product of these two matrices takes a form $aX^2 + bXY + cY^2$ to its inverse $aX^2 - bXY + cY^2$, and so $\rho(\tau(F))$ is equivalent to this inverse by an action of Γ, and thus equal in $\mathcal{F}$. □

Corollary 3. *With a principal indefinite integral binary quadratic form F_0 of discriminant Δ, and a reduced form $F_n = \rho^n(F_0)$ on the principal cycle,*

$$d(\tau(F_n), F_0) = d(F_0, F_{n+1}).$$

Proof. Note that by Theorem 1, we have $d(\tau(F_n), \rho(\tau(F_n))) = d(F_n, F_{n+1})$. By Theorem 3, we have $d(\rho(\tau(F_n)), F_0) = d(F_0, F_n)$. And so

$$\begin{aligned} d(\tau(F_n), F_0) &= d(\tau(F_n), \rho(\tau(F_n))) + d(\rho(\tau(F_n)), F_0) \\ &= d(F_n, F_{n+1}) + d(F_0, F_n) = d(F_0, F_{n+1}). \end{aligned}$$ □

This is not used for SQUFOF2, as SQUFOF2 constructs an inverse form that is not reduced, with square in the first coefficient, and as such directly uses the inverse operation $(a, b, c) \mapsto (a, -b, c)$ instead of τ.

4. Summary of the original SQUFOF

These preliminaries permit a brief description of SQUFOF. For more detail see [10].

To factor a composite nonsquare positive integer N, SQUFOF computes some forms in the principal cycle of forms with (fundamental) discriminant $\Delta = 4N$ (or N if $N \equiv 1 \pmod 4$). If $N \equiv 1 \pmod 4$, replace N with $2N$. Let $q = \lfloor\sqrt{N}\rfloor$ and $F_0 = (1, 2q, N - q^2)$. Compute $F_n = \rho^n(F_0)$ for $n = 2, 3, \ldots$ until you find a square form $F = F_n = (u, v, w^2)$ with $w > 0$. The index n will be even and $u < 0$. The inverse square root of F under composition is $G = F^{-1/2} = (-w, v, -uw)$. Now compute $G_m = \rho^m(G)$ for $m = 1, 2, 3, \ldots$ until you reach a *symmetry point*, that is, two consecutive forms G_m, G_{m+1} with the same middle coefficient f. Then either f (if f is odd) or $f/2$ (if f is even) has a good chance of being a proper factor of N. The infrastructure distance between G and G_{m+1} is exactly one-half that between F_0 and F_{n+1}, so that m is approximately $n/2$.

The symmetry point signals an *ambiguous* form $G_{m+1} = (g, f, e)$, described in Section 2.5, yielding a divisor g of $4N$. If $g \neq \pm1, \pm2$, then the algorithm yields a factor of N. The probability that SQUFOF succeeds this way depends on the number of distinct prime factors of N and is always at least 0.5 (heuristically).

As SQUFOF computes the F_n, it maintains a list of "bad" square forms which lead to a trivial factorization of N. These are the square forms whose inverse square roots lie in a cycle whose ambiguous form has $g = \pm 1$ or $g = \pm 2$. They are recognized in the sequence F_0, F_1, … by having a very small end coefficient. Use of this list improves the chance of success of SQUFOF to 1.0. See [10] for details.

SQUFOF has a time complexity of $O(N^{1/4})$ because that is the average distance between square forms on the principal period. It uses negligible memory.

5. Two examples of SQUFOF

Here is an example in which SQUFOF works well. Let

$$N_1 = 13290059, \quad q_0 = 2\left\lfloor \sqrt{N_1} \right\rfloor = 7290, \quad F_0 = (1, 7290, -4034).$$

Then $F = F_{51} = (-5107, 7256, 5^2)$ is the first square form. See Table 1.

Table 1. Principal period F_n for N_1

n		F_n		d_n
	a	b	c	
0	1	7290	-4034	0.000000
1	-4034	778	3257	4.743078
2	3257	5736	-1555	4.850191
3	-1555	6704	1321	5.912935
4	1321	6506	-2050	7.498563
49	-2327	5738	2174	51.098906
50	2174	2978	-5107	52.162371
51	-5107	7256	5^2	52.592824
52	5^2	7244	-6847	55.606202

Then $F^{-1/2} = (-5, 7256, 25535)$, $G_0 = (-5, 7286, 3722)$, and the symmetry point is $G_{23} = (571, 6238, -6238)$. Finally, $6238/2 = 3119$ divides N_1 and we have $N_1 = 3119 \cdot 4261$. Note that $m = 23 \approx \frac{1}{2}51$, $f_{23} = f_{24} = 6238$. The first square form, F_{51}, succeeded. See Table 2.

The second example shows how SQUFOF could fail without a list.

Let $N_2 = 42854447$. The square form is $F = F_{315} = (-6022, 10186, 53^2)$, as shown in Table 3.

