Polynomial selection for the number field sieve

Thorsten Kleinjung

EPFL IC LACAL, Station 14, CH-1015 Lausanne, Switzerland

Abstract

In this chapter the development of the polynomial selection step of the number field sieve is described, emphasising Peter Montgomery's contributions.

This article appeared as Chapter 6 of the book *Topics in Computational Number Theory inspired by Peter L. Montgomery*, edited by Joppe W. Bos and Arjen K. Lenstra and published by Cambridge University Press. See `www.cambridge.org/9781107109353`.

Contents

6
Polynomial selection for the number field sieve

Thorsten Kleinjung

6.1 The problem

Given an integer n to be factored, the very first step of the number field sieve consists in choosing two coprime polynomials $f_1, f_2 \in \mathbb{Z}[x]$, each of whose coefficients are also coprime, such that the polynomials have a common root r modulo n (equivalently, that the resultant of f_1 and f_2 is a non-zero multiple of n). In the following step, the sieving stage, one searches for sufficiently many relations, i.e., coprime pairs $(a, b) \in \mathbb{Z}^2$ such that both values $f_i(\frac{a}{b})b^{\deg f_i}$, $i = 1, 2$, are L-smooth, i.e., split into primes below L, for some parameter L. The running time of the sieving stage (and, in general, of the number field sieve) is determined by the number of pairs (a, b) one needs to inspect which in turn depends on the choice of the polynomials. Therefore it is important to carefully select a polynomial pair so that the running time of the number field sieve computation is minimised, as much as is practically possible.

In the following it is assumed that the polynomials f_1 and f_2 are both irreducible; otherwise one can split n non-trivially or one can replace the polynomials by divisors that have r as a root modulo n since the polynomial values for a pair (a, b) will be replaced by divisors and thus remain smooth.

6.2 Early methods

Since the goal of polynomial selection is to minimise the running time of the number field sieve and since it is easy to produce polynomial pairs satisfying the conditions given in the previous section, for instance, $f_1 = x + n$ and $f_2 = x$, it is important to have some means to compare (or to assess) polynomial pairs. Obviously, a more precise assessment is more costly than an imprecise one so that, if the number of produced polynomial pairs is too big, a quick imprecise assessment method is needed (or even a chain of methods with increasing accuracy and decreasing speed). The most accurate method, namely executing the remaining steps of the number field sieve, is impractical but can be altered into executing a small but representative fraction of the sieving stage and counting the number of relations. This is still very expensive and can only be used for comparing a small number of polynomial pairs. Faster assessment methods can be devised by observing that the quality of a polynomial pair seems to be mainly influenced by two properties, namely by the size of the coefficients and by the number of roots of the polynomials modulo small primes. Taking into account both properties gives a method (cf. Section 6.7 on page 11) which is faster than the sieving based assessment from above but which is not fast enough for processing a huge set of candidates. A much faster method is to focus only on the size of the coefficients by computing $||f_1||_\infty ||f_2||_\infty$, the product of the maximum norms of the polynomials. This assessment function (and its variant in Section 6.5 on page 8) is considered in the following sections until Section 6.7. Notice that it only gives a meaningful result if the degrees of the polynomials are fixed; polynomial pairs of different degree pairs must be compared by other means, e.g., the sieving based assessment method described above. In the first years after the introduction of the number field sieve the standard procedure was to consider the product of the norms followed by a sieving based assessment for the best polynomial pairs.

The first and most natural method of constructing polynomial pairs consists of setting $f_2 = x - m$ for some positive integer $m < n$, writing n in base m, i.e., $n = \sum_{i=0}^{d} a_i m^i$ with $0 \leq a_i < m$ and d minimal, as well as setting $f_1 = \sum_{i=0}^{d} a_i x^i$ so that the polynomials are coprime and have the common root m modulo n. It can be assumed that the coefficients of f_1 are coprime, otherwise their greatest common divisor is a non-trivial divisor of n. Usually one wants to find f_1 of a given degree d which can be achieved by choosing m appropriately.

In the early days of the number field sieve m was chosen to be slightly smaller than $n^{\frac{1}{d}}$ which results in a monic degree d polynomial f_1, with the monicity simplifying some parts of the later stages of the number field sieve. This gives coefficients of size about $n^{\frac{1}{d}}$ for each polynomial, so the product of the maximum norms is about $n^{\frac{2}{d}}$. Heuristically and informally, the size of the polynomial f_1 can be reduced by generating many polynomial pairs in this way and hoping to encounter one with small coefficients. More precisely, let $C > 1$ (C stands for cost) be the number of polynomial pairs one is willing to examine, i.e., for C random choices of m slightly smaller than $n^{\frac{1}{d}}$ the corresponding monic polynomial f_1 is computed and its maximum norm is examined. It is assumed that C is not too big, e.g., $C < n^{\frac{1}{2d}}$ would do, which is no restriction in practice for integers for which the number field sieve is used. One expects to find (within cost C) one value of m for which the coefficients $a_0, \dots, a_{d-1}$ of the corresponding f_1 are bounded by $C^{-\frac{1}{d}} n^{\frac{1}{d}}$, thus reducing the product of the maximum norms by a factor of $C^{\frac{1}{d}}$.

