Doubly focused enumeration of locally square polynomial values

Daniel J. Bernstein
Department of Mathematics, Statistics, and Computer Science (M/C 249)
The University of Illinois at Chicago
Chicago, IL 60607–7045
djb@cr.yp.to

Abstract. Let $f$ be a nonconstant squarefree polynomial. Which of the values $f(c + 1), f(c + 2), \ldots, f(c + H)$ are locally square at all small primes? This paper presents an algorithm that answers this question in time $H/M^{2+o(1)}$ for an average small $c$ as $H \to \infty$, where $M = H^{1/2 + o(1)}$. In contrast, the usual method takes time $H/M^{1+o(1)}$. This paper also presents the results of two record-setting computations: an enumeration of locally square integers up to $24 \cdot 2^{54}$, and an enumeration of locally square values of $x^3 + y^7$ for small $x$ and $y$.

1 Introduction

A rational number is locally square at a prime $p$ if it is a square in the $p$-adic field $\bar{\mathbb{Q}}_p$. In particular:

- A nonzero integer is locally square at 2 if and only if it is a square modulo 2, 4, 8, 16, \ldots: equivalently, it has the form $2^{2e} s$ where $s$ is an odd square modulo 8, i.e., $s \mod 8 = 1$.
- A nonzero integer is locally square at an odd prime $p$ if and only if it is a square modulo $p, p^2, p^3, p^4, \ldots$: equivalently, it has the form $p^{2e} s$ where $s$ is a nonzero square modulo $p$, i.e., $s^{(p-1)/2} \mod p = 1$.

A nonnegative rational number is a square if and only if it is locally square at every prime $p$.

Consider the problem of enumerating all integers $x$ in a specified interval such that a specified polynomial in $x$ is locally square at all primes below a specified bound. Consider, for example, the problem of enumerating all integers $x$ between 0 and $10^{17} - 1$ such that $x^3 + 1$ is locally square at all primes in $\{2, 3, 11, 23\}$.

One can simply check, for each of the $10^{17}$ values of $x$, the local squareness of $x^3 + 1$. This is an example of what I call “unfocused enumeration.”

2000 Mathematics Subject Classification. Primary 11Y16.
A standard improvement is to first compute the possibilities for \( x \) modulo some small integer \( m \), then focus on the corresponding arithmetic progressions. For example, \( x \mod 5 \) must be in \( \{0, 2, 4\} \), so one can focus attention on the \( 10^{17}/5 \) values of \( x \) with \( x \mod 5 = 0 \), then the \( 10^{17}/5 \) values of \( x \) with \( x \mod 5 = 2 \), and finally the \( 10^{17}/5 \) values of \( x \) with \( x \mod 5 = 4 \), avoiding 40% of the work. Even better, there are only 18 possibilities for \( x \mod 55 \), so one can focus attention on 18 arithmetic progressions modulo 55, avoiding more than 67% of the work. These are examples of what I call “focused enumeration.”

The number of \( x \)'s drops exponentially with the number of prime divisors of \( m \): each additional prime reduces the number of \( x \)'s by a factor of approximately 2. But the number of arithmetic progressions, and thus the overhead of considering each arithmetic progression, grows more than exponentially with the number of prime divisors. The minimum computation time is achieved for \( m \) somewhere around \( 10^{14} \); the best choice of \( m \) depends on the relative costs of considering an arithmetic progression and considering an \( x \).

The point of this paper is that another technique, which I call “doubly focused enumeration,” makes the overhead much smaller, allowing \( m \) to be chosen much larger. For example, one can reasonably take \( m \approx 3.1 \cdot 10^{20} \) as 8 times the product of all primes between 3 and 67, reducing the number of \( x \)'s by a factor of about 66123.

Section 2 of this paper explains doubly focused enumeration in a more general setting. Section 3 returns to locally square polynomial values:

- It presents an algorithm that uses doubly focused enumeration to figure out which of \( f(c+1), f(c+2), \ldots, f(c+H) \) are locally square at all primes \( p \leq h \), given a nonconstant squarefree polynomial \( f \), an integer \( c \), a positive integer \( H \), and an integer \( h \geq 2 \log H \).
- It proves that this algorithm typically takes time \( H/M^{2+o(1)} \) where \( M = H^{1/\log \log H} \); here \( \log \) is \( \log_2 \) as usual. More precisely: The average time for the algorithm, when \( c \) is a uniform random element of an interval of length \( H^2 \), is \( H/M^{2+o(1)} \) for fixed \( f \) as \( H \to \infty \), provided that \( c \) has at most \( M^{o(1)} \) digits and \( h \) is in \( M^{o(1)} \). I do not know how to prove the same bound for a fixed \( c \).