So $F^{-1/2} = (-53, 10186, 319166)$, $G_0 = (-53, 13048, 5507)$, and the symmetry point is $G_{141} = (4331, 13092, -1)$. The factor of N_2 should be $13092/2 = 6546$, but this number does not divide N_2. See Table 4. Note that G_{136} through G_{141} are the forms F_5 through F_0 in Table 3 with the end coefficients reversed and their signs changed. The reason for failure is that the square root operation led into the period containing a form with end coefficient $g = -1$, the negative of the principal period. In this situation, SQUFOF would return to F_{315} and resume its search for a square form. SQUFOF with a list would notice that F was the square of a form earlier in the principal period and continue to the next square form.

Note that in both examples, the infrastructure distance d_n traversed in the second sequence is exactly half of that traversed in the first sequence, and that $d_n \approx \ell n$.

Table 2. Nonprincipal period G_n for N_1

n	G_n			d_n
	e	f	g	
$F^{-1/2}$	-5	7256	25535	
0	-5	7286	3722	0.000000
1	3722	158	-3569	3.978333
2	-3569	6980	311	4.000007
3	311	6704	-6605	5.912935
4	-6605	6506	410	7.498563
5	410	6614	-5741	8.931761
21	1130	5206	-5765	10.442852
22	-5765	6324	571	25.204238
23	571	6238	-6238	26.526552
24	-6238	6238	571	27.803101

Table 3. Principal period F_n for N_2

n	F_n			d_n
	a	b	c	
0	1	13092	-4331	0.000000
1	-4331	12894	298	5.293005
2	298	12734	-7771	7.729873
3	-7771	2808	5261	9.868265
4	5261	7714	-5318	10.086118
5	-5318	2922	7657	10.762535
313	-907	12088	6973	361.941673
314	6973	1858	-6022	363.552386
315	-6022	10186	53^2	363.695262
316	53^2	12286	-1822	364.735528

6. The new algorithm SQUFOF2

The expected value of the location n of the first square form in the principal period is $cN^{1/4}$ where the constant c is about $1.77/(2^k - 1)$ when N has $1+k$ (distinct) prime factors. (See [10].) This makes SQUFOF a $O(N^{1/4})$ time algorithm.

The new algorithm does not compute the forms in the principal cycle, so it cannot maintain a list of bad forms. Therefore, it sometimes fails. But, in contrast to SQUFOF, the new algorithm has many chances to succeed, so it does not matter that up to half of the chances fail. (The same statement is true of the Quadratic and Number Field Sieves.)

This work was inspired by the following quote from an unpublished manuscript [20] by Shanks found in his office after his death.

> Gauss [9] proved that *any* form F in the principal genus has a square root f such that $f \cdot f$ (under composition and reduction) $= F$. His constructive proof gives a remarkable algorithm for computing f.

Table 4. A (nonprincipal) period G_n for N_2

n	G_n			d_n
	e	f	g	
$F^{-1/2}$	-53	10186	319166	
0	-53	13048	5507	0.000000
1	5507	8880	-4121	3.186067
2	-4121	7504	6983	4.026201
3	6983	6462	-4642	4.678396
4	-4642	12106	1339	5.219150
136	-7657	2922	5318	169.581327
137	5318	7714	-5261	171.605229
138	-5261	2808	7771	172.281646
139	7771	12734	-298	172.499500
140	-298	12894	4331	174.637891
141	4331	13092	-1	177.074759
142	-1	13092	4331	182.367764

> The w^2 in (u, v, w^2) is the *value* of a form with $x = 0$ and $y = 1$. If, for any F_n, its value is a square for certain values of x and y, then one can easily construct an equivalent form (probably not reduced) that is a square form.
>
> This variation on SQUFOF was suggested by R. de Vogelaere when I first spoke on SQUFOF in [19]. He calls it the "fat" SQUFOF. One tries small pairs (x, y) in F_n to see if it has a small square value.

SQUFOF2 avoids the slow search for a square form by *constructing* one. It uses a *sieve* to factor some *values* of the first quadratic form F_0 in the principal period. (It could have used any form F_n in the principal cycle, but they all represent the same integers, and F_0 is more convenient because it is the identity for composition.) Integers with no prime factor other than those $< B$ are called "smooth" or "B-smooth." The sieve finds smooth values of $F_0(x, y)$ for (x, y) in a certain rectangle. Lenstra and Pomerance [15] devised a similar factoring algorithm using positive definite binary quadratic forms.

SQUFOF2 chooses a bound B for the primes to consider and a size S for the sieve region. The sieve begins by solving the congruence $r^2 \equiv N \bmod p$ for each prime $p < B$. For each $0 < y < S$ and for each $p < B$ the two solutions of $F_0(x, y) \equiv 0 \bmod p$ are computed using the quadratic formula. The sieve then divides each $F_0(x, y)$ with $-S < x < S$ by the primes that are known to divide it because they lie in two arithmetic progressions with common difference p. If the remaining cofactor is 1, then the number is smooth and the triple $(x, y, F_0(x, y))$ is saved.