More important was the size reduction by choosing m slightly bigger than $n^{\frac{1}{d+1}}$, thus giving rise to a non-monic f_1. The sizes of the coefficients of f_1 and f_2 are about $n^{\frac{1}{d+1}}$ so that the product of the maximum norms is about $n^{\frac{2}{d+1}}$ which is smaller than $C^{-\frac{1}{d}} n^{\frac{2}{d}}$ as $C < n^{\frac{1}{2d}}$ is assumed. As above this method can be improved by repeatedly choosing random m and selecting the m that gives rise to the polynomial f_1 with the smallest maximum norm. In order to increase the chance of hitting an f_1 with small coefficients one observes that by choosing $m = \left\lfloor \sqrt[d]{\frac{n}{a_d}} \right\rfloor$ for a given leading coefficient a_d the size of the coefficient a_{d-1} in the base m expansion is about the size of a_d (factors like $2d$ are considered to be negligible since d is assumed to be small). Indeed, for $m \geq d$ one has with $\sqrt[d]{\frac{n}{a_d}} = m + \mu, 0 \leq \mu < 1$

$$\begin{aligned} 0 \leq a_{d-1} \leq \frac{n - a_d m^d}{m^{d-1}} &= a_d \frac{(m+\mu)^d - m^d}{m^{d-1}} < a_d m \left(\left(1 + \frac{1}{m} \right)^d - 1 \right) \\ &< a_d m (e^{\frac{d}{m}} - 1) < a_d (e-1) d. \end{aligned}$$

For the analysis of the expected gain let, as above, $C > 1$ be the number of polynomial pairs one is willing to examine and let $c = C^{\frac{1}{d^2-1}}$. Choosing a_d near $c^{-d} n^{\frac{1}{d+1}}$ and m as above, the values of the $d-1$ coefficients $a_0, \dots, a_{d-2}$ which are a priori of size $cn^{\frac{1}{d+1}}$, will be of size a_d with probability $c^{-(d+1)(d-1)} = C^{-1}$ so that after about C trials one expects to find a polynomial pair with $\|f_1\|_\infty = c^{-d} n^{\frac{1}{d+1}}$. Since $\|f_2\|_\infty = cn^{\frac{1}{d+1}}$, the product of the maximum norms can be decreased for $d > 1$; with an effort of C a factor of $C^{\frac{1}{d+1}}$ can be gained.

In all methods described so far a factor of about 2 can be gained by allowing signed coefficients in the base m expansion.

There are two further methods from the early days which were not really used at that time since it was not known how to exploit them, but which became important later on (cf. Section 6.8 on page 13). One consists in choosing a non-monic $f_2 = lx - m$ and adapting the base m expansion to a base $\frac{m}{l}$ expansion, thus resulting in a larger supply of linear polynomials. Since monic f_2 already provided more than enough polynomials, there was no need to consider non-monic f_2. The other method used lattice reduction to find, given $f_2 = x - m$ (or $f_2 = lx - m$), polynomials f_1 with small coefficients and the same root modulo n as f_2. In this method the size of the coefficients is about $n^{\frac{1}{d+1}}$ which is the same as above.

6.3 General remarks

Before proceeding with Montgomery's first involvement in polynomial selection, a counting argument along the lines of [4, Section 12] is described that computes what coefficient sizes one can expect for given degrees.

Let $d_i = \deg f_i$, $i = 1, 2$, be fixed, let M be an integer and let $c_i \geq 0$, $i = 1, 2$. The goal is to compute the expected number of triples (n, f_1, f_2) such that n is in the interval $[M, 2M]$, (f_1, f_2) is a valid polynomial pair for n and (the absolute values of) the coefficients of f_i are bounded by M^{c_i}. The number of polynomial pairs (f_1, f_2) satisfying the above restriction on the coefficient sizes is $\Theta(M^{(d_1+1)c_1+(d_2+1)c_2})$. If one restricts to pairs for which f_1 and f_2 each have coprime coefficients and for which $\gcd(f_1, f_2) = 1$, the number of pairs is still $\Theta(M^{(d_1+1)c_1+(d_2+1)c_2})$. This number must be at least $\Theta(M)$ in order to obtain a valid polynomial pair for each n in the interval $[M, 2M]$, which gives the condition

$$(d_1 + 1)c_1 + (d_2 + 1)c_2 \geq 1. \tag{6.1}$$

The resultant of f_1 and f_2 can be bounded by $O(M^{d_2c_1+d_1c_2})$ where the O-constant depends on d_1 and d_2. Since the resultant must be divisible by $n \geq M$, this gives a second condition

$$d_2c_1 + d_1c_2 \geq 1. \tag{6.2}$$

If this second condition is satisfied, the number of triples (n, f_1, f_2) as above is expected to be $\Theta(M^{(d_1+1)c_1+(d_2+1)c_2})$, i.e., for any n in the interval $[M, 2M]$ one expects to find on average $\Theta(M^{(d_1+1)c_1+(d_2+1)c_2-1})$ valid polynomial pairs.

Heuristically, for the pairs $(c_1, c_2) \in \mathbb{R}^2_{\geq 0}$ satisfying the two inequalities above it is possible, for an integer n, to find polynomial pairs with coefficient bounds n^{c_i}. This region is defined by the points in the first quadrant lying above the two lines given by the equality cases of the two conditions (cf. Figure 6.1 on page 6). Since smaller coefficients are assumed to be better, the interesting part consists of the pairs lying on one (or both) lines.

