# Tests and Constructions of Irreducible Polynomials over Finite Fields

Shuhong Gao[1] and Daniel Panario[2]

[1] Department of Mathematical Sciences, Clemson University, Clemson, South Carolina 29634-1907, USA
E-mail: sgao@math.clemson.edu
[2] Department of Computer Science, University of Toronto, Toronto, Canada M5S-1A4
E-mail: daniel@cs.toronto.edu

**Abstract.** In this paper we focus on tests and constructions of irreducible polynomials over finite fields. We revisit Rabin's (1980) algorithm providing a variant of it that improves Rabin's cost estimate by a $\log n$ factor. We give a precise analysis of the probability that a random polynomial of degree $n$ contains no irreducible factors of degree less than $O(\log n)$. This probability is naturally related to Ben-Or's (1981) algorithm for testing irreducibility of polynomials over finite fields. We also compute the probability of a polynomial being irreducible when it has no irreducible factors of low degree. This probability is useful in the analysis of various algorithms for factoring polynomials over finite fields. We present an experimental comparison of these irreducibility methods when testing random polynomials.

## 1 Motivation and results

For a prime power $q$ and an integer $n \geq 2$, let $\mathbb{F}_q$ be a finite field with $q$ elements, and $\mathbb{F}_{q^n}$ be its extension of degree $n$. Extensions of finite fields are important in implementing cryptosystems and error correcting codes. One way of constructing extensions of finite fields is via an irreducible polynomial over the ground field with degree equal to the degree of the extension. Therefore, finding irreducible polynomials and testing the irreducibility of polynomials are fundamental problems in finite fields.

A probabilistic algorithm for finding irreducible polynomials that works well in practice is presented in [26]. The central idea is to take polynomials at random and test them for irreducibility. Let $I_n$ be the number of irreducible polynomials of degree $n$ over a finite field $\mathbb{F}_q$. It is well-known (see [21], p. 142, Ex. 3.26 & 3.27) that

$$\frac{q^n}{n} - \frac{q(q^{n/2}-1)}{(q-1)n} \leq I_n \leq \frac{q^n - q}{n}. \tag{1}$$

This means that a fraction $1/n$ of the polynomials of degree $n$ are irreducible, and so we find on average one irreducible polynomial of degree $n$ after $n$ tries. In order to transform this idea into an algorithm one has to consider irreducibility tests.

In sections 2 and 3, we focus on tests for irreducibility. Let $f \in \mathbb{F}_q[x]$, $\deg f = n$, be a polynomial to be tested for irreducibility. Assume that $p_1, \ldots, p_k$ are the distinct prime divisors of $n$. In practice, there are two general approaches for this problem:

- Butler (1954): $f$ is irreducible if and only if $\dim\ker(\Phi - I) = 1$, where $\Phi$ is the Frobenius map on $\mathbb{F}_q[x]/(f)$ that sends $h \in \mathbb{F}_q[x]/(f)$ to $h^q \in \mathbb{F}_q[x]/(f)$, and $I$ is the identity map on $\mathbb{F}_q[x]/(f)$ (see [4]);
- Rabin (1980): $f$ is irreducible if and only if $\gcd(f, x^{q^{n/p_i}} - x) = 1$ for all $1 \le i \le k$, and $x^{q^n} - x \equiv 0 \bmod f$ (see [26]).

Other irreducibility tests can be found in [14], [31], and [12].