Then the new algorithm uses linear algebra over $GF(2)$ to match the prime factors of a subset of the $F_0(x, y)$ and find a set whose product is a square (as in the quadratic sieve). See Example 8.7 in [22]. The linear algebra finds the left null space of a matrix (its cokernel) over $GF(2)$ with one row for each triple $(x, y, F_0(x, y))$ and one column for each prime $< B$. The i, j entry of the initial matrix is 1 if the j-th prime divides the i-th value of $F_0(x, y)$ to an odd power and 0 otherwise. The 1s in each vector in the left null space tell which values $F_0(x, y)$ to multiply to produce a square integer.

SQUFOF2 multiplies these values by composing the quadratic forms, using Formulas (2) and (3). When $\gcd(x, y) > 1$, the algorithm removes their common factor, which removes a square from the product. (Thus the square value of the composition of all the forms might be a proper divisor of the square value constructed by the linear algebra.)

It uses Formulas (4), (5), (6) to convert this form and its square value into another form which is a square form having the same square value as its end coefficient.

Then SQUFOF2 computes the inverse square root of the square form as in the regular SQUFOF. Proposition 3.1 of [10] says that if $\gcd(v, w) = 1$ and $F = (u, v, w^2)$ is a square form on the principal cycle, then $(-w, v, -uw)$ is a square root of F. The proof using the formulas in Section 2.1.3 of [10] shows that this is true so long as F is equivalent to a form in the principal cycle, whether F is reduced or not. It is clear from the quote above that Shanks (and Gauss) knew this.

See Wagstaff [22] Chapters 8 and 6 for an introduction to the Quadratic Sieve and basic SQUFOF. See Gower and Wagstaff [10] for an analysis of basic SQUFOF. See Crandall and Pomerance [7] Section 6.1 or Pomerance [16] for more about the Quadratic Sieve, including a proof of its time complexity. The proof of the time complexity of SQUFOF2 will closely resemble Pomerance's argument in [16]. We give the proof in Section 8 below.

7. Two examples of SQUFOF2

Here we trace the entire SQUFOF2 algorithm for factoring $N_3 = 13847$. Part of the principal period for N_3 is given in Table 5. The forms all have discriminant $\Delta = 4N_3 = 55388$.

Table 5. Principal period F_n for $N_3 = 13847$

n	F_n			d_n
	a	b	c	
0	1	234	−158	0.000000
1	−158	82	77	2.926897
2	77	226	−14	3.290544
3	−14	222	109	5.240115
4	109	214	−22	7.007206
27	−46	182	121	27.043476
43	1	234	−158	45.133347

The regular SQUFOF algorithm with a list would compute the principal period for 27 steps and encounter the square form $(-46, 182, 11^2)$, which leads to a nonprincipal period where it finds a proper factor of N_3 after 10 more steps. On its way through the principal period it places on its list two small values c from forms (a, b, c) that prevent it from failing when it encounters two other square forms before the one in step 27, as these lead to trivial factors.

SQUFOF2 sieves the first form F_0 in the principal period seeking smooth numbers. The prime factors of the values of this form at (x, y) with $\gcd(x, y) = 1$ (that is, the values primitively represented by F_0) are restricted to 2 and those p for which the Legendre symbol $(N_3/p) = 1$. The set of these primes that are $< B$ is called the "factor base." Since F_0 is indefinite ($\Delta > 0$), the number -1 is included as a "prime" in the factor base. In this tiny example we use $B = 75$ and the factor base

$$\{-1, 2, 7, 11, 17, 23, 37, 43, 59, 71, 73\}$$

consisting of -1, 2 and the first 9 primes p with $(N_3/p) = +1$.

We sieve the first form over the range $-20 < x < 20$ and $0 < y < 20$, saving only smooth values with $\gcd(x, y) = 1$. We omit $y < 0$ since $F_0(-x, -y) = F_0(x, y)$. We find 57 values that factor completely using the factor base, including these eight:

$$\begin{aligned}
F_0(2,3) &= -14 = -2 \cdot 7 \\
F_0(-1,1) &= -391 = -17 \cdot 23 \\
F_0(7,10) &= 629 = 17 \cdot 37 \\
F_0(-4,1) &= -1078 = -2 \cdot 7^2 \cdot 11 \\
F_0(8,5) &= 5474 = 2 \cdot 7 \cdot 17 \cdot 23 \\
F_0(14,3) &= 8602 = 2 \cdot 11 \cdot 17 \cdot 23 \\
F_0(-19,2) &= -9163 = -7^2 \cdot 11 \cdot 17 \\
F_0(1,12) &= -19943 = -7^2 \cdot 11 \cdot 37
\end{aligned}$$

Linear algebra over $GF(2)$ constructs subsets of the form values whose product is square. It finds that

$$F_0(8,5)F_0(2,3)F_0(-1,1) = 5474 \cdot (-14) \cdot (-391) = 2^2 7^2 17^2 23^2$$

is a square.