Notice the different nature of Conditions (6.1) and (6.2), with the former being based on an elementary counting argument and the latter being imposed by the common root requirement. The line corresponding to (6.1) marks the border at which one can expect to find polynomial pairs for every integer. Below this line only a fraction of these integers admits polynomial pairs, this is the realm of the special number field sieve which is not further considered here. The other line is, however, a hard bound and no polynomial pair can exist below it.

In the case $d_1 \neq d_2$ the two lines intersect in the point

$$P = \left(\frac{d_1 - (d_2 + 1)}{d_1(d_1 + 1) - d_2(d_2 + 1)}, \frac{(d_1 + 1) - d_2}{d_1(d_1 + 1) - d_2(d_2 + 1)} \right) \in \mathbb{R}^2_{\geq 0}.$$

For $d_1 > d_2$ the interesting part consists of the line segment between the points $(0, \frac{1}{d_2+1})$ and P, and the line segment between the points P and $(\frac{1}{d_2}, 0)$; similarly for $d_1 < d_2$ it consists of the line segment between the points $(0, \frac{1}{d_1})$ and P, and the line segment between the points P and $(\frac{1}{d_1+1}, 0)$ (note that one of the segments has length zero if d_1 and d_2 differ by 1). From the slopes of the two lines it follows that the minimum of $c_1 + c_2$ is $\frac{2}{d_1+d_2+1}$ and it is attained at P; thus the region near this point is the most interesting one. The polynomial selection method of the preceding section achieves (at cost $C = 1$) in the case $d_2 = 1$ (assuming $d_1 > 1$) coefficient sizes corresponding to the point $(\frac{1}{d_1+1}, \frac{1}{d_1+1})$ which lies on the second line but is not P. Hence it might be advantageous to try to move the coefficient sizes towards P which is exactly what was done in the preceding section by considering many polynomial pairs. However, attaining the point P results in an exponential cost[1] with the current state-of-the-art algorithms. For the case $d_1 = 5$, $d_2 = 1$ the situation is depicted in Figure 6.1.

In the case $d = d_1 = d_2$ Condition (6.2) implies Condition (6.1). Therefore the interesting part consists of the line segment between the points $(0, \frac{1}{d})$ and $(\frac{1}{d}, 0)$ on which $c_1 + c_2$ takes the value $\frac{1}{d}$ and one expects to find $\Theta(n^{\frac{1}{d}})$ polynomial pairs on average. It can be argued that $d = d_1 = d_2$ is a bad choice of degrees since the region satisfying the two inequalities for $d = d_1 = d_2$ is a proper subset of the union of the corresponding regions for degrees $(d+1, d-1)$ and for degrees $(d-1, d+1)$; for the case $d = 3$ this is illustrated in Figure 6.2 below. However, due to the absence of polynomial selection algorithms attaining the point P in acceptable time, the case $d = d_1 = d_2$ is still of interest.

6.4 A lattice based method

In 1993 Montgomery [5] addressed the case $d = d_1 = d_2$ and described a polynomial selection method using lattices. For $d = 2$, i.e., two quadratic polynomials, this method produces optimal polynomials in the sense of the preceding section, namely $c_1 + c_2 = \frac{1}{2}$.

[1] This is no longer true if n is prime, i.e., if the number field sieve is used for computing discrete logarithms in $\mathbb{F}_n$. The method in [9] addresses the case $d_1 = d_2 + 1$ and attains the point $P = (0, \frac{1}{d_1})$ by picking a random polynomial f_1 with coefficients of size $O(1)$ and computing its roots modulo n (here the primality of n is used). Then, using lattice reduction, each root allows to construct polynomials f_2 with coefficients of size $O(n^{\frac{1}{d_1}})$.

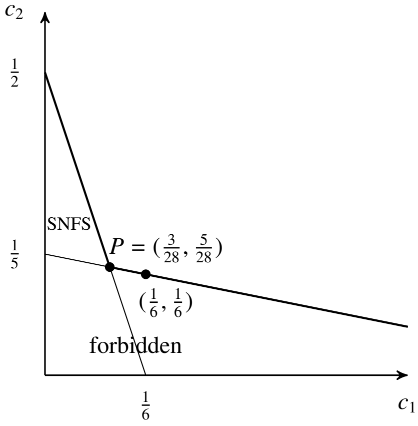

Figure 6.1 For the case $d_1 = 5$, $d_2 = 1$ the two lines corresponding to the equality cases of Conditions (6.1) and (6.2) on page 4 as well as their intersection P are depicted. The region above the bold part of the lines satisfies both inequalities. Moreover, the point $(\frac{1}{6}, \frac{1}{6})$ corresponding to the method from Section 6.2 on page 1 is featured.

By identifying polynomials in $\mathbb{Z}[x]$ of degree at most d with vectors in $\mathbb{Z}^{d+1}$ via $f = \sum_{i=0}^{d} a_i x^i \mapsto (a_0, \ldots, a_d)^T$, it is easy to see that the set of such polynomials having r modulo n as a root is a lattice $L_r \subset \mathbb{Z}^{d+1}$ of covolume n. Moreover, polynomials whose absolute values of the coefficients are bounded by $O(n^c)$ correspond to vectors of length (with respect to the Euclidian norm) $O(n^c)$. Therefore, in the case of equal degrees $d = d_1 = d_2$, the polynomial selection problem can be rephrased in terms of lattices. The task is to find a lattice L_r such that it contains two short independent vectors for which the corresponding polynomials are coprime. In general, the latter condition is satisfied but it can be violated in special situations, e.g., if there exists a polynomial f of degree smaller than d with small coefficients such that $f(r) \equiv 0 \pmod{n}$ then f and xf correspond to short vectors violating the coprimality condition.

