Finding Small Solutions to Small Degree Polynomials

Don Coppersmith

IBM Research, T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
copper@watson.ibm.com

Abstract. This talk is a brief survey of recent results and ideas concerning the problem of finding a small root of a univariate polynomial mod N, and the companion problem of finding a small solution to a bivariate equation over $\mathbb{Z}$. We start with the lattice-based approach from [2],[3], and speculate on directions for improvement.
Keywords: Modular polynomials, lattice reduction

1 Univariate modular polynomial

Our basic setup is a univariate polynomial $p(x)$ of small degree d, and a modulus N of unknown factorization. For a suitable bound B, we wish to find all integers x_0 such that $|x_0| < B$ and $p(x_0) = 0 \bmod N$. (Call such integers "*small roots*".) Our efforts will be concentrated in increasing the bound B.

An early paper in this line of research was [2], but the author was working with an unnatural space. Here we follow the more natural presentation of Howgrave-Graham [8].

For simplicity we assume p is monic, although we really only need that the gcd of its coefficients be relatively prime to N; see Remark 1 below. We set

$$p(x) = x^d + p_{d-1}x^{d-1} + \cdots + p_2x^2 + p_1x + p_0.$$

The first approach is essentially due to Håstad [7]: Consider the collection C_1 of polynomials:

$$C_1 = \{x^i, 0 \le i < d\} \cup \{p(x)/N\}.$$

For each polynomial $q \in C_1$, for each small root x_0, we see that $q(x_0)$ is an integer. The same will be true of any integer linear combination of polynomials in C_1.

So it makes sense to consider the lattice of dimension $d+1$ generated by the *columns* of the matrix

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & p_0/N \\ 0 & B & 0 & \cdots & 0 & 0 & p_1B/N \\ 0 & 0 & B^2 & \cdots & 0 & 0 & p_2B^2/N \\ & & & \vdots & & & \\ 0 & 0 & 0 & \cdots & B^{d-2} & 0 & p_{d-2}B^{d-2}/N \\ 0 & 0 & 0 & \cdots & 0 & B^{d-1} & p_{d-1}B^{d-1}/N \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1B^d/N \end{bmatrix}.$$

Each column corresponds to a polynomial $q(x)$ in C_1, expressed in the basis $\{x^i/B^i\}$. So row i corresponds to the coefficient of x^i in $q(x)$, multiplied by a scaling factor B^i.

Now apply lattice basis reduction [13]. Because L_1 has dimension $d+1$ and determinant $N^{-1}B^{d(d+1)/2}$, we will find a nonzero vector $\mathbf{v}$ of length

$$|\mathbf{v}| < c_1(d)\,(\det L_1)^{1/(d+1)} = c_1(d)N^{-1/(d+1)}B^{d/2},$$

where $c_1(d)$ is a constant depending only on the dimension.

We interpret the vector $\mathbf{v} = [v_0, v_1B, v_2B^2, \ldots, v_dB^d]$ as a polynomial $v(x) = \sum v_i x^i$, again expressing $v(x)$ in the basis $\{x^i/B^i\}$.

Suppose we know that

$$c_1(d)\,(\det L_1)^{1/(d+1)} < \frac{1}{d+1},$$

or equivalently,

$$B \le c_1'(d)N^{2/[d(d+1)]},$$

where $c_1'(d)$ is another constant depending only on d. Then we also know that $|\mathbf{v}| < 1/(d+1)$, and each entry of $\mathbf{v}$ satisfies $|v_iB^i| < 1/(d+1)$. Evaluate the polynomial $v(x)$ at a small root x_0. On one hand

$$|v(x_0)| \le \sum |v_i x_0^i| \le \sum |v_iB^i| < \sum \frac{1}{d+1} = 1,$$

so that $|v(x_0)| < 1$. On the other hand, $v(x_0)$ is an integer. These two conditions together imply that $v(x_0) = 0 \in \mathbb{Q}$.

In summary: We have computed a polynomial $v(x)$ in $\mathbb{Q}[x]$ whose roots include all "small roots" x_0, that is, all those x_0 with $|x_0| < B = c_1'(d)N^{2/d(d+1)}$ and with $p(x_0) = 0 \bmod N$.