For comparison: Focused enumeration takes time \( H/M^{1+o(1)} \).

Section 4 and Section 5 report, as examples of doubly focused enumeration, two record-setting computations. The first computation showed that every non-square positive integer below \( 24 \cdot 2^{64} \) is locally non-square at some prime in \( \{2, 3, \ldots, 283\} \); this is numerical evidence for a fast deterministic primality proving. The second computation, which is in progress right now, is scanning \( 10^{20} \) small pairs \((x, y)\) to find locally square values of \( x^3 + y^7 \) with \( x, y \) coprime. I should have the second computation done by mid-May; perhaps I’ll also extend the first computation.

## 2 Doubly focused enumeration

Consider the general problem of finding all integers \( x \in [1, H] \) such that \( x \mod m_1 \in S_1 \) and \( x \mod m_2 \in S_2 \). Here \( H \) is a positive integer; \( m_1 \) and \( m_2 \) are coprime positive integers; \( S_1 \) is a subset of \( \mathbb{Z}/m_1 \); and \( S_2 \) is a subset of \( \mathbb{Z}/m_2 \).
This section presents three solutions to this problem. The solutions do not have standard names; I call them “unfocused enumeration,” “focused enumeration,” and “doubly focused enumeration,” as in Section 1.

In common applications, focused enumeration is asymptotically faster than unfocused enumeration, and doubly focused enumeration is asymptotically faster than focused enumeration.

The following sizes are typical for applications: $H \approx 10^{30}$; $m_1 \approx m_2 \approx 10^{14}$; and $\#S_1 \approx \#S_2 \approx 10^{11}$, so that $H(\#S_1/m_1)(\#S_2/m_2) \approx 10^{14}$. In many situations, one can prove that the number of outputs is approximately $H(\#S_1/m_1)(\#S_2/m_2)$; see, for example, Section 3.

**Unfocused enumeration.** The first method is to consider the possibilities $x = 1$, $x = 2$, $x = 3$, and so on, checking for each $x$ in turn whether $x \mod m_1 \in S_1$ and $x \mod m_2 \in S_2$.

The sets $S_1$ and $S_2$ can be represented in the obvious way as circular arrays of $m_1$ and $m_2$ bits respectively. There is very little work for each new $x$: check the next bit in each array, and record $x$ on the rare occasions that both bits are 1. Common general-purpose computers can check 32 or 64 values of $x$ simultaneously.

**Focused enumeration.** The second method is to generate, for each $r \in S_1$, the arithmetic progression of $x \in [1, H]$ such that $x \mod m_1 = r$, and then check for each $x$ successively whether $x \mod m_2 \in S_2$.

The advantage of focused enumeration over unfocused enumeration is that the number of operations drops from $H$ to about $H(\#S_1/m_1) + m_1$. The disadvantage is that $S_2$ is no longer checked sequentially.

**Doubly focused enumeration.** The third method uses the following special case of the explicit Chinese remainder theorem: every integer $x \in [1, H]$ may be written as a difference between a reasonably small multiple of $m_1$ and a reasonably small multiple of $m_2$. More precisely: $x$ may be written in the form $a_1 - a_2$, where $a_1$ is a multiple of $m_1$ in $[m_1, H + (m_1 - 1)m_2]$ and $a_2$ is a multiple of $m_2$ in $[0, (m_1 - 1)m_2]$. Notice that $x \mod m_1 \in S_1$ if and only if $-a_2 \mod m_1 \in S_1$, and $x \mod m_2 \in S_2$ if and only if $a_1 \mod m_2 \in S_2$.