In this paper, we concentrate on Rabin's test, and a variant presented in [1]. In section 2, we review Rabin's and Ben-Or's irreducibility algorithms. We state a variant of Rabin's algorithm that allows a $\log n$ factor saving. In section 3, we focus on Ben-Or's algorithm. This leads us to study the behavior of *rough* polynomials, i.e., polynomials without irreducible factors of low degrees. The analysis is expressed as an asymptotic form in $n$, the degree of the polynomial to be tested for irreducibility. First, we fix a finite field $\mathbb{F}_q$, and then we study asymptotics on $q$. As was noted in [7], probabilistic properties of polynomials over finite fields frequently have a shape that resembles corresponding properties of the cycle decomposition of permutations to which they reduce when the size of the field goes to infinity. An instance of this is derived for the probability that a polynomial of degree $n$ over $\mathbb{F}_q$ contains no factors of degree $m$, $1 \le m \le O(\log n)$, when $q \to +\infty$. This probability relates naturally with Ben-Or's algorithm. The probability of a polynomial being irreducible when it has no irreducible factors of low degree provides useful information for factoring polynomials over finite fields (see for instance [12], §6). We provide the probability of a polynomial being irreducible when it has no irreducible factors of degree at most $O(\log n)$.

In section 4, we give an experimental comparison on the algorithms discussed in section 2. We provide tables of running times of the algorithms for various fields and polynomial degrees. These results suggest that Ben-Or's algorithm has a much better average time behavior than others, even though its worst-case complexity is the worst.

Very sparse irreducible polynomials are useful for several applications: pseudorandom number generators using feedback shift registers ([15]), discrete logarithm over $\mathbb{F}_{2^n}$ ([6], [23]), and efficient arithmetic in finite fields (Shoup private communication, 1994). However, few results are known about these polynomials beyond binomials and trinomials (see [22], Chapter 3, and the references there). In section 5, we present a construction of irreducible polynomials over $\mathbb{F}_q$ of degree $n$ with up to $O(1)$ nonzero terms (not necessarily the lowest coefficients), for infinitely many degrees $n$.

We assume that arithmetic in $\mathbb{F}_q$ is given. The cost measure of an algorithm will be the number of operations in $\mathbb{F}_q$. The algorithms in this paper use basic polynomial operations like products and gcds. We consider in this paper exclusively FFT based arithmetic; similar results hold for classical arithmetic. Let

$M(n) = n \log n \log\log n$. For constants $\tau_1$ and $\tau_2$, the cost of multiplying two polynomials of degree at most $n$ using "fast" arithmetic ([28], [27], [5]) can be taken as $\tau_1 M(n)$, and the cost of a gcd between two polynomials of degree at most $n$ can be taken as $\tau_2 \log n M(n)$ operations in $\mathbb{F}_q$. The number of multiplications needed to compute $h^q \bmod f$ by means of the classical *repeated squaring* method (see [20], p. 441–442), where $h$ is a polynomial over $\mathbb{F}_q$ of degree less than $n$, is $C_q = \lfloor \log_2 q \rfloor + \nu(q)$, with $\nu(q)$ the number of ones in the binary representation of $q$. Therefore, the cost of computing $h^q \bmod f$ by this method is $\tau_1 C_q M(n)$ operations in $\mathbb{F}_q$ using FFT based methods.

## 2 Irreducibility tests

In this section, we review Rabin's and Ben-Or's tests, and we present a variant of Rabin's method.

### 2.1 Rabin irreducibility test and an improvement

**Algorithm:** Rabin Irreducibility Test
**Input:** A monic polynomial $f \in \mathbb{F}_q[x]$ of degree $n$,
and $p_1, \ldots, p_k$ all the distinct prime divisors of $n$ .
**Output:** Either "$f$ is irreducible" or "$f$ is reducible".

```
for j := 1 to $k$ do
    $n_j$ := $n/p_j$;
for i := 1 to $k$ do
    $g$ := gcd($f$, $x^{q^{n_i}} - x$ mod $f$);
    if $g \neq 1$, then '$f$ is reducible' and STOP;
endfor;
$g$ := $x^{q^n} - x$ mod $f$;
if $g = 0$, then '$f$ is irreducible'
        else '$f$ is reducible';
```

The correctness of Rabin's algorithm is based on the following fact (see [26], p. 275, Lemma 1).