Incidentally, we have also bounded the number of small roots, by $\dim L_1$.

Remark 1: If $p(x)$ is not monic, but its content is relatively prime to N, we augment C_1 with x^d, that is, we replace C_1 by

$$C_1' = \{x^i, 0 \le i \le d\} \cup \{p(x)/N\}.$$

One checks that the corresponding lattice L_1' still has dimension $d+1$ and determinant $N^{-1}B^{d(d+1)/2}$, and the rest of the argument goes through.

Remark 2: Using the Cauchy-Schwarz inequality, one can replace the condition

$$c_1(d)\,(\det L_1)^{1/(d+1)} < \frac{1}{d+1}$$

by the weaker condition

$$c_1(d)\,(\det L_1)^{1/(d+1)} < \frac{1}{\sqrt{d+1}}.$$

Remark 3: Sometimes we scale thing differently, using $C_1'' = \{Nx^i, 0 \le i < d\} \cup \{p(x)\}$ and using the fact that for each $q \in C_1''$ and small root x_0 we have that $q(x_0)$ is a multiple of N. The two approaches are numerically equivalent, the only difference being aesthetics.

2 Improvements in the exponent

The first improvement comes when we consider a larger collection of polynomials. Define

$$C_2 = \{x^i, 0 \le i < d\} \cup \{(p(x)/N)x^i, 0 \le i < d\}.$$

The corresponding lattice L_2 has dimension $2d$ and determinant $N^{-d}B^{2d(2d-1)/2}$. The enabling condition becomes

$$c_2(d)\left(N^{-d}B^{2d(2d-1)/2}\right)^{1/(2d)} < \frac{1}{2d},$$

or equivalently

$$B \le c_2'(d)N^{1/(2d-1)}.$$

The exponent of N has improved from $2/d(d+1)$ to $1/(2d-1)$. The improvement came about because the dimension of L increased, while its determinant decreased.

We obtain a second improvement by considering higher powers of the modulus N, along with a still larger collection of polynomials. Fix a positive integer h. Define

$$C_3 = \{(p(x)/N)^j x^i, 0 \le i < d, 0 \le j < h\}.$$

For each polynomial $q(x) \in C_3$, for each small root x_0, we see that $q(x_0)$ is an integer. Again the same holds for any integer linear combination of polynomials in C_3.

The corresponding lattice L_3 has dimension dh and determinant

$$N^{-dh(h-1)/2}B^{(dh)(dh-1)/2}.$$

(The powers of N on the diagonal consist of d each of $N^0, N^{-1}, \ldots, N^{-(h-1)}$, while the powers of B are $B^0, B^1, \ldots, B^{dh-1}$.)

Our enabling equation is now:

$$\left[N^{-dh(h-1)/2}B^{(dh)(dh-1)/2}\right]^{1/(dh)} < c_3(d,h),$$

which will be satisfied if

$$B \le c_3'(d,h)N^{(h-1)/(dh-1)}.$$

The exponent of N, namely $\frac{h-1}{dh-1}$, differs from $\frac{1}{d}$ by $\frac{d-1}{d(dh-1)} < \frac{1}{dh}$; this difference can be made arbitrarily small by choosing h larger, at the expense of computational complexity. Put another way, we achieve a bound

$$B = c_3''(d,\epsilon)N^{(1/d)-\epsilon}$$

by choosing $h = O\left(\frac{1}{d\epsilon}\right)$. The running time is polynomial in $(d, 1/\epsilon, \log N)$.

We can extend the bound to $N^{1/d}$ by breaking the interval of size $2N^{1/d}$ into N^ϵ intervals of size $2N^{(1/d)-\epsilon}$. This is still polynomial time, but in practice it gets much more expensive as the exponent gets closer to $1/d$.

As before, we have bounded the number of small roots, as well as showing how to compute them all in polynomial time. The existential results match those of Konyagin and Steger [12], who bound the number of small roots by

$$O\left(\frac{1+\log B}{\log(1+B^{-1}N^{1/d})}\right).$$

For $B = N^{(1/d)-\epsilon}$ their bound is essentially $O\left(\frac{1}{d\epsilon}\right)$, while for $B = N^{(1/d)+\epsilon}$ their bound becomes $O\left(\frac{N^{\epsilon}\log N}{d}\right)$.