Formulas (2) and (3) tell us that $F_0(8,5)F_0(2,3) = F_0(2386, 3544) = -76636$. Since $\gcd(2386, 3544) = 2$, we cancel the common factor 2 and find that $F_0(1193, 1772) = -19159$.

Formulas (2) and (3) tell us that $F_0(1193, 1772)F_0(-1, 1) = F_0(238783, 414069)$. Since $\gcd(238783, 414069) = 391$, we cancel the common factor 391 and find that $F_0(713, 1059) = 49 = 7^2$. Formulas (4), (5), (6) convert this to the square form $(7^2, -226, -22)$. The inverse square root of $(-22, 226, 7^2)$ is $(7, 226, -22 \cdot 7) = (7, 226, -154)$. This leads in 2 steps (applications of ρ) to $(79, 234, -2)$ and failure.

We try another dependency. Linear algebra finds that

$$F_0(14,3)F_0(-1,1)F_0(-4,1) = 8602 \cdot (-391) \cdot (-1078) = 2^2 7^2 11^2 17^2 23^2,$$

another square.

Formulas (2) and (3) tell us that $F_0(14,3)F_0(-1,1) = F_0(460, 713)$. Removing the common factor 23 gives us $F_0(20, 31) = -6358$. Formulas (2) and (3) give $F_0(20,31)F_0(-4,1) = F_0(4818, 7150)$. We cancel the common factor 22 and find that $F_0(219, 325) = 119^2$.

Formulas (4), (5), (6) convert this to the square form $(119^2, -4244, 317)$. The inverse square root of $(317, 4244, 119^2)$ is $(119, 4244, 317 \cdot 119) = (119, 4244, 37723)$. This reduces to $(-113, 40, 119)$ and leads in 3 steps to $(83, 122, -122)$ and the factorization $13847 = 61 \cdot 227$.

Now we factor $N_1 = 13290059$ with less detail. The factorization of this N_1 with regular SQUFOF was shown above. The first form is $F_0 = (1, 7290, -4034)$. After some experimentation, we choose a factor base consisting of all primes < 115 and sieve region $-226 < x < 226$ and $0 < y < 226$. Thus the factor base is $\{-1, 2, 5, 13, \ldots, 113\}$ of size 15. After sieving only the first 5 rows, that is, $1 \le y \le 5$, we have found 13 smooth values. We decide to perform the linear algebra and discover two solutions.

The first solution tells us that the product of the four values

$$F_0(-22,1), F_0(94,3), F_0(-69,4), F_0(55,4)$$

is a square. As we compose these forms using Formulas (2) and (3) we remove common factors of 2, 1261 and 3827 and arrive at $F_0(1067091, 1928527) = 11405172025 = 106795^2$.

Formulas (4), (5), (6) produce the square form

$$(11405172025, 10816923944, 2564754029) = (106795^2, 10816923944, 2564754029)$$

with inverse square root $(-273902906527055, 10816923944, -106795)$. This form reduces to $(3469, 2876, -3235)$, which leads in four steps to the symmetry point $(-2, 7290, 2017)$ and failure.

The second solution tells us that the product of the ten values

$$\begin{gathered} F_0(-53,1), F_0(-22,1), F_0(-1,1), F_0(-61,2), F_0(21,2), \\ F_0(-157,3), F_0(49,3), F_0(-69,4), F_0(-69,4), F_0(216,5) \end{gathered}$$

is a square. We remove many common factors as we compose these forms and find

$$\begin{aligned} F_0(1474289783211707013, 2664449249336597236) &= 2416556752202799184225 \\ &= 49158486065^2. \end{aligned}$$

Formulas (4), (5), (6) produce the square form

$$(49158486065^2, 1610582552405188463444, 2683545664453655576761)$$

with inverse square root

$$(-1319190421508362028903630633546 5, 1610582552405188463444, -49158486065).$$

This form reduces to $(4315, 2634, -2678)$, which leads in eleven steps to the symmetry point $(3119, 6238, -1142)$ and the factor $6238/2 = 3119$ of N_1.

8. Time and space complexity of SQUFOF2

8.1. **Parameters of SQUFOF2.** Following Pomerance [16], let $L = L(N)^{1+o(1)}$. Hiding the $o(1)$ this way allows us to absorb constants and powers of $\log N$ and $\log\log N$ into L and greatly simplify the presentation. We may write, for example, seemingly incorrect equations like $L \log N = L \log\log N = 2L = \pi(L) = L$, where $\pi(B) \approx B/\log B$ is the number of primes $\leq B$.