For any $(d + 1)$-dimensional lattice of covolume n a basis $v_1, \ldots, v_{d+1}$ satisfying $\prod_{i=1}^{d+1} |v_i| = O(n)$ with the O-constant depending on d can, for instance, be found using the lattice basis reduction method from [12]. Therefore, for any r it is possible to find two vectors in L_r with product of their lengths $O(n^{\frac{2}{d+1}})$; in general one also expects that the two vectors have length $O(n^{\frac{1}{d+1}})$ (as well as all other basis vectors). Since this is optimal in the case $d = 1$ (cf. final paragraph of Section 6.3 on page 5; note that the coprimality condition is satisfied), $d \geq 2$ is assumed from now on.

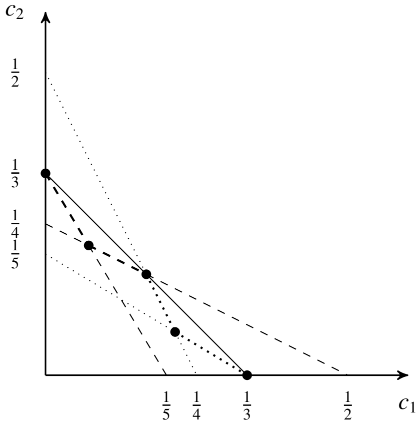

Figure 6.2 For each of the cases $d_1 = 3$, $d_2 = 3$ (normal line), $d_1 = 4$, $d_2 = 2$ (dashed lines) and $d_1 = 2$, $d_2 = 4$ (dotted lines) the two lines corresponding to the equality cases of Conditions (6.1) and (6.2) on page 4 are depicted. The bold parts of the dashed and dotted lines (connecting the highlighted intersection points) are below the line corresponding to $d_1 = d_2 = 3$.

If a lattice L_r admits a basis with two short basis vectors, i.e., the product of their lengths being much smaller than $n^{\frac{2}{d+1}}$, then the other $d - 1$ basis vectors must be longer on average. One way to achieve this is to construct L_r such that at least one of the vectors in a reduced basis is long by stipulating the existence of a non-zero linear form $\lambda : L_r \to \mathbb{Z}$ with small coefficients, i.e., a short vector in the dual lattice L_r^*. Via the standard identification of the dual of $\mathbb{Q}^{d+1}$ with $\mathbb{Q}^{d+1}$ the lattice L_r^* is generated by $\mathbb{Z}^{d+1}$ and the vector $w = \frac{1}{n}(1, r, \ldots, r^d)^T$. Then a vector of length $O(n^{-z})$ in L_r^* ensures the existence of a vector of length at least of order n^z in a reduced basis which implies the existence of two basis vectors with product of their lengths $O(n^{\frac{2(1-z)}{d}})$. For $z > \frac{1}{d+1}$ this is shorter than in the simple construction above.

In order to find a lattice L_r such that its dual lattice contains a short vector one can set $r \equiv \frac{t}{u} \pmod{n}$ with $t = O(n^{\frac{1}{d}})$, $u = \Theta(n^{\frac{1}{d}})$ so that the first d coordinates of $u^{d-1}w$ plus an appropriate vector in $\mathbb{Z}^{d+1}$ are equal to $\frac{1}{n}(t^i u^{d-1-i})$ for $i = 0, \ldots, d-1$, thus are $O(n^{-\frac{1}{d}})$. If u divides $t^d - n$ (or $t^d + hn$ for some small integer $h \neq 0$), the last coordinate is $\frac{t^d-n}{nu} = O(n^{-\frac{1}{d}})$. Thus the product of the two shortest basis vectors is $O(n^{\frac{2(d-1)}{d^2}})$, i.e., if the coprimality condition is satisfied, one gets $c_1 + c_2 \leq \frac{2(d-1)}{d^2}$ (and in general $c_1 = c_2 = \frac{d-1}{d^2}$). Finding t and u such that u divides $t^d - n$ can be done in many ways; Montgomery suggested to choose u as a prime such that n is a d-th power modulo u.

For $d = 2$ this gives the optimal bound $\frac{1}{2}$ for $c_1 + c_2$ (cf. Section 6.3 on page 4); notice that if the coprimality condition is not satisfied, one can split n. Moreover, the number of polynomial pairs of this construction is heuristically $\Theta(n^{\frac{1}{2}})$ which coincides with the expected value from the preceding section (after a slight modification of the construction, namely replacing the prime u by a product of primes for all of which n is a quadratic residue). In the early 1990s this method with the improvement described in the next section was used for several at that time large factoring projects [14],[8].

6.5 Skewness

In the sieving stage of the number field sieve a certain set of pairs $(a, b) \in [-A, A] \times [1, B] \cap \mathbb{Z}^2$ for suitable A and B is processed; this set of pairs is called the sieving region. Line sieving processes all pairs in $[-A, A] \times [1, B] \cap \mathbb{Z}^2$ by first considering the pairs with $b = 1$, then $b = 2$ and so on where each change of b involves some overhead. Additionally, if A is small compared to the size of the primes in the factor base, the procedure becomes less efficient. Therefore it is desirable to keep $E = AB$ constant while decreasing B in order to speed up sieving if this can be achieved without increasing the values of the polynomials f_1 and f_2 over the sieving region.