3 Minor improvements

We can improve the lower order factor—the $c(d)$ factor in the estimate of the bound B—by more careful consideration of the process.

One idea, due independently to Hendrik Lenstra [15] and to Nick Howgrave-Graham [9], is to recognize that, for each positive integer k, the rational polynomial $b_k(x) = x(x-1)\cdots(x-k+1)/k!$ takes on integer values for all integer arguments x. So we augment (say) the collection of polynomials

$$C_3 = \{(p(x)/N)^j x^i, 0 \le i < d, 0 \le j < h\},$$

with the polynomials $b_k(x), 0 \le k < dh$, that is

$$C_3' = \{(p(x)/N)^j x^i, 0 \le i < d, 0 \le j < h\} \cup \{b_k(x), 0 \le k < dh\}.$$

(Assume here that N is free of small factors, that is, N is relatively prime to $(dh-1)!$.) Whereas L_3 could be represented by an upper triangular matrix whose kth diagonal element is $B^k N^{-\lfloor k/d \rfloor}$, one finds that L_3' can be represented by an upper triangular matrix whose kth diagonal element is $(1/k!)B^k N^{-\lfloor k/d \rfloor}$. This decreases $\det(L_3)$ by a factor

$$\prod_{0 \le k < dh} k!.$$

This allows us to increase B in compensation, by a factor

$$\left(\prod_{0 \le k < dh} k!\right)^{2/[(dh)(dh-1)]} \approx \frac{dh}{e^{3/2}} \approx \frac{dh}{4.5}.$$

Phong Nguyen [17] reports that in practice this does speed up computations, by perhaps a factor of 5. Nguyen also remarks that one could further augment C_3' with the polynomials $\{b_j(p(x)/N)b_i(x)\}$, but that one does not thereby change the lattice L_3'.

Another idea, developed here in its explicit form but related to earlier work by Boneh [1], is probably less profitable. We have used the fact that the monomials

$(x/B)^i$ are bounded by 1 when $|x| < B$. The Chebyshev polynomials [18] share that property, but more efficiently. These polynomials are defined by:

$$T_k(\cos\theta) = \cos(k\theta),$$

$$T_k(x) = 2^{k-1}x^k + \text{smaller terms}(k \geq 1).$$

Where we currently use the monomial basis—row i corresponds to $(x/B)^i$—to express $q(x) \in C$ as a column of L, we can instead use the Chebyshev basis—row i corresponds to $T_i(x/B)$. This will decrease $\det(L)$ by a factor

$$2^{0+0+1+2+\cdots+(dh-2)} = 2^{(dh-1)(dh-2)/2},$$

leading to an increase in B by a factor of

$$2^{(dh-2)/(dh)} \approx 2.$$

The two ideas can be applied simultaneously, and the improvements accumulate. But they increase B by only a polynomial factor, and therefore improve running time only by that factor. The same effect could be achieved by solving several polynomials $p(x - 2iB), |i| \leq k/2$ over the range $|x| < B$ and using the result to solve the single polynomial $p(x)$ over the larger range $|x| < kB$.

4 Speculative improvement of exponent

Can we improve the asymptotic bound $B = N^{1/d}$? The bound is a natural-looking bound, and it matches the existential results of Konyagin and Steger [12]. Indeed, even in the simple case $p(x) = x^3 - A \pmod{N}$, we don't know an efficient way of finding roots x larger than $B = N^{1/3}$, while those smaller than $B = N^{1/3}$ are trivially found by solving $x^3 - A = 0$ over the integers.

The following example gives cause for pessimism. Set $N = q^3$ with q prime, and set $p(x) = x^3 + Dqx^2 + Eq^2x$ with $D, E \in \mathbb{Z}$. Clearly if x_0 is any multiple of q then $p(x_0) = 0 \pmod{N}$. So if we select a bound $B = N^{(1/3)+\epsilon}$, the number of "small roots" x_0 with $|x_0| < B$ and $p(x_0) = 0 \pmod{N}$ is about $2N^\epsilon$, i.e. exponentially many. (Again this essentially matches the bound of Konyagin and Steger [12].) Our lattice techniques cannot hope to find them, since in our setup all the small roots are roots of $v(x)$, so that the number of small roots needs to be bounded by $\dim(L)$.