Let $M(k)$ denote the time needed to multiply (or divide or remainder) two integers of length k digits. The schoolboy methods show $M(k) = O(k^2)$. Using Schönhage-Strassen (See Section 4.3.3 of Knuth [12]), one can improve this to $M(k) = O(k \log k \log\log k)$. Of course, it is well known that one can add and subtract two k-digit integers in $O(k)$ steps. As the example of SQUFOF2 factoring $N = 13290059$ shows, the intermediate numbers may be much larger than N. We will show that the entire arithmetic with all of them is not slower than the sieve or linear algebra steps.

The algorithm has two parameters: the size of the factor base and the area of the sieve region. We specify these using two constants α, β in the interval $(0.1, 1)$ to be determined later. The factor base consists of -1, 2, and all primes $p < L^\alpha$ with $(N/p) = +1$. We will sieve the values of $F_0(x, y)$ with $\gcd(x, y) = 1$ and $-L^\beta < x < L^\beta$, $0 < y < L^\beta$.

8.2. **Heuristic assumptions.** We now examine the individual steps of SQUFOF2 and estimate the complexity of each in terms of α and β.

The initialization of $F_0 = (1, b, c)$ consists of computing $b = 2\lfloor\sqrt{N}\rfloor$ and $c = (b^2 - \Delta)/4$. The square root may be found in $O(\log^2 N)$ steps by an integer variation of Newton's method as in Algorithm 1.7.1 of Cohen [5]. The rest of the arithmetic may be done in $O(\log^2 N)$ steps.

The proof of the time complexity is heuristic. It requires several plausible hypotheses. First we assume there are enough primes in the factor base.

Hypothesis 1. *There is a constant n_1 such that if $N > n_1$, then for any $\alpha \in (0.1, 1)$ the number of primes $p < L^\alpha$ for which $p \nmid N$ and $(N/p) = +1$ is at least $\pi(L^\alpha)/3$.*

Hypothesis 1 is plausible because the expected number of such primes is $\pi(L^\alpha)/2$. Next, we assume the sieve will produce enough smooth numbers.

Hypothesis 2. *There is a constant n_2 such that if $N > n_2$, then for any $\alpha \in (0.1, 1)$ and any $\beta \in (0.1, 1)$ the values $|F_0(x, y)|$ with $-L^\beta < x < L^\beta$, $0 < y < L^\beta$ and $\gcd(x, y) = 1$ have the same probability of being L^α-smooth as all integers in $(1, \max |F_0(x, y)|)$.*

We need $\gcd(x, y) = 1$ to construct a form (r, s, t) with $r = |F_0(x, y)|$ as explained at the end of Section 2.4. Hypothesis 2 is plausible because when $\gcd(x, y) = 1$ there is no reason to expect $|F_0(x, y)|$ to have larger or smaller prime factors than other integers of the same size. If $\gcd(x, y) > 1$, then $\gcd(x, y)^2$ would divide $|F_0(x, y)|$, so it might have more prime factors than usual. Another concern might be that only half of the primes p can divide $|F_0(x, y)|$, namely those with $(N/p) = +1$. But each of these primes has twice the chance of dividing $|F_0(x, y)|$ because for each y the quadratic congruence $F_0(x, y) \equiv 0 \bmod p$ has two solutions when $(N/p) = +1$. These two effects exactly cancel and leave the probability of being smooth unchanged. See page 118 of [16] for details of this probability calculation.

The sieve begins by solving the congruence $r^2 \equiv N \bmod p$ for each prime in the factor base. The naive algorithm of testing each r in $1 < r < p/2$ takes $O(L^{2\alpha})$ steps and is good enough for us. (The modular square root may be done quickly by Algorithm 2.3.9 of [7].) For each $0 < y < L^\beta$ and for each p the solutions of $F_0(x, y) \equiv 0 \bmod p$ are computed using the quadratic formula. This takes $O(L^\beta L^\alpha) = O(L^{\alpha+\beta}) = O(L^{2\max(\alpha,\beta)})$ steps.

The sieve then divides each $F_0(x, y)$ by the primes that are known to divide it. If the remaining cofactor is 1, then the number is smooth and the triple $(x, y, F_0(x, y))$ is saved. It is known [17] or [2] that this can be done for k values in $O(k \log k \log\log k)$ steps. SQUFOF2 has L^β sieves, one for each y, of length $2L^\beta$, so the total number of steps for all sieving is $O(L^{2\beta})$.

The linear algebra finds the left null space of a matrix over $GF(2)$ with one row for each triple $(x, y, F_0(x, y))$ and one column for each prime in the factor base. The sieve finishes when there are a few more triples than primes, so the matrix is nearly square with order $\pi(L^\alpha)$, that is, $O(L^\alpha)$. Gaussian elimination finds the left null space of a matrix of order k in $O(k^3)$ steps. Other methods reduce this complexity to $O(k^r)$ steps for some $r \in (2, 3]$. For example, Coppersmith and Winograd [6] give a method with $r \approx 2.49$. We shall assume that SQUFOF2 uses a method with complexity $O(L^{r\alpha})$ steps.