**Fact 2.1** *Let $p_1, \ldots, p_k$ be all the prime divisors of $n$, and denote $n/p_i = n_i$, for $1 \leq i \leq k$. A polynomial $f \in \mathbb{F}_q[x]$ of degree $n$ is irreducible in $\mathbb{F}_q[x]$ if and only if gcd $(f, x^{q^{n_i}} - x \bmod f) = 1$, for $1 \leq i \leq k$, and $f$ divides $x^{q^n} - x$.*

The basic idea of this algorithm is to compute $x^{q^{n_i}} \bmod f$ independently for each value $n_1, \ldots, n_k$ by repeated squaring, and then to take the correspondent gcd. The worst-case analysis given in [26] is $O\left(nM(n) \log n \log q\right)$ operations in $\mathbb{F}_q$. However, it can be shown that $O\left(nM(n) \log\log n \log q\right)$ is an upper bound of the number of operations in $\mathbb{F}_q$ for this algorithm. Indeed, first note that the number of distinct prime factors of $n$ is at most $\log n$. The cost of $k$ exponentiations is

$$\sum_{i=1}^{k} \frac{n}{p_i} \log q\, M(n) \leq nM(n) \log q \sum_{i=1}^{\log n} \frac{1}{p_i} \leq nM(n) \log q\, H_{\log n},$$

where $H_m = \sum_{k=1}^{m} 1/k$ is the harmonic sum. Using the well-known approximation of the harmonic sum ([16], p. 452), $H_{\log n} = \log\log n + \gamma + O(\frac{1}{\log n})$, we obtain $O(nM(n) \log\log n \log q)$, which dominates the cost $O(M(n) \log^2 n)$ of computing $k$ gcd's. Therefore, the total cost of Rabin's algorithm is $O(nM(n) \log\log n \log q)$.

As an improvement, we propose the following variant for the computation of $x^{q^{n_i}} - x \bmod f$, for $1 \leq i \leq k$.

---

**Algorithm:** Variant of Rabin Irreducibility Test
**Input:** A monic polynomial $f \in \mathbb{F}_q[x]$ of degree $n$,
and $p_1, \ldots, p_k$ all the distinct prime divisors of $n$ .
**Output:** Either "$f$ is irreducible" or "$f$ is reducible".

```
$n_0$ := 0;   $h_0$ := $x$;
for j := 1 to $k$ do
    $n_j$ := $n/p_j$;
sort($n_1, \ldots, n_k$);   (* Assume $n_1 < n_2 < \ldots < n_k$. *)
for i := 1 to $k$ do
    $h_i$ := $h_{i-1}^{q^{n_i - n_{i-1}}}$ mod $f$;
    $g$ := gcd($f$, $h_i - x$);
    if $g \neq 1$, then '$f$ is reducible' and STOP;
endfor;
$g$ := $h_k^{q^{n-n_k}} - x$ mod $f$;
if $g = 0$, then '$f$ is irreducible'
         else '$f$ is reducible';
```

---

**Theorem 2.2** *The above variant of Rabin's algorithm correctly tests for polynomial irreducibility, and uses $O(nM(n) \log q)$ operations in $\mathbb{F}_q$.*

*Proof.* The correctness of the algorithm follows from the correctness of the power computations. We prove that $h_i = x^{q^{n_i}} \bmod f$, $1 \leq i \leq k$, by induction on $k$. Basis: when $k = 1$,

$$h_1 \equiv h_0^{q^{n_1 - n_0}} \equiv \left(x^{q^{n_0}}\right)^{q^{n_1 - n_0}} \equiv x^{q^{n_0} \cdot q^{n_1}} \equiv x^{q^{n_1}} \bmod f.$$

Inductive step: for some $k$, $h_i = x^{q^{n_i}} \bmod f$, $1 \leq i \leq k$. Then,

$$h_{k+1} \equiv h_k^{q^{n_{k+1} - n_k}} \equiv \left(x^{q^{n_k}}\right)^{q^{n_{k+1} - n_k}} \equiv x^{q^{n_k} \cdot q^{n_{k+1} - n_k}} \equiv x^{q^{n_{k+1}}} \bmod f.$$