Here is the algorithm. Enumerate, in increasing order, the multiples $a_1$ of $m_1$ in $[m_1, H + (m_1 - 1)m_2]$ such that $a_1 \mod m_2 \in S_2$. Simultaneously enumerate, in increasing order, the multiples $a_2$ of $m_2$ in $[0, (m_1 - 1)m_2]$ such that $-a_2 \mod m_1 \in S_1$. Merge these two lists to see all differences $a_1 - a_2$ in $[1, H]$.

The advantage of doubly focused enumeration over focused enumeration is that the number of operations drops from about $H(\#S_1/m_1) + m_1$ to, typically, about $H(\#S_1/m_1)(\#S_2/m_2) + m_1 + m_2$. The disadvantage is that each operation is fairly complicated: for example, a multidigit comparison.

This idea is so simple that it must have been written down before. However, I have not been able to locate it in the literature, and it is certainly not widely known in the context of enumerating locally square polynomial values.

**Further factorization.** In most applications, one can factor $m_1$ and $m_2$ into much smaller pieces, and correspondingly factor $S_1$ and $S_2$. Consider, for example, co-prime positive integers $m_{11}$ and $m_{12}$ and sets $S_{11}$ and $S_{12}$ such that $m = m_{11}m_{12}$ and $S_1 = \{ r : r \mod m_{11} \in S_{11}, r \mod m_{12} \in S_{12} \}$. 
For unfocused enumeration: One can store $S_{11}$ and $S_{12}$ instead of $S_1$, using $m_{11} + m_{12}$ bits of memory instead of $m_1$ bits of memory. One then checks bits of $S_1$ by checking the corresponding bits of $S_{11}$ and $S_{12}$.

For focused enumeration: One can enumerate values $r \in S_1$ more quickly than trying each $r \in \mathbb{Z}/m_1$, by applying the explicit Chinese remainder theorem to each pair in $S_{11} \times S_{12}$. This well-known technique again avoids the need to store $S_1$, and reduces the number of operations from about $H(\#S_1/m_1) + m_1$ to about $H(\#S_{11}/m_1) + \#S_1 + m_{11} + m_{12}$.

For doubly focused enumeration: The only extra difficulty is that values $r \in S_1$ need to be enumerated in increasing order. This can be done without much memory; see, e.g., [4].

3 Enumeration of locally square polynomial values

Fix a nonconstant squarefree polynomial $f$ in one variable over $\mathbb{Z}$. Consider the problem of finding all integers $x \in [1, H]$ such that $f(c + x)$ is locally square at all primes $p \leq h$, given a positive integer $H$, an integer $c$, and an integer $h \geq 2 \log H$.

Here is an algorithm for solving this problem. Select coprime positive integers $m_1$ and $m_2$ such that $p \leq h$ for every prime $p$ dividing $m_1 m_2$. Define $S_1$ as the set of $r \in \mathbb{Z}/m_1$ such that $f(c + r)$ is a square in $\mathbb{Z}/m_1$, and define $S_2$ as the set of $r \in \mathbb{Z}/m_2$ such that $f(c + r)$ is a square in $\mathbb{Z}/m_2$. Enumerate all integers $x \in [1, H]$ such that $x \mod m_1 \in S_1$ and $x \mod m_2 \in S_2$, as explained in Section 2. Check, for each such $x$, whether $f(c + x)$ is locally square at all primes $p \leq h$.

The rest of this section shows that this algorithm takes time $H/M^{2+o(1)}$ for an average $c$ under mild assumptions, if $m_1$ and $m_2$ are selected properly. Here $M = H^{1/\lg \log H}$, as in Section 1, and $o(1)$ is as $H \to \infty$ for $f$ fixed.

Assume for simplicity that $m_1$ and $m_2$ are squarefree; that they are each in $H/M^{2+o(1)}$; that each prime $p$ dividing $m_1 m_2$ is smaller than $2 \log H$; and that the number of $p$'s is in $(2+o(1))(\log H)/\log \log H = (2+o(1)) \lg M$. Theorem 3.1 below explains one way to construct $m_1$ and $m_2$ satisfying these conditions.

Assume also that $c$ has at most $M^o(1)$ digits and that $h$ is in $M^o(1)$. The basic operations in the algorithm—checking whether $f(c + x)$ is a square modulo various primes, or is locally square at various primes—then take time $M^{o(1)}$ with negligible memory. (One can speed up the algorithm by using more memory, as discussed in Section 2, and by choosing $m_1$ and $m_2$ somewhat larger. However, the speedup is only $M^{o(1)}$, which is not visible at the level of precision of this analysis.)