More generally, we can expect trouble whenever $q^2|N$ and $p(x)$ has a repeated root modulo q. (We don't know whether this family contains all the polynomials with an exponentially large number of roots smaller than $N^{(1/d)+\epsilon}$.) When this happens, we know that N shares a common factor with the discriminant of $p(x)$,

$$\mathrm{Discr}(p) = \mathrm{Res}(p, p'),$$

$$\gcd(N, \mathrm{Discr}(p)) > 1.$$

So any improvement of the exponent past $1/d$ must somehow rule out this case.

With this in mind, we hypothesize a *"Discriminant Attack"*:

Suppose we can guarantee that we are never in the unfavorable situation. We can demand that $\gcd(N, \mathrm{Discr}(p)) = 1$. Equivalently, we can demand existence of $D(x), E(x) \in \mathbb{Z}[x]$ and $F \in \mathbb{Z}$ satisfying

$$D(x)p(x) + E(x)p'(x) + FN = 1;$$

if $\gcd(N, \mathrm{Discr}(p)) = 1$, then $D(x), E(x), F$ exist and are easily computed. Perhaps D, E, F can be incorporated into the construction of the lattice L, in such a way that the bound B can be improved to $N^{(1/d)+\epsilon}$. But I don't see how to do it.

A related effort is the *"Divided Difference Attack"*:

Suppose we know that there are *two* small roots x, y, which differ modulo each prime factor of N; that is, $\gcd(N, x - y) = 1$. Then besides the modular equations of degree d,

$$p(x) = 0 \pmod{N},$$
$$p(y) = 0 \pmod{N},$$

we get a third equation of total degree $d - 1$:

$$r(x, y) = \frac{p(x) - p(y)}{x - y} = 0 \pmod{N}.$$

(Despite its appearance, $r(x, y)$ is actually an integer polynomial.)

We are now dealing with *bivariate* modular polynomials. As mentioned in [2] and in Section 6 below, our techniques can sometimes handle this, but there are no guarantees. Let's try an example and see.

Select a positive integer h. Let the family of polynomials be

$$C = \{(p(x)/N)^k (r(x, y)/N)^\ell x^i y^m, k + \ell \le h, 0 \le i < d, 0 \le m < d - 1\}.$$

(We are not using $p(y)$ because it is already generated by $p(y) = p(x) + (y - x)r(x, y)$.) The polynomials of C are related to their monomial basis, which is *nearly*

$$\{x^a y^b; \quad a, b \ge 0; \quad \frac{a}{d} + \frac{b}{d-1} \le h + 1\}.$$

where B_x, B_y are the bounds on $|x|$ and $|y|$. The number of such monomials is roughly the area of a triangle, namely $A = [d(h + 1)][(d - 1)(h + 1)]/2$. (We are ignoring inaccuracies due to edge effects near the hypotenuse of the triangle.) Associate with each polynomial $q \in C$ the monomial corresponding to its leading term (in reverse lexicographical order). Estimate that the average exponent x (in the monomial basis) is about $d(h + 1)/3$, so that the average exponent of $p(x)$ (among $q \in C$) is about $h/3$. Similarly the average exponent of $r(x, y)$ is about $h/3$, and the average exponent of N is about $-2h/3$. Build the lattice L as before, incorporating the scaling factors B_x, B_y (the bounds on $|x_0|$ and $|y_0|$). We estimate the determinant of L as