For each basis vector of the left null space we must compose the forms whose rows appear in the linear dependency. There are no more than L^α of them. As we iterate Formulas (2) and (3) let the (x, y) for the i-th form be (x_1, y_1), (x_2, y_2), etc. The i-th iteration replaces (x, y) by $(xx_i - cyy_i, xy_i + yx_i + byy_i)$. Since b, $|c| < 2\sqrt{N}$ and each $|x_i|$, $y_i < L^\beta$, the final (x, y) of the composition of all the forms in one dependency has $|x|$, $|y|$ bounded by $(2\sqrt{N}L^\beta)^{L^\alpha}$. The size of this number, its logarithm, is $O(L^\alpha \log(2\sqrt{N}L^\beta)) = O(L^\alpha)$. The complexity of arithmetic with numbers of that size is $O(L^{2\alpha})$ and there are no more than $O(L^\alpha)$ such arithmetic operations, for a total complexity of $O(L^{3\alpha})$ steps for the composition process. Using fast multiplication techniques, such as the Schönhage-Strassen method mentioned above, we can reduce this to $O(L^{2\alpha})$ steps.

The next step is to construct a square form from the form with a square value using Formulas (4), (5), (6). This arithmetic takes $O(L^{2\alpha})$ steps or $O(L^{\alpha})$ steps using fast multiplication. Finding the inverse square root of the square form has the same complexity.

According to Formula (1), the number of steps in the reduction of the inverse square root is proportional to the size of its third coefficient, which is $O(L^{\alpha})$ steps. Each step involves arithmetic with numbers of this size or smaller, so the total complexity of the reduction is $O(L^{2\alpha})$ steps.

The final step of SQUFOF2 is the search for the symmetry point. The infrastructure distance to it from the (reduced) inverse square root is exactly half that from F_0 to the square form. The infrastructure distance from F_0 to the square form is given by Corollary 2. To estimate this distance, we assume the denominator is at least 1 for a positive proportion of the values.

Hypothesis 3. *For pairs (x, y) with square value $F_0(x, y)$ computed as in SQUFOF2, at least $\frac{1}{2}$ of the pairs satisfy $|2ax + y(b - \sqrt{\Delta})| > 1$.*

This hypothesis is reasonable, as the relative size of x and y vary considerably through the possible solutions. By Hypothesis 3 and Corollary 2, we can find solutions with infrastructure distance bounded by $\frac{1}{2}\log\left|2x + y(b + \sqrt{\Delta})\right| \leq L^{\alpha}$. In sum, the total number of forms traversed seeking the symmetry point is $O(L^{\alpha})$, the number of steps in the reduction and the return.

At the symmetry point the only remaining operation is a gcd of N and the middle coefficient of size $O(\sqrt{N})$, which may be done in $O(\log^2 N)$ steps. We need to assume that the symmetry point gives a proper factor of N with probability $\geq 1/2$. Hypothesis 4 is the same as Assumption 4.19 in [10].

Hypothesis 4. *Each of the reduced ambiguous forms of the fundamental discriminant Δ has an equal chance of being the one at the symmetry point.*

For real numbers $0 < B \leq A$, let $\psi(A, B)$ denote the number of integers $\leq A$ all of whose prime factors are $\leq B$. Dickman [8] was the first to notice that one should use a log scale to estimate $\psi(A, B)$. He sketched a proof that for large A one has $\psi(A, B) \approx Au^{-u}$, where $u = (\log A)/\log B$. In other words, the probability that a positive integer $\leq A$ is B-smooth is approximately u^{-u}. See Knuth and Trabb-Pardo [13] and Canfield, Erdős, Pomerance [4] for proofs of precise versions of Dickman's theorem. Hypothesis 2 assumes that the probability that values $F_0(x, y)$ in the sieve rectangle $-L^{\beta} < x < L^{\beta}$, $0 < y < L^{\beta}$ with $\gcd(x, y) = 1$ are L^{α}-smooth is the same as for all integers in $(1, \max|F_0(x, y)|)$.

The cover design of [1] illustrates the following lemma, which is well known.