There are three bottlenecks in the algorithm:

- Checking whether $f(c + x)$ is locally square at all primes $p \leq h$, for each $x \in [1, H]$ such that $x \mod m_1 \in S_1$ and $x \mod m_2 \in S_2$, i.e., for each $x \in [1, H]$ such that $f(c + x)$ is a square modulo $m_1 m_2$. If $c$ is a uniform random element of an interval of length $m_1 m_2$ then the average number of candidates $x$ is at most $H/2^{(2+o(1)) \lg M} = H/M^{2+o(1)}$ by Theorem 3.3 below.
- Checking whether $a_1 \mod m_2$ is in $S_2$, for each multiple $a_1$ of $m_1$ between $m_1$ and $H + (m_1 - 1)m_2$. There are $H/M^{2+o(1)}$ multiples to check.
- Checking whether $-a_2 \mod m_1$ is in $S_1$, for each multiple $a_2$ of $m_2$ between 0 and $(m_1 - 1)m_2$. There are $H/M^{2+o(1)}$ multiples to check.

The total time is $H/M^{2+o(1)}$ if $c$ is a uniform random element of an interval of length $m_1 m_2 \approx H^2$. 
Theorem 3.1 Let \( H \) be a positive integer. Define \( u = \log H \); assume that 
\( 2u \geq 41 \). Define \( M = \exp(u/\log u) \). Let \( v \) be the largest integer such that the product 
m of the primes in \([1, v]\) satisfies \( m \leq H^2/M^4 \). Let \( v_1 \) be the largest integer such 
that the product \( m_1 \) of the primes in \([1, v_1]\) satisfies \( m_1 \leq H/M^2 \). Then there are 
\((2 + o(1))u/\log u\) prime divisors of \( m \); all prime divisors of \( m \) are smaller than \( 2u \); 
and both \( m_1 \) and \( m/m_1 \) are in \( H/M^{2+o(1)} \).

Proof The prime number theorem implies that \( v \) is in \((1 + o(1)) \log(H^2/M^4) = (2 + o(1))u \). Thus there are \((2 + o(1))u/\log 2u = (2 + o(1))u/\log u\) prime divisors of \( m \). Furthermore, \( m(v + 1) > H^2/M^4 \) by definition of \( v \), so \( m \) is in \( H^2/M^{4+o(1)} \); 
by the same argument, \( m_1 \) is in \( H/M^{2+o(1)} \); so \( m/m_1 \) is also in \( H/M^{2+o(1)} \).

Apply one of the Roeder-Schoenfeld theorems from [15]: the product of the 
primes in \([1, 2u]\) is larger than \( \exp(2u(1-1/\log 2u)) \) since \( 2u \geq 41 \). But \( \log 2u > 
\log u > (1/2) \log u \), so \( \exp(2u/\log 2u) < \exp(4u/\log u) = M^4 \); hence the product of the 
primes in \([1, 2u]\) is larger than \( H^2/M^4 \). Consequently \( v < 2u \). \( \Box \)

Theorem 3.2 Let \( d \) be a positive integer. Let \( f \) be a polynomial of degree \( d \) over \( \mathbb{Z} \). Let \( p \) be an odd prime number that does not divide the leading coefficient of \( f \) and 
does not divide the discriminant of \( f \). Then there are at most \((p + (d-1)\sqrt{p} + d)/2\) 
elements \( r \) of \( \mathbb{Z}/p \) such that \( f(r) \) is a square in \( \mathbb{Z}/p \).

Proof Define \( \chi : \mathbb{Z}/p \to \{-1, 0, 1\} \) as the Legendre symbol modulo \( p \). Define \( X_i = \{ r \in \mathbb{Z}/p : \chi(f(r)) = i \} \). Define \( a \) as the leading coefficient of \( f \). Define \( g \) as the 
polynomial \( a^{-1}f \) over the field \( \mathbb{Z}/p \).