With this variant, in the worst-case, the number of polynomial multiplications in Rabin's algorithm to compute all powers using repeated squaring is

$$n_1 \log q + (n_2 - n_1)\log q + \cdots + (n - n_k)\log q = n \log q.$$

Hence the cost of $k$ exponentiations is $O(nM(n)\log q)$. Since the number of distinct prime factors of $n$ is at most $\log n$, the cost of taking all the gcd's in the algorithm is $O(M(n)\log^2 n)$. Therefore the total cost of this variant is $O(nM(n)\log q)$. $\square$

### 2.2 Ben-Or irreducibility test

**Algorithm:** Ben-Or Irreducibility Test
**Input:** A monic polynomial $f \in \mathbb{F}_q[x]$ of degree $n$.
**Output:** Either "$f$ is irreducible" or "$f$ is reducible".

```
for i := 1 to $n/2$ do
    begin
       $g$ := gcd($f$, $x^{q^i} - x$ mod $f$);
       if $g \neq 1$, then '$f$ is reducible' and STOP;
    end;
'$f$ is irreducible';
```

The correctness of Ben-Or's procedure is based on the following fact (see [21], p. 91, Theorem 3.20).

**Fact 2.3** *For $i \geq 1$, the polynomial $x^{q^i} - x \in \mathbb{F}_q[x]$ is the product of all monic irreducible polynomials in $\mathbb{F}_q[x]$ whose degree divides $i$.*

Indeed, Ben-Or's algorithm computes $x^{q^i}$ mod $f$, and $\gcd(f, x^{q^i} - x)$ for $i = 1, \ldots, \frac{n}{2}$. The polynomial is reducible if and only if one of the gcd's is different from 1.

In the worst case, this algorithm computes $\frac{n}{2}$ times a $q$th power and a gcd of polynomials of degree at most $n$. Recalling the cost of these operations from section 1, the worst-case behavior of Ben-Or's algorithm is $O(nM(n)\log(qn))$ using FFT based multiplication algorithms, and therefore, it is worse than our Rabin's variant. However, as can be seen from our theoretical and experimental results in Sections 3 and 4, Ben-Or's algorithm is very efficient. The main reason for the efficiency of this algorithm is that random polynomials of large degree are very likely to have an irreducible factor of small degree, and Ben-Or's algorithm quickly discards these polynomials (see also [19]).

We should point out that the average cases of Rabin and our variant are not known. Ben-Or's average-case analysis is only known when $q$ goes to infinity in the following sense. Let $s(f)$ be the expected value of the smallest degree among the irreducible factors of $f$; then the expected cost of Ben-Or's algorithm is $O(s(f)M(n)\log(qn))$. Ben-Or ([1], Theorem 2) derives an $O(\log n)$ estimate for $s(f)$. In fact, he relates the factorial decomposition of polynomials with the cyclic decomposition of permutations. The result follows from the study of the expected length of the shortest cycle in a random permutation ([30]). However, this relation between irreducible factors of polynomials and cycles of permutations just holds when the size of the field is large, as it was observed in [7].

## 3 Distribution of rough polynomials

Polynomials without irreducible factors of low degree make Ben-Or's irreducibility test to execute a large number of iterations. The probability that a random polynomial of degree $n$ contains no factors of low degree gives meaningful information on the behavior of Ben-Or's algorithm. We call a polynomial $m$*-rough* if it has no irreducible factors of degrees $\leq m$. In this section we are interested in the distribution of rough polynomials.

The following theorem is proved in [2] when $m$ is fixed.