Lemma 1. *Let m be a large integer. Let $G(m)$ be the number of pairs (x, y) of integers with $1 \leq x \leq m$, $1 \leq y \leq m$ and $\gcd(x, y) = 1$. Then $G(m) = (6/\pi^2)m^2 + O(m \log m)$ as $m \to \infty$.*

Proof. The set counted by $G(m)$ is the union of the two sets $\{(x, y) : 1 \leq x \leq y \leq m;\ \gcd(x, y) = 1\}$, $\{(x, y) : 1 \leq y \leq x \leq m;\ \gcd(x, y) = 1\}$ whose intersection is the singleton $\{(1, 1)\}$. Each of these two sets has size $\sum_{x=1}^{m} \phi(x)$, where ϕ is Euler's function. But $\sum_{x=1}^{m} \phi(x) = (3/\pi^2)m^2 + O(m \log m)$ by Theorem 330 of [11] or Theorem 3.7 of [1]. □

8.3. Time and space complexity of SQUFOF2.

Theorem 4. *Assuming Hypotheses 1, 2, 3 and 4, the expected time complexity of SQUFOF2 to factor a large square free integer N using an elimination method with an exponent r is $L(N)^{r/\sqrt{4r-4}+o(1)}$. The space complexity is $L(N)^{1/\sqrt{r-1}+o(1)}$.*

Proof. The assumption that N is square free implies that Δ is a fundamental discriminant. This property is needed for Hypothesis 4 to make sense as in [10]. Hypothesis 1 is needed to construct the factor base for the sieve. Hypothesis 2 is used to ensure that enough smooth values are found so that the linear algebra will produce linear dependencies. Hypothesis 4 guarantees that each square form has probability at least 1/2 of leading to a proper factor of N. Hypothesis 3 is used to bound the distance travelled in the return step.

This proof is for the case $N \equiv 3 \bmod 4$. The proof for $N \equiv 1 \bmod 4$ is similar.

The discussion above shows that the most time-consuming steps of SQUFOF2 are the sieve ($L^{2\beta}$) and linear algebra ($L^{r\alpha}$). The sieve initialization, composition and reduction each take $L^{2\alpha}$ steps, but $2 < r \le 3$. The other steps are even faster.

By Hypothesis 2, the probability that $F_0(x,y)$ is smooth is the same as the probability that a random integer in the interval $[1,m]$ is smooth, where m is the maximum value of $|F_0(x,y)|$ in the sieve rectangle. Since b and $|c| < 2\sqrt{N}$, $-L^\beta < x < L^\beta$ and $0 < y < L^\beta$, we see that $m < 4\sqrt{N}L^{2\beta}$. We will choose β so that we must sieve at least one-tenth of the sieve rectangle to get enough smooth values, so $\log m \approx \log(4\sqrt{N}L^{2\beta})$. There are $L^{2\beta}$ pairs (x,y) in the rectangle and, by Lemma 1, $(6/\pi^2)L^{2\beta}$ of them have $\gcd(x,y)=1$. The constant $6/\pi^2$ and the $O(L^\beta \log L^\beta)$ from Lemma 1 are absorbed by our convention on L. The number of smooth relations is $L^{2\beta}$ times the probability that $|F_0(x,y)|$ is L^α-smooth, which by Hypothesis 2 is $L^{2\beta}\psi(m,L^\alpha)/m$. We will have enough smooth relations, that is, L^α of them, when $L^{2\beta}\psi(m,L^\alpha)/m = L^\alpha$. By Dickman's theorem (or Theorem 2.1 of [16]), $\psi(A,B) \approx Au^{-u}$, where $u = (\log A)/(\log B)$. Then, ignoring the 4,

$$u = \frac{\log m}{\log L^\alpha} = \frac{\log(\sqrt{N}L^{2\beta})}{\log L^\alpha} = \frac{\log N}{2\alpha \log L} + \frac{2\beta}{\alpha}.$$

We may ignore the constant $\frac{2\beta}{\alpha}$. A short calculation shows that $\log u = (\log\log N)/2$, $\log u^{-u} = \frac{-1}{4\alpha}\sqrt{\log N \log\log N}$ and $u^{-u} = L^{-1/(4\alpha)}$. We will have enough smooth relations if $L^{2\beta}L^{-1/(4\alpha)} = L^\alpha$, or $2\beta - 1/(4\alpha) = \alpha$, or $\beta = \frac{\alpha}{2} + \frac{1}{8\alpha}$. Choose this value for β. Then the time complexity will be

$$L^{\max(2\alpha,2\beta,r\alpha)} = L^{\max(2\alpha,\alpha+1/(4\alpha),r\alpha)} = L^{r\alpha}.$$

The space requirement is $L^{2\alpha}$ for the matrix and $L^\beta = L^{\alpha/2+1/(8\alpha)} < L^{2\alpha}$ for the sieve. Choose

$$\alpha = \frac{1}{2\sqrt{r-1}}$$

to obtain the theorem statement. □

We have shown that the time and space complexities of SQUFOF2 are the same as for the Quadratic Sieve. Compare Theorem 7.1 of [16].

Note that the exponent for the time complexity is

$$r\alpha = \frac{r}{2\sqrt{r-1}} = 1 + O((r-2)^2)$$

as $r \to 2$, so is not sensitive to small changes in r between 2 and 3.

With $r = 3$ the time and space complexity exponents are $3/(2\sqrt{2}) \approx 1.06$ and $1/\sqrt{2} \approx 0.71$. With $r = 2.49$ they are 1.02 and 0.82. With $r = 2$ (slightly better than the best conceivable elimination method) they would be 1.00 and 1.00.