Check the conditions of Weil’s theorem, as stated in [11, Theorem 5.41]: \( \chi \) is 
a multiplicative character of \( \mathbb{Z}/p \) of order 2; \( g \) is a monic polynomial over \( \mathbb{Z}/p \) of 
positive degree (namely, degree \( d \)); \( g \) is not a square, because otherwise \( p \) would 
divide the discriminant of \( f \); and \( g \) has at most \( d \) (in fact, exactly \( d \)) distinct roots 
in its splitting field over \( \mathbb{Z}/p \).

Conclusion: \( X_1 - X_{-1} = \sum_r \chi(f(r)) = \sum_r \chi(g(r)) \) is at most \((d-1)\sqrt{p} \). But 
\( X_1 + X_0 + X_{-1} = \) exactly \( p \); and \( X_0 \) is exactly the number of roots of \( f \) in \( \mathbb{Z}/p \), 
which is at most \( d \). Add: \( 2X_1 + 2X_0 \leq p + (d-1)\sqrt{p} + d \). \( \Box \)

Theorem 3.3 Let \( f \) be a nonconstant squarefree polynomial over \( \mathbb{Z} \). Then 
there is a function \( \epsilon : \mathbb{N} \to \mathbb{R} \) with \( \epsilon \in o(1) \), such that \( f(r) \) is a square in \( \mathbb{Z}/m \) 
with probability at most \( 2^{-1+\epsilon(k)k} \) if \( m \) is a squarefree positive integer, \( k \) is the 
number of prime divisors of \( m \), and \( r \) is a uniform random element of \( \mathbb{Z}/m \).

Proof Write \( d = \deg f \geq 1 \). Note that the discriminant of \( f \) is nonzero. Define \( \alpha \) as 
the maximum of the following quantities: \( d; \sqrt{2}; \sqrt{p} \) for the primes \( p \) dividing 
the discriminant of \( f \); and \( \sqrt{\sqrt{p}} \) for the primes \( p \) dividing the leading coefficient of \( f \).

I claim that if \( p \) is prime and \( r \mod p \) is uniform then \( f(r) \) is a square modulo \( p \) 
with probability at most \((1 + \alpha/\sqrt{\sqrt{p}})/2 \). Indeed, if \( p > d^2 \) is an odd prime 
that divides neither the discriminant of \( f \) nor the leading coefficient of \( f \), then the 
probability is at most

\[
\frac{1}{2} \left( 1 + \frac{d-1}{\sqrt{\sqrt{p}}} + \frac{d}{p} \right) < \frac{1}{2} \left( 1 + \frac{d}{\sqrt{\sqrt{p}}} \right) \leq \frac{1}{2} \left( 1 + \frac{\alpha}{\sqrt{\sqrt{p}}} \right)
\]

by Theorem 3.2. Otherwise \( \sqrt{\sqrt{p}} \leq \alpha \) by definition of \( \alpha \); the probability is at most 
\( 1 \leq (1 + \alpha/\sqrt{\sqrt{p}})/2 \).
The probability that $f(r)$ is a square modulo $m$ is the product, over the primes $p$ dividing $m$, of the probability that $f(r)$ is a square modulo $p$; which is, in turn, at most the product of $(1 + \alpha/\sqrt{p})/2$ for these $k$ primes $p$; which is, in turn, at most $(1 + \alpha/\sqrt{2})(1 + \alpha/\sqrt{3})(1 + \alpha/\sqrt{4})\cdots(1 + \alpha/\sqrt{k+1})/2^k$; in other words, at most $2^{-(1+\epsilon(k))k}$, where $\epsilon(0) = 0$ and
\[
\epsilon(k) = \frac{1}{k} \left( \log \left(1 + \frac{\alpha}{\sqrt{2}}\right) + \log \left(1 + \frac{\alpha}{\sqrt{3}}\right) + \cdots + \log \left(1 + \frac{\alpha}{\sqrt{k+1}}\right) \right)
\]
for $k \geq 1$. The sum $\log(1+\alpha/\sqrt{2})+\cdots+\log(1+\alpha/\sqrt{k+1}) \leq \alpha/\sqrt{2}+\cdots+\alpha/\sqrt{k+1}$ is bounded by a multiple of $\sqrt{k+1}$, so $\epsilon(k) \to 0$ as $k \to \infty$. \hfill \Box

4 Example: locally square integers

Let $x$ be a positive non-square integer in $1 + 8\Z$. What is the smallest odd prime $p$ such that $x^{(p-1)/2} \mod p \neq 1$? In other words, what is the smallest odd prime $p$ such that $x$ is divisible by $p$ or locally non-square at $p$ or both?