If a pair of quadratic polynomials is selected as described in the preceding section, they will, in general, have coefficients of size $n^{\frac{1}{4}}$, so most polynomial values for $A = B = \sqrt{E}$ will be of size $En^{\frac{1}{4}}$. Thus, by changing A and B as above most polynomial values will be of size $\frac{A}{B}En^{\frac{1}{4}}$ which is bigger by a factor of $S = \frac{A}{B}$, called skewness, due to the increased contribution of the leading terms. There is an efficient procedure to fix this problem by demanding that the leading coefficients are of size $S^{-1}n^{\frac{1}{4}}$, the middle coefficients of size $n^{\frac{1}{4}}$ and the constant coefficients of size $Sn^{\frac{1}{4}}$; such polynomials are called skewed polynomials. In the following it is supposed that $S \geq 1$ holds; the case $S \leq 1$ is symmetric.

These considerations led Montgomery to adapt his lattice based method, described in the previous section, to skewed polynomials by changing the inner product from $\langle y, z\rangle = \sum_{i=1}^{d+1} y_i z_i$ to $\langle y, z\rangle_S = \sum_{i=1}^{d+1} S^{i-1-\frac{d}{2}} y_i z_i$ where y_i resp. z_i is the i-th coordinate of y resp. z and $d = d_1 = d_2$ as in the previous section. For the dual lattice the inner product has to be changed to $\langle , \rangle_{S^{-1}}$ and in the construction of r one has to choose $u = \Theta(S^{-1}n^{\frac{1}{d}})$. Care has to be taken to not choose S too large, otherwise the vector corresponding to $ux - t$ will occur in the reduced lattice basis; for more details cf. [17] and also [15],[7]. In the case $d = 2$ any skewness $S = O(n^{\frac{1}{4}-\epsilon})$ with $0 \leq \epsilon \leq \frac{1}{4}$ works and produces polynomial pairs as asked for above. Asymptotically as well as in practice for integers n of, say, more than 40 digits E is smaller than $n^{\frac{1}{4}}$ so that one can even choose $S = E$ resulting in $B = 1$, i.e., a sieving region consisting of a single line, which almost completely removes the overhead. Montgomery noticed as well that lattice sieving, a more efficient but more complicated sieving variant, can benefit significantly from squeezing the sieving region to a single line, thus making it extremely efficient in this case. Details and reports of some computations can be found in [8] (beware, in that paper lattice sieving for $B = 1$ is called line sieving).

The reader may have noticed that the number of polynomial pairs obtained by the construction above with $d = 2$ is heuristically $\Theta(S^{-1}n^{\frac{1}{2}})$ which becomes smaller as S gets bigger. However, each polynomial pair can be expanded into $\Theta(S)$ polynomial pairs of similar coefficient sizes by translating them by an integer h of size up to $O(S)$, i.e., replacing (f_1, f_2) by $(f_1(x+h), f_2(x+h))$. These translated polynomial pairs do not provide new information since the translation corresponds to a shear mapping of the sieving region. Thus, for a fixed skewness S, one gets heuristically again $\Theta(n^{\frac{1}{2}})$ skewed polynomial pairs, although only $\Theta(S^{-1}n^{\frac{1}{2}})$ of them are essentially different.

This is a general phenomenon as will be explained by revising the discussion from Section 6.3 on page 4 in the light of skewness. For a given skewness $S = M^s$ (with the notation from that section, i.e., n being in the interval $[M, 2M]$) the S-maximum norm of a polynomial $f = \sum_{i=0}^{d} a_i x^i \in \mathbb{Z}[x]$ is defined as $\|f\|_{S,\infty} = \max(|a_i|S^{i-\frac{d}{2}})$. Since the existence of a degree d polynomial with $\|f\|_{S,\infty} \leq M^c$ for $c \in \mathbb{R}$ implies $S^{\frac{d}{2}} \leq M^c$, fixing a skewness $S = M^s$ entails the bound $\frac{sd}{2} \leq c$. As before let $d_i = \deg f_i$, $i = 1, 2$, be fixed and let $\|f_i\|_{S,\infty} \leq M^{c_i}$ for $i = 1, 2$ so that the number of such polynomial pairs (f_1, f_2) is again $\Theta(M^{(d_1+1)c_1+(d_2+1)c_2})$. On average, translation by an integer of size $O(S)$ does not change the S-maximum norms of f_1 and f_2 by much and it does not change the resultant of f_1 and f_2. Therefore the polynomial pairs are clustered into classes having the same resultant so that one needs at least $\Theta(MS)$ polynomial pairs in order to obtain a valid polynomial pair for each n in the interval $[M, 2M]$. Thus Condition (6.1) on page 4 becomes

$$(d_1 + 1)c_1 + (d_2 + 1)c_2 \geq 1 + s. \tag{6.3}$$

Condition (6.2) on page 4 is not affected by the skewness and remains the same.

In the case $d = d_1 = d_2$ Condition (6.2) implies Condition (6.3) since $c_1 + c_2 \geq 2\frac{sd}{2} \geq s$. Therefore one can expect to find $\Theta(M^{\frac{1}{d}-s})$ essentially different polynomial pairs with $c_1 + c_2 = \frac{1}{d}$ whenever $c_i \geq \frac{sd}{2}$, $i = 1, 2$. For $d_1 = d_2 = 2$ this is exactly what was observed above.