**Theorem 3.1** *Denote by $P_q(n,m)$ the probability of a random monic polynomial of degree $n$ over $\mathbb{F}_q$ being $m$-rough. Then when $n \to \infty$,*

$$P_q(n,m) = \prod_{k=1}^{m} \left(1 - \frac{1}{q^k}\right)^{I_k} (1 + o(1)),$$

*uniformly for $q$ and $1 \leq m \leq O(\log n)$.*

*Proof.* Let $\mathcal{I}$ be the collection of all monic irreducible polynomials in $\mathbb{F}_q$. Formally, the summation of all monic polynomials with all irreducible factors with degree $> m$ is

$$P = \prod_{\omega \in \mathcal{I},\ |\omega| > m} (1 + \omega + \omega^2 + \cdots) = \prod_{\omega \in \mathcal{I},\ |\omega| > m} (1 - \omega)^{-1}.$$

Let $z$ be a formal variable, and $|\omega|$ the degree of $\omega \in \mathcal{I}$. The substitution $\omega \mapsto z^{|\omega|}$ produces the generating function $P_m(z)$ of polynomials with all irreducible factors having degrees $> m$

$$P_m(z) = \prod_{\omega \in \mathcal{I},\ |\omega| > m} \left(1 - z^{|\omega|}\right)^{-1} = \prod_{k > m} \left(1 - z^k\right)^{-I_k} = \frac{1}{1 - qz} \prod_{k=1}^{m} (1 - z^k)^{I_k}.$$

Note that $m$ may vary when $n \to +\infty$, and thus we can not apply the transfer lemmas in [8, 24].

As usual, $[z^n]P_m(z)$ represents the coefficient of $z^n$ in $P_m(z)$, and observe that $P_q(n,m) = [z^n]P_m(z)/q^n$. In order to estimate $P_q(n,m)$, we apply Theorem 10.8 in [24]. $P_m(z)$ presents a pole of order 1 at $z = \frac{1}{q}$ with residue

$$-\frac{1}{q} \prod_{k=1}^{m} (1 - q^{-k})^{I_k}.$$

Denote by $g_q(m)$ the product $\prod_{k=1}^{m}(1-q^{-k})^{I_k}$. Suppose that $m \le c \log n$ for some constant $c > 0$. Let $b$ be a constant such that $1 < b < e^{1/c}$, and take $r = \frac{b}{q} > \frac{1}{q}$. By Odlyzko ([24], Theorem 10.8),

$$\left|[z^n]P_m(z) + \left(-\frac{1}{q} g_q(m) \left(\frac{1}{q}\right)^{-n-1}\right)\right| \le w\, r^{-n} + \left(r - \frac{1}{q}\right)^{-1} r^{-n}\, \frac{1}{q}\, g_q(m),$$

where $w = \max_{|z|=r} |P_m(z)|$. Therefore,

$$\begin{aligned}|P_q(n,m) - g_q(m)| &\le \left(w + \left(r - \frac{1}{q}\right)^{-1} \frac{1}{q}\, g_q(m)\right) (rq)^{-n} \\ &= \left(w + \frac{1}{b-1}\, g_q(m)\right) / b^n \le \left(w + \frac{1}{b-1}\right) / b^n,\end{aligned}$$

as $0 \le g_q(m) \le 1$. Since $b > 1$ is a constant independent of $m, q$ and $n$, we only need to estimate $w$ in term of $n$. When $|z| = r = \frac{b}{q}$, $|1 - qz| \ge b - 1$, and $|1 - z^k| \le 1 + r^k$. Considering that $I_k\, r^k \le q^k\, r^k = b^k$, we obtain

$$\begin{aligned}|P_m(z)| &= \frac{1}{|1-qz|} \prod_{k=1}^{m} |1 - z^k|^{I_k} \le \frac{1}{b-1} \prod_{k=1}^{m} (1 + r^k)^{I_k} \\ &= \frac{1}{b-1} \prod_{k=1}^{m} \exp\left(I_k \log(1 + r^k)\right) \le \frac{1}{b-1} \prod_{k=1}^{m} \exp(I_k\, r^k) \\ &\le \frac{1}{b-1} \prod_{k=1}^{m} \exp(b^k) = \frac{1}{b-1} \exp\left(\sum_{k=1}^{m} b^k\right) \le \frac{1}{b-1} \exp\left(\frac{b^{m+1}}{b-1}\right) \\ &\le \frac{1}{b-1} \exp\left(\frac{b}{b-1} b^{c \log n}\right) = \frac{1}{b-1} \exp\left(\frac{b}{b-1} n^{c \log b}\right).\end{aligned}$$