It is widely conjectured that $p/\log p$ is at most $(1 + o(1)) \log x$, where $\log y = \log_2 y$. In fact, no examples are known in which $p/\log p$ is larger than $\log x$. A proof of an explicit bound such as $p/\log p \leq 2\log x$ would imply, among other things, that there is a deterministic primality-proving algorithm taking essentially cubic time. See [7], [2], [13, pages 130–136], and [14, Section 2].

I have verified that $p \leq 283$ for all $x < 24 \cdot 2^{64} \approx 4.4 \cdot 10^{20}$. Here are the cutoffs for each $p$ between 149 and 281 inclusive:

- $p \leq 149, \quad p/\log p < 29.777$ if $x < 2625087023729 \approx 1.492 \cdot 2^{44}$
- $p \leq 157, \quad p/\log p < 31.601$ if $x < 112434732901969 \approx 1.598 \cdot 2^{46}$
- $p \leq 173, \quad p/\log p < 33.571$ if $x < 17893622537081 \approx 1.271 \cdot 2^{47}$
- $p \leq 181, \quad p/\log p < 34.818$ if $x < 696161110200049 \approx 1.237 \cdot 2^{49}$
- $p \leq 193, \quad p/\log p < 36.674$ if $x < 285490968103881 \approx 1.268 \cdot 2^{51}$
- $p \leq 197, \quad p/\log p < 37.288$ if $x < 6450045516630769 \approx 1.432 \cdot 2^{52}$
- $p \leq 211, \quad p/\log p < 39.426$ if $x < 1164139247947921 \approx 1.292 \cdot 2^{53}$
- $p \leq 227, \quad p/\log p < 41.844$ if $x < 190621428905186449 \approx 1.323 \cdot 2^{57}$
- $p \leq 229, \quad p/\log p < 42.145$ if $x < 196640248121928601 \approx 1.364 \cdot 2^{57}$
- $p \leq 233, \quad p/\log p < 42.745$ if $x < 712624335095093521 \approx 1.236 \cdot 2^{59}$
- $p \leq 239, \quad p/\log p < 43.642$ if $x < 1773855791877850321 \approx 1.539 \cdot 2^{60}$
- $p \leq 241, \quad p/\log p < 43.940$ if $x < 2327687064124474441 \approx 1.009 \cdot 2^{61}$
- $p \leq 251, \quad p/\log p < 45.427$ if $x < 6384991873059836689 \approx 1.385 \cdot 2^{62}$
- $p \leq 257, \quad p/\log p < 46.315$ if $x < 8019204661305419761 \approx 1.739 \cdot 2^{62}$
- $p \leq 263, \quad p/\log p < 47.199$ if $x < 10198100582046287689 \approx 1.106 \cdot 2^{63}$
- $p \leq 277, \quad p/\log p < 49.254$ if $x < 69848283200186969 \approx 1.893 \cdot 2^{65}$
- $p \leq 281, \quad p/\log p < 49.838$ if $x < 20893636579044975961 \approx 1.416 \cdot 2^{67}$
- $p \leq 283, \quad p/\log p < 50.129$ if $x < 24 \cdot 2^{64} = 1.5 \cdot 2^{68}$ (not maximal)

This computation took ten days, about $1.2 \cdot 10^{15}$ clock cycles, on a Pentium 4 running at 1400MHz.

Another way to phrase the same result: Every non-square integer below $24 \cdot 2^{64}$ is locally non-square at some prime in $\{2, 3, \ldots, 283\}$. Indeed, write the integer in the form $xy^2$ where $x$ is squarefree. If $x$ is a square then the original integer is a square, contradiction. If $x \not\equiv 1 \mod 8$ then $xy^2$ is locally non-square at 2. If $x$ is a non-square in $1 + 8\Z$ then, by this computation, there is an odd prime $p \leq 283$ for which $x^{(p-1)/2} \mod p \neq 1$. If $x^{(p-1)/2} \mod p = p - 1$ then $x$ is
locally non-square at \( p \). If \( x^{(p-1)/2} \mod p = 0 \) then \( x \) is divisible by \( p \) but, being squarefree, not by \( p^2 \), so it is locally non-square at \( p \).