With $r = 3$ the exponents on L for the size of the factor base and the length of the sieve interval are $\alpha = 1/(2\sqrt{2}) \approx 0.35$ and $\beta = 3/(4\sqrt{2}) \approx 0.53$. With $r = 2.49$ they are 0.41 and 0.51. With $r = 2$ they would be 0.50 and 0.50.

These values are for factoring large N in theory. For a practical program one should experiment with values of α and β to determine which are best. For factoring N between 10 and 30 decimal digits we found that $\alpha \approx 0.7$ and $\beta \approx 0.8$ are about right.

9. Conclusion

The new integer factoring algorithm SQUFOF2 presented here is interesting, but it is no faster than the Quadratic Sieve, and that method is slower than the Number Field Sieve for large integers. That is why we have not tried SQUFOF2 on very large integers.

The Quadratic Sieve has several variations that accelerate it but do not change its theoretical time complexity. These include using multipliers, using large primes, adding approximate logarithms in the sieve rather than dividing, fast linear algebra, and using multiple polynomials with self initialization. See [16] and Section 6.1 of [7]. All of these variations work well in SQUFOF2. Self initialization of multiple polynomials works especially well in SQUFOF2 because $F_0 = (a, b, c)$ with $a = 1$, so that the calculations in Formula (6.3) on page 239 of [7], which are the same for SQUFOF2 as for the Quadratic Sieve, become trivial addition and subtraction.

SQUFOF2 factors 30-digit integers in about half a minute on a PC, while SQUFOF would take about a minute for numbers of that size. A basic Quadratic Sieve would also take about half a minute on a PC.

References

[1] T. M. Apostol. *Introduction to Analytic Number Theory.* Springer-Verlag, New York, 1976.

[2] E. Bach and J. Shallit. *Algorithmic Number Theory, Volume I: Efficient Algorithms.* The MIT Press, Cambridge, Massachusetts, 1996.

[3] D. A. Buell. *Binary Quadratic Forms, Classical Theory and Modern Computations.* Springer-Verlag, Berlin, New York, 1989.

[4] E. Canfield, P. Erdős, and C. Pomerance. On a problem of Oppenheim concerning "factorisatio numerorum". *J. Number Theory*, 17:1–28, 1983.

[5] H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer-Verlag, New York, 1996.

[6] D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplication. *SIAM J. Comput.*, 11:472–492, 1982.

[7] R. Crandall and C. Pomerance. *Prime Numbers: A Computational Perspective.* Springer-Verlag, New York, 2001.

[8] K. Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Ark. Mat., Astronomi och Fysik*, 22A, 10:1–14, 1930.

[9] C. F. Gauss. *Disquisitiones Arithmeticae.* Yale University Press, New Haven, English edition, 1966.

[10] J. Gower and S. S. Wagstaff, Jr. Square form factorization. *Math. Comp.*, 77:551–588, 2008.

[11] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers.* Clarendon Press, Oxford, England, Fifth edition, 1979.

[12] D. E. Knuth. *The Art of Computer Programming, Volume 2, Seminumerical Algorithms.* Addison-Wesley, Reading, Massachusetts, Second edition, 1981.

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

[14] H. W. Lenstra, Jr. On the calculation of regulators and class numbers of quadratic fields. In J. V. Armitage, editor, *Journées Arithmétiques, 1980*, volume 56 of *Lecture Notes Series*, pages 123–150. London Math. Soc., 1982.

[15] H. W. Lenstra, Jr. and C. Pomerance. A rigorous time bound for factoring integers. *Jour. Amer. Math. Soc.*, 5(3):483–516, 1992.

[16] 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 1*, volume 154 of *Math. Centrum Tract*, pages 89–139, CWI, Amsterdam, 1982.

[17] P. A. Pritchard. A sublinear additive sieve for finding primes. *Communications of the ACM*, 24:18–23, 1981.

[18] D. Shanks. The infrastructure of a real quadratic field and its applications. In *Proceedings of the 1972 Number Theory Conference, Boulder*, pages 217–224, 1972.

[19] D. Shanks. Square forms factorization. Lecture, before 1975.

[20] Daniel Shanks. SQUFOF Notes. Manuscript, 30 pages, available at `http://homes.cerias.purdue.edu/∼ssw/shanks.pdf`.

[21] Alfred J. van der Poorten. A note on NUCOMP. *Math. Comp.*, 72:1935–1946, 2003.

[22] S. S. Wagstaff, Jr. *The Joy of Factoring*, volume 68 of *Student Mathematical Library.* Amer. Math. Soc., Providence, Rhode Island, 2013.

Email address: `clintonbradford@gmail.com`

Department of Mathematics, Purdue University, West Lafayette, IN 47907-2067, USA

Email address: `ssw@cerias.purdue.edu`

Center for Education and Research in Information Assurance and Security and Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398, USA