In the general case $d_1 > d_2$ the two lines given by the equality cases of Conditions (6.3) and (6.2) intersect in the point

$$P_s = \left(\frac{d_1(1+s) - (d_2+1)}{d_1(d_1+1) - d_2(d_2+1)}, \frac{(d_1+1) - d_2(1+s)}{d_1(d_1+1) - d_2(d_2+1)} \right).$$

For $s = 0$, i.e., skewness $S = 1$, this is the point P from Section 6.3 on page 4 and for increasing s it moves on the line associated to Condition (6.2) to the right. For $s > s_1 = \frac{2d_1 - 2d_2 - 2}{d_1(d_1(d_1+1) - d_2(d_2+1) - 2)}$ the first coordinate of P_s violates $c_1 \geq \frac{sd_1}{2}$ and for larger s it also violates $c_2 \geq \frac{sd_2}{2}$. Thus for $s_1 < s \leq s_2 = \frac{1}{d_1 d_2}$ the optimal point (with respect to minimising $c_1 + c_2$) is $(\frac{sd_1}{2}, \frac{1}{d_1} - \frac{sd_2}{2})$ and for $s > s_2$ it is $(\frac{sd_1}{2}, \frac{sd_2}{2})$.

When considering these optimal points, the value of $c_1 + c_2$ is minimal for $s = 0$, increases slightly for $0 < s \leq s_1$ and more rapidly for $s > s_1$. Similarly the expected number of polynomial pairs is $\Theta(1)$ for $s = 0$ and grows with s. These considerations suggest that searching for skewed polynomial pairs is a bad idea. However, as remarked above, there are no known methods for attaining the optimal point P (resp. P_s) so that using skewness might be a good idea, and, as shown in the next section, it turns out that using skewness is indeed very useful. Moreover, assessing the quality of the polynomial pair via the value $c_1 + c_2$ is adequate for asymptotic considerations but is a very rough assessment in practice where one wants to distinguish between c_1 and c_2 as well as include the number of roots modulo small primes. The impact of using skewness on the latter is discussed in Section 6.7.

6.6 Base m method and skewness

Once the concept of skewness has been introduced it is relatively easy to include it in the base m method. In the following the notation from Section 6.2 on page 1 is used, in particular $d = d_1$, $d_2 = 1$, and $f_2 = x - m$. For simplifying the presentation it is also assumed that $d > 3$; the case $d = 3$ can be handled with some modifications although it is probably irrelevant in practice. As before, many polynomial pairs are generated by choosing a_d smaller than $n^{\frac{1}{d+1}}$, picking m near $\sqrt[d]{\frac{n}{a_d}}$ so that the coefficient a_{d-1} is of size a_d, and then checking whether the remaining coefficients are small enough with respect to some previously fixed skewness S. More precisely, denote the cost, i.e., the number of polynomial pairs to be inspected, by C (again assuming that C is not too big), set $S = C^{\frac{2}{(d-3)(d-2)}}$, and choose a_d near $(nS^{(1-d)d})^{\frac{1}{d+1}}$. This choice follows from stipulating that a_{d-i}, $i = 2, \ldots, d-1$ is bounded by $a_d S^i$, that the bound $a_d S^{d-1}$

for a_1 is of size $m \approx \sqrt[d]{\frac{n}{a_d}}$ and that these bounds are satisfied with probability C^{-1}. Thus, after having checked about C coefficients a_d, one can expect to find one polynomial pair satisfying

$$||f_1||_{S,\infty} \le C^{-\frac{d}{(d-2)(d+1)}} n^{\frac{1}{d+1}} \quad \text{and} \quad ||f_2||_{S,\infty} \le C^{\frac{1}{(d-2)(d+1)}} n^{\frac{1}{d+1}},$$

i.e., $||f_1||_{S,\infty}||f_2||_{S,\infty} \le C^{-\frac{d-1}{(d-2)(d+1)}} n^{\frac{2}{d+1}}$. In other words, with an effort of C one can gain a factor of $C^{\frac{d-1}{(d-2)(d+1)}}$ which is better than the factor of $C^{\frac{1}{d+1}}$ in the unskewed base m method. Notice that the basic operations involved in checking a polynomial pair are the same as in the unskewed base m method so this speedup carries over to practice.

6.7 Root sieve

Using skewed polynomials provides another advantage, namely that it is possible to increase the number of roots of the polynomials modulo small primes. Montgomery contributed substantially to the development of the algorithm described below (and presented in more detail in [16], cf. also [6]).

With the parameter choice as in the previous section the bound for $||f_1||_{S,\infty}$ implies that the coefficient a_1 can be of size m and that the coefficient a_0 can be of size Sm. Therefore one can expect to find about $\Theta(S)$ integers i such that $||f_1 + if_2||_{S,\infty} \le C^{-\frac{d}{(d-2)(d+1)}} n^{\frac{1}{d+1}}$. This provides $\Theta(S)$ polynomial pairs $(f_1 + if_2, f_2)$ which each have approximately the same S-maximum norm as the original pair (f_1, f_2). Notice that this is a different type of amplifying polynomial pairs than the translations $x \mapsto x + h$ discussed in Section 6.5 on page 8. The main difference is that translating conserves the number of roots modulo a prime, whereas adding a multiple of f_2 often changes this number. Thus, by considering the polynomials $f_1 + if_2$ for integers $i = O(S)$, the quality of a polynomial pair can be improved at cost $O(S)$ which is $O(C)$ for $d = 4$ and $o(C)$ for $d > 4$. Before describing a procedure for inspecting this set of polynomials it is necessary to know how the number of roots modulo small primes influences the quality of a polynomial pair.