$$B_x^{Ad(h+1)/3} B_y^{A(d-1)(h+1)/3} N^{-2Ah/3}.$$

The "enabling condition" then becomes

$$B_x^{Ad(h+1)/3} B_y^{A(d-1)(h+1)/3} N^{-2Ah/3} < c,$$

$$B_x^{d(h+1)} B_y^{(d-1)(h+1)} < c' N^{2h},$$

and in the limit of large h with d held fixed,

$$B_x^d B_y^{d-1} < c'' N^2.$$

If the enabling condition is satisfied, we will obtain an equation $v_1(x, y)$ in $\mathbb{Z}[x, y]$ relating x_0 and y_0 for all small pairs (x_0, y_0) satisfying our original modular equations p and r. But a single equation is not enough to solve for x_0 and y_0. We have to hope that two independent equations are generated. Indeed, some work by Charanjit Jutla [11] indicates that under certain conditions we can guarantee that at least two equations $v_1(x, y), v_2(x, y)$ will be generated, both with coefficients small enough that they hold in $\mathbb{Z}$. If v_1, v_2 are algebraically independent, then we can solve them by using the resultant: $u(y) = \mathrm{Res}(v1, v2; x) \in \mathbb{Z}[y]$ is a univariate equation in $\mathbb{Z}$ whose roots contain all y_0 that participate in any small pair (x_0, y_0) of interest. For each y_0 we can then easily find all the corresponding x_0.

But we cannot always guarantee that the two equations will be algebraically independent. One can be a multiple of the other, in which case the resultant will be 0 and we will learn nothing.

Let's examine the limit of the enabling equation:

$$B_x^d B_y^{d-1} < c'' N^2.$$

If either root were below $N^{1/d}$, the standard methods would find it. But the present method may work when both roots are in the narrow range:

$$N^{1/d} < |x_0|, |y_0| < N^{1/(d-1)},$$

so that we obtain a slight advantage.

We have tried to abuse this method to obtain information that should otherwise be hard to get, and we always fail. Here are some examples.

Suppose we know a root x_0 with $|x_0| < N^{1/d}$, and we want to find a second small root y_0 with $\gcd(N, x_0 - y_0) = 1$. The straightforward approach would be to divide $p(y)$ by $y - x_0 \pmod N$ to obtain a polynomial of degree $d - 1$. Our usual method will solve this as long as $|y_0| < N^{1/(d-1)}$. But using the bivariate method, we might expect to be able to find y_0 as long as

$$|x_0^d y_0^{d-1}| < c'' N^2.$$

Since $|x_0| < N^{1/d}$, it seems superficially that we can allow $|y_0| > N^{1/(d-1)}$ and still satisfy our condition. But when we try it, the equations $v_i(x, y) = 0$ that we recover always involve multiples of $x - x_0 = 0$, giving no information about y.

As a second example, suppose $p(x)$ is of degree 2, and we are told there are two independent small roots x_0 and y_0, both of size about $N^{2/3}$. So

$$p(x) = x^2 + Ax + B = 0 \pmod{N}.$$

Since the roots satisfy

$$|x_0^d y_0^{d-1}| = |x_0^2 y_0^1| \approx N^2,$$

the present method should apply. But again the equations it gives are useless: multiples of

$$x_0 + y_0 + A = 0,$$

which we could have gotten from the original equation by inspection. Since both roots are small, A is also bounded by about $N^{2/3}$, and the equation $x_0+y_0 = -A$ can be taken to hold in $\mathbb{Z}$.

But there may exist situations where this bivariate approach gives answers that we could not otherwise obtain.

Remark 3: One relation between the "discriminant attack" and the "divided difference attack" is the usage of $p'(x)$ in the former case and $\frac{p(x)-p(y)}{x-y}$ in the latter case. Where the divided difference attack treats the circumstance that there *exist* two (small) roots that differ modulo each prime factor of N, the discriminant attack demands that there *not exist* repeated roots, or in some sense that all roots are different mod q.

5 Bivariate integer polynomials

The present author [3] applied techniques similar to those of his other paper [2], to the problem of finding a small solution to a bivariate integer polynomial

$$p(x, y) = 0 \in \mathbb{Z}.$$

The primary application was to integer factorization when half the bits of one of the factors are known.

The presentation in [3] is difficult to understand. Once again Howgrave-Graham's presentation makes it more accessible, but this simplification seems to only apply to the specific equation describing integer factorization (and some related equations), and not to the general bivariate integer polynomial.

For integer factorization, suppose we have an integer N of unknown factorization $N = PQ$, where we have some approximation to the factors P and Q. We can write

$$N = (P' + x_0)(Q' + y_0)$$

where x_0, y_0 are small. If $P = N^\beta$ and $Q = N^{1-\beta}$, we will be able to solve this as long as

$$|x_0| < B_x = N^{\beta^2}, |y_0| < B_y = N^{(1-\beta)^2}.$$

Notice that in this setup, we know the $\beta(1-\beta)\log N$ high order bits of P, which is equivalent to knowing the $\beta(1-\beta)\log N$ high order bits of Q.