A series of previous computations, initiated by Kraitchik in 1924 and continued by Lehmer, Lehmer, Shanks, Patterson, Williams, Stephens, and Lukes, showed with considerably more effort that \( p \leq 281 \) for all \( x \) up to about \( 7 \cdot 10^{19} \approx 2^{66} \). See [8], [9], [10], [16], [13, page 134], and [14]. For example, the computation of Lukes, Patterson, and Williams in [14] was a focused enumeration of all small \( y \) such that \( 1 + 24y \) is a non-unit square modulo \( m_1 = 5 \cdot 7 \cdot 11 \cdot 13 \); there are about \( H/27 \) such values of \( y \) in \([1, H]\).

My computation was a doubly focused enumeration, as explained in Section 2 and Section 3, of all small \( y \) such that \( 1 + 24y \) is a non-unit square modulo both \( m_1 = 41 \cdot 43 \cdot 47 \cdot 53 \cdot 59 \cdot 61 \cdot 71 \cdot 73 \) and \( m_2 = 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 67 \). There are about \( H/1386487 \) such values of \( y \) in \([1, H]\).

Further speed improvements are possible. I could use somewhat larger moduli \( m_1 \) and \( m_2 \), for example, especially if I balance the primes.

**A note on terminology: pseudosquares.** Lehmer in [9] defined, for each prime \( q \geq 3 \), the corresponding “pseudo-square” \( L_q \) as the smallest positive non-square integer \( x \in 1 + 8Z \) for which \( p > q \). The same terminology is used in [17, page 522], [13], and [14].

On the other hand, Lehmer, Lehmer, and Shanks in [10] defined a “pseudo-square” for \( q \) as any non-square integer \( x \in 1 + 8Z \) for which \( p > q \). The same terminology is used in [5]. The “Table of Pseudo-Squares” in [10, page 434] includes one “least solution” column and one “least prime solution” column. A “prime pseudo-square” in this terminology does not have a short name in the previous terminology.

“Pseudo-squares” have an unrelated definition in [1] and [3]: a sequence of “pseudo-squares” is a sequence of integers in which the \( n \)th integer is close to \( n^2 \) in the usual metric.

### 5 Example: locally square values of \( x^3 + y^7 \)

The problem here is to find all coprime integer pairs \( (x, y) \) with \( y \in [1, 1000] \) and \( x \in [-y^3, 10^{17} - 1 - y^3] \) such that \( x^3 + y^7 \) is locally square at all primes \( p \leq 251 \). This is inspired by the well-known problem of finding all coprime integer pairs \( (x, y) \) for which \( x^3 + y^7 \) is a square.

Define \( m = 3 \cdot 5 \cdot 7 \cdot 8 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \cdot 53 \cdot 59 \cdot 61 \cdot 67 \approx 3.1433 \cdot 10^{25} \). For each \( y \) separately, I am doing a doubly focused enumeration of all \( x \in [-y^3, 10^{17} - 1 - y^3] \) such that \( x \) is coprime to \( \gcd \{y, m\} \) and \( x^3 + y^7 \) is a square modulo \( m \). This enumeration is producing only a small fraction of the original \( 10^{20} \) pairs \( (x, y) \). I am then checking, for each enumerated \( x \), whether \( x^3 + y^7 \) is locally square at all primes \( p \leq 251 \), and whether \( x \) is coprime to \( y \).

A subsequent draft of this paper will report the results.

Previously known results: \( 1^3 + 0^7 = 1^2 \); \((-1)^3 + 1^7 = 0^2 \); \( 0^3 + 1^7 = 1^2 \); \( 2^3 + 1^7 = 3^2 \); \( 17^3 + 2^7 = 71^2 \); \( 76271^3 + 17^7 = 21063928^2 \); \((-1414)^3 + 65^7 = 2213459^2 \); \((-9262)^3 + 113^7 = 15312283^2 \). The last three were discovered in 1993 by Beukers and Zagier.

I should look at a wider range of \( y \)'s, including some negative \( y \)'s.
References