In order to simplify the discussion assume that sieving is done with only one polynomial f of degree d and that the polynomial values $f(\frac{a}{b})b^d$ over the sieving region do not depend much on the polynomials to be assessed, e.g., all polynomials to be assessed have the same degree d and the sizes of their coefficients do not vary by much. Let p be a (small) prime that neither divides the leading coefficient nor the discriminant of f. This condition implies that the number of roots of f modulo p^k does not depend on $k \ge 1$; denote this number by $n_p(f)$. Therefore, exactly $(p-1)p^{k-1}n_p(f)$ of the $(p^2-1)p^{2k-2}$ polynomial values $f(\frac{a}{b})b^d$ where $0 < a, b < p^k$, $p \nmid \gcd(a,b)$, are divisible by p^k, whereas for the same number of random values one expects $(p^2-1)p^{k-2}$ of them to be divisible by p^k. Summing over all $k \geq 1$ one expects that a polynomial value in the sieving region is divisible by $p^{\frac{pn_p(f)}{p^2-1}}$ on average (geometric mean) and that a random value is divisible by $p^{\frac{1}{p-1}}$ on average. This suggests to define $\alpha_p(f) = (\frac{1}{p-1} - \frac{pn_p(f)}{p^2-1})\log p$. For primes dividing the leading coefficient or the discriminant of f a similar (slightly more complicated) definition can be derived, and one sets

$$\alpha(f) = \sum_{p\,<\,P,\ p \text{ prime}} \alpha_p(f) \tag{6.4}$$

where P is a small bound, e.g., $P = 1000$. Notice that the sum $\sum_{p \text{ prime}} \alpha_p(f)$ converges [3] and that the contribution of $p \geq P$ can be neglected in practice.

The interpretation of $\alpha(f)$ is that on average the P-smooth part of a polynomial value in the sieving region is $e^{-\alpha(f)}$ times the P-smooth part of a random value of similar size. This suggests to use $||f||_{S,\infty}e^{\alpha(f)}$ for measuring the quality of a polynomial f. Notice that $\alpha(f)$ is constant and positive for linear polynomials f so that the corresponding values $f(\frac{a}{b})b^d$ behave slightly worse than random integers with respect to the number of roots modulo small primes. Indeed, one has $n_p(f) = 1$ so that $(p-1)p^{k-1}$ of the $(p^2-1)p^{2k-2}$ polynomial values considered above are divisible by p^k which is less than the $(p^2-1)p^{k-2}$ values for random integers. Since f_2 is linear in the base m method, the function $||f_1||_{S,\infty}||f_2||_{S,\infty}e^{\alpha(f_1)}$ can be used for assessing polynomial pairs produced by this method. More functions for determining the quality of polynomial pairs can be found in [16].

Returning to the set of polynomial pairs $(f_1 + if_2, f_2)$ one notices that $||f_2||_{S,\infty}$ does not depend on i and that, by construction, $||f_1 + if_2||_{S,\infty}$ does usually not depend on i so that it is sufficient to compute $\alpha(f_1 + if_2)$ in order to find the best polynomial pairs in the set. Notice that computing $\alpha(f)$ for a polynomial f involves finding the roots of f modulo many small primes p so that the computation of $\alpha(f)$ is much slower than, say, the computation of $||f||_{S,\infty}$. Therefore it is desirable to speed up this computation, which can be done as follows by using a sieving procedure, called root sieve, that computes an approximation of $\alpha(f_1 + if_2)$ for i in an interval. For a prime $p < P$ and an s with $0 \leq s < p$ it is easy to determine all i such that $f_1 + if_2$ has s as a root modulo p; prime powers p^k can be handled similarly. The cases where a prime divides the discriminant of $f_1 + if_2$ for an i in the interval require a slightly bigger effort but are rare. Suppressing these cases and prime powers for this description, the sieving procedure consists of initialising an array indexed by i with $\sum_{p\,<\,P,\ p \text{ prime}} \frac{\log p}{p-1}$ and subtracting for each pair (p, s) with p a prime, $p < P$, $0 \leq s < p$, (an approximation of) the value $\frac{\log p}{p+1}$ from all positions i for which s is a root of $f_1 + if_2$ modulo p. This gives approximations of $\alpha(f_1 + if_2)$ and one can further inspect the best ones, i.e., the smallest values.

In general, it is possible to sieve over polynomials $f_1 + gf_2$ for arbitrary $g \in \mathbb{Z}[x]$ which for $\deg g = 0$ is described above. However, for g of bigger degree the S-maximum norm of $f_1 + gf_2$ is usually much bigger. This can be countered by changing the skewness or by translations but so far, i.e., for n up to 768 bit, it is sufficient to consider g of degree 1. In this case the sieving is carried out for $f_1 + (i_1x + i_0)f_2$ which can be done as above with the obvious modifications; notice that the bounds on i_1 are much smaller than those on i_0 so that a simple loop over i_1 and proceeding per i_1-value as above is almost sufficient.