Select positive integers h, k with $h < k$. Define the family of polynomials

$$C = \{N^{h-i}(P' + x)^i, 0 \le i < h\} \cup \{x^{i-h}(P' + x)^h, h \le i < k\}.$$

These polynomials, when evaluated at $x = x_0$, are all multiples of the (unknown) integer P^h.

The corresponding lattice L has dimension k and determinant $N^{h(h+1)/2} B_x^{k(k-1)/2}$. The enabling equation is then

$$\left(N^{h(h+1)/2} B_x^{k(k-1)/2}\right)^{1/k} \le cP^h = cN^{\beta h},$$

$$B_x \le c' N^{h(2k\beta - h - 1)/[k(k-1)]}.$$

For large h, k, we optimize this by selecting $h = k\beta$, obtaining

$$B_x \approx N^{\beta^2}.$$

The rest of the development is similar to the univariate modular case. We find an equation in $\mathbb{Z}[x]$, whose roots include the root x_0 of interest.

Howgrave-Graham [10] develops these techniques even further, applied to the problem of an "approximate gcd", finding a gcd when the inputs are only approximately known.

The same technique can be applied, almost without change, to the problem of divisors in residue classes. Hendrik Lenstra [14] showed that if $0 < r < s < N$ are given positive integers, pairwise relatively prime, with $s > N^\alpha$ and $\alpha > 1/4$, the number of divisors of N equivalent to r mod s is bounded by a function of α, independent of r, s, N. Applying the present techniques we not only recover that existential bound but actually construct those divisors in polynomial time. [5]

For more general bivariate integer equations, the reader is referred to the author's earlier work [3], [4], where the development is less intuitive but handles a more general situation.

The reader may also enjoy the more recent work of Elkies [6], the techniques being closely related. Elkies treats a more general setting, where instead of lattice points *on* a curve, he is looking for lattice points *near* a curve. He finds much wider applicability, including non-algebraic curves.

6 Possible (and impossible) extensions

We can sometimes extend these techniques, to bivariate modular equations or to multivariate integer equations. But the extensions are not guaranteed to work, and in fact there are impossibility results that argue against their application to the general case.

Consider the bivariate modular case. As in the example above, we can easily build a lattice consisting of multiples of N and of $p(x, y)$ (or of their powers), and we can find a short vector in that lattice, corresponding to a polynomial in

$\mathbb{Z}$ satisfied by all small roots. But this polynomial will, in general, be difficult to solve.

As in the example, we can hope to find two short vectors, corresponding to two polynomials, and we can hope that they are algebraically independent, so that taking their resultant we can recover a single polynomial in a single variable, whose roots include all those y_0 belonging to a short pair of roots (x_0, y_0). The trouble is that although we can arrange things so that two short vectors will be found, we cannot in general guarantee that the corresponding polynomials will be algebraically independent.

We start with the following theorem of Manders and Adleman [16]:

Theorem 1. *(Manders and Adleman) The (problem of accepting the) set of Diophantine equations (in a standard binary encoding) of the form*

$$\alpha x_1^2 + \beta x_2 - \gamma = 0$$

which have natural-number solutions x_1, x_2 is NP-complete.

Manders and Adleman go on to remark that the problem remains NP-complete even when β is given in fully factored form.

We need to make minor adjustments to use this theorem. Let us first center the range of x_2. Let δ approximate half of its range:

$$\delta = \left\lfloor \frac{\gamma}{2\beta} + \frac{1}{2} \right\rfloor,$$

and define

$$\begin{aligned} x &= x_1, \\ y &= x_2 - \delta, \\ \tau &= \gamma - \beta\delta. \end{aligned}$$

Then we are asking for existence of solutions (x_0, y_0) to

$$\alpha x^2 + \beta y - \tau = 0,$$

with $|x_0| < B_x \approx \sqrt{\gamma/\alpha}$ and $|y_0| < B_y \approx \delta$. (Clearly if we can compute all small solutions (x_0, y_0) within these bounds, we can decide whether exact solutions to the original problem exist.)