6.8 Later developments

In this final section a few later improvements are described. The skewed base m method is a two stage algorithm: first a brute force search produces a few polynomial pairs with a good product of the S-maximum norms, then for each such pair a root sieve is applied to improve the α-value while almost conserving the norms. Since a better α-value can be expected for a larger search area, it is tempting to increase the search area, thereby increasing the norms, and hoping that this is compensated for by an improved α-value. If this approach is adopted, the time spent in the root sieve will increase and may eventually dominate the polynomial selection running time so that improvements for speeding up the root sieve are needed. One improvement is to inspect only the most promising candidates in the search area, namely those for which the first few terms in the sum (6.4) are already good, i.e., small. This can result in a significant speedup at the cost of missing a few good candidates, and is described in [2]. The paper also explains how to reduce to almost zero the time spent in dealing with prime powers.

Returning to the first stage of the polynomial selection, the two methods mentioned in the final paragraph of Section 6.2 (on page 3) eventually proved to be useful. The first method, namely considering non-monic f_2, was used in [11] (and earlier in the less efficient algorithm [10]) to produce polynomial pairs with a small third coefficent a_{d-2}. Since a_d and a_{d-1} are small by construction, a_{d-2} has the biggest impact on $||f_1||_{S,\infty}$ and it is therefore important to reduce its size. With the notation from Section 6.6 on page 10 one has to check about $C^{\frac{2}{d-2}}$ values of a_d in order to find an expansion for which the bound on

a_{d-2} is satisfied. The new method produces with an effort $C^{\frac{1}{d-2}}$ (suppressing logarithmic terms) a polynomial pair satisfying the bound on a_{d-2}. Thus, rebalancing all parameters, one expects, by spending an effort of C, to gain a factor of $C^{\frac{d-1}{(d-3)(d+1)}}$. After the effort for producing polynomials with small third coefficient a_{d-2} has been reduced, the main obstacle for $d > 5$ is the fourth coefficient a_{d-3}. This is considered in [1] where the second method from the final paragraph of Section 6.2 is combined with the method just described and a careful choice of a translation.

These new developments do not introduce fundamentally new concepts but build on the methods described before, many of which having been invented by Peter Montgomery.

Bibliography

[1] S. Bai, C. Bouvier, A. Kruppa, and P. Zimmermann. Better polynomials for GNFS. *Mathematics of Computation*, pages 1–12, December 2015. (Cited on page 14.)

[2] S. Bai, R. P. Brent, and E. Thomé. Root optimization of polynomials in the number field sieve. *Mathematics of Computation*, 84(295), 2015. (Cited on page 13.)

[3] R. Barbulescu and A. Lachand. Some mathematical remarks on the polynomial selection in NFS. *Mathematics of Computation*, 86(303):397–418, 2017. (Cited on page 12.)

[4] J. P. Buhler, H. W. Lenstra Jr., and C. Pomerance. Factoring integers with the number field sieve. pages 50–94 in [13], 1992. (Cited on page 4.)

[5] J. P. Buhler, P. Montgomery, R. Robson, and R. Ruby. Technical report implementing the number field sieve. *Oregon State University, Corvallis, OR*, 1994. (Cited on page 5.)

[6] S. Cavallar, B. Dodson, A. K. Lenstra, P. C. Leyland, W. M. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, and P. Zimmermann. Factorization of RSA-140 using the number field sieve. In K.-Y. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology – ASIACRYPT'99*, volume 1716 of *Lecture Notes in Computer Science*, pages 195–207. Springer, Heidelberg, Nov. 1999. (Cited on page 11.)

[7] N. Coxon. Montgomery's method of polynomial selection for the number field sieve. *Linear Algebra and its Applications*, 485:72–102, 2015. (Cited on page 8.)

[8] M. Elkenbracht-Huizing. An implementation of the number field sieve. *Experimental Mathematics*, 5(3):231–253, 1996. (Cited on pages 8 and 9.)

[9] A. Joux and R. Lercier. Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the gaussian integer method. *Mathematics of Computation*, 72(242):953–967, 2003. (Cited on page 5.)

[10] T. Kleinjung. On polynomial selection for the general number field sieve. *Mathematics of Computation*, 75(256):2037–2047, 2006. (Cited on page 13.)

[11] T. Kleinjung. Polynomial selection, presented at the CADO workshop. See `http://cado.gforge.inria.fr/workshop/slides/kleinjung.pdf`, 2008. (Cited on page 13.)

[12] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. (Cited on page 6.)

[13] A. K. Lenstra and H. W. Lenstra Jr. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993. (Cited on page 15.)

[14] P. L. Montgomery. A survey of modern integer factorization algorithms. *CWI Quarterly*, 7(4):337–366, December 1994. (Cited on page 8.)

[15] P. L. Montgomery. Searching for higher-degree polynomials for the general number field sieve. `helper.ipam.ucla.edu/publications/scws1/scws1_6223.ppt`, October 2006. (Cited on page 8.)

[16] B. A. Murphy. *Polynomial selection for the number field sieve integer factorisation algorithm*. PhD thesis, Australian National University, 1999. (Cited on pages 11 and 12.)

[17] T. Prest and P. Zimmermann. Non-linear polynomial selection for the number field sieve. *J. Symb. Comput.*, 47(4):401–409, 2012. (Cited on page 8.)

Subject index