Now select N arbitrarily large, as long as N exceeds $|\alpha B_x^2| + |\beta B_y| + |\tau|$. Given the bivariate modular equation

$$\alpha x^2 + \beta y - \tau = 0 \pmod{N},$$

and bounds B_x, B_y as before, it will be hard to decide whether there are small solutions (x_0, y_0); the reduction mod N is meaningless. Now, the bounds B_x, B_y stay fixed as N grows arbitrarily large.

Recall that in the univariate modular case, the allowable bound B_x grew with the $1/d$ power of N. In the bivariate modular case we cannot hope to find a similar theorem. The achievable bounds cannot grow as N grows.

Our method will derive, from the bivariate modular equation

$$\alpha x^2 + \beta y - \tau = 0 \pmod{N},$$

a bivariate integer equation, namely

$$\alpha x^2 + \beta y - \tau = 0.$$

But it cannot enable us to solve either one.

Exactly the same example shows that *trivariate integer* equation

$$\alpha x^2 + \beta y - \tau - zN = 0,$$

is difficult to solve with bounds B_x, B_y as before, and $B_z = 2$. In the work on bivariate integer equations [3], the bounds grew with the coefficients of $p(x, y)$ (in a complicated way that depended on the degree of p), and because we have an arbitrarily large coefficient N here, again we cannot hope to achieve a similar theorem in the trivariate integer case.

But these negative results should not dissuade us. As Jutla and others have shown, many times one *can* use the multivariate versions of the present techniques. A tool that has been shown to be ineffective in one percent of the cases, can still be quite useful in the other 99 percent.

7 Acknowledgements

The author is grateful to Joe Silverman and Phong Nguyen for prodding him to actually write down the ideas that have been floating around. Conversations with Nick Howgrave-Graham are always informative and stimulating.

References

1. Dan Boneh, personal communication.
2. D. Coppersmith, Finding a small root of a univariate modular equation. *Advances in Cryptology – EUROCRYPT '96*, LNCS 1070, Springer, 1996, 155-165.
3. D. Coppersmith, Finding a small root of a bivariate integer equation; factoring with high bits known, *Advances in Cryptology – EUROCRYPT '96*, LNCS 1070, Springer, 1996, 178-189.
4. D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Crypt.* **vol 10 no 4** (Autumn 1997), 233-260.
5. D. Coppersmith, N.A. Howgrave-Graham, S.V. Nagaraj, Divisors in Residue classes—Constructively. Manuscript.
6. N. Elkies, Rational points near curves and small nonzero $|x^3 - y^2|$ via lattice reduction, *ANTS-4*, LNCS vol 1838 (2000) Springer Verlag, 33-63.
7. J. Håstad, On using RSA with low exponent in a public key network, *Advances in Cryptology – CRYPTO '85*, LNCS 218, Springer-Verlag, 1986, 403-408.

8. N.A. Howgrave-Graham, Finding small solutions of univariate modular equations revisited. *Cryptography and Coding* LNCS vol 1355. (1997) Springer-Verlag. 131-142.
9. N.A. Howgrave-Graham, personal communication, 1997.
10. N.A. Howgrave-Graham, Approximate integer gcds, This volume.
11. C.S. Jutla, On finding small solutions of modular multivariate polynomial equations, *Advances in Cryptology – EUROCRYPT '98*, LNCS 1403, Springer, 1998, 158-170.
12. S.V. Konyagin and T. Steger, On polynomial congruences, *Mathematical Notes* **Vol 55** No 6 (1994), 596-600.
13. A.K. Lenstra, H.W. Lenstra, and L. Lovasz, Factoring polynomials with rational coefficients, *Math. Ann.* **261** (1982), 515-534.
14. H. W. Lenstra, Jr., "Divisors in Residue Classes," *Mathematics of Computation*, volume 42, number 165, January 1984, pages 331-340.
15. H.W. Lenstra, personal communication.
16. K.L. Manders and L.M. Adleman, NP-Complete Decision Problems for Binary Quadratics. *JCSS* **16**(2), 1978, 168-184.
17. Phong Nguyen, personal communication.
18. T.J. Rivlin, *Chebyshev Polynomials, From Approximation Theory to Algebra and Number Theory*, Wiley (1990).