Hence, $w \le \frac{1}{b-1} \exp\left(\frac{b}{b-1} n^{c \log b}\right)$, and

$$\left(w + \frac{1}{b-1}\right) / b^n \le \frac{2}{b-1} \exp\left(\frac{b}{b-1} n^{c \log b} - (\log b)\, n\right).$$

By Theorem 3.2 below,

$$g_q(m) \ge \frac{1}{em} \ge \frac{1}{ec \log n}.$$

Therefore,

$$\left|\frac{P_q(n,m)}{g_q(m)} - 1\right| \le \frac{2}{b-1} ec \log n \exp\left(\frac{b}{b-1} n^{c \log b} - (\log b)\, n\right). \tag{2}$$

As $\log b > 0$ and $c \log b < 1$, the right-hand of (2) approaches to 0 as $n \to \infty$. Since the quantity on the right-hand of (2) is independent of $q$ and $m$, we see that $P_q(n,m)/g_q(m)$ approaches to 1 uniformly for $q$ and $m \le c \log n$ when $n \to \infty$. This completes the proof. $\square$

In the next theorem we estimate the function $g_q(m) = \prod_{k=1}^{m}\left(1-\frac{1}{q^k}\right)^{I_k}$.

**Theorem 3.2** *For any prime power $q$ and positive integer $m$, we have*

$$\frac{1}{em} \le \exp\left(-\sum_{k=1}^{m}\frac{1}{k}\right) \le \prod_{k=1}^{m}(1-q^{-k})^{I_k} \le \left(1-\frac{1}{\sqrt{q}}\right)^{-\frac{q}{q-1}} \cdot \exp\left(-\sum_{k=1}^{m}\frac{1}{k}\right).$$

*When $q \to \infty$, we have*

$$g_q(m) = \prod_{k=1}^{m}\left(1-\frac{1}{q^k}\right)^{I_k} \to e^{-H_m} \sim \frac{e^{-\gamma}}{m},$$

*where $\gamma$ is the Euler's constant, and $e^{-\gamma} = 0.56416\ldots$*

*Proof.* Note that $\log(1+x) \le x$ for $x > -1$, and $\sum_{k=1}^{\infty}\frac{1}{kq^{k/2}} = -\log\left(1-\frac{1}{\sqrt{q}}\right)$. By (1), we have

$$\begin{aligned}
g_q(m) &= \prod_{k=1}^{m}\exp\left(I_k \log\left(1-\frac{1}{q^k}\right)\right) \\
&\le \prod_{k=1}^{m}\exp\left(-\frac{I_k}{q^k}\right) = \exp\left(-\sum_{k=1}^{m}\frac{I_k}{q^k}\right) \le \exp\left(-\sum_{k=1}^{m}\frac{\frac{q^k}{k}-\frac{q(q^{k/2}-1)}{(q-1)k}}{q^k}\right) \\
&\le \exp\left(-\sum_{k=1}^{m}\frac{1}{k}\right)\cdot \exp\left(\frac{q}{q-1}\sum_{k=1}^{m}\frac{q^{k/2}-1}{kq^k}\right) \\
&\le \exp\left(-\sum_{k=1}^{m}\frac{1}{k}\right)\cdot\left(\exp\left(\sum_{k=1}^{\infty}\frac{1}{kq^{k/2}}\right)\right)^{\frac{q}{q-1}} \\
&= \left(1-\frac{1}{\sqrt{q}}\right)^{-\frac{q}{q-1}}\cdot \exp\left(-\sum_{k=1}^{m}\frac{1}{k}\right).
\end{aligned}$$

We have derived the upper bound for $g_q(m)$. The lower bound was derived by the authors when studying lower bounds for the Euler totient function for polynomials and for the density of normal elements (see [11]). Since it is simple, we reproduce it here. As $I_k \le (q^k-1)/k$ and $0 < 1-1/q^k < 1$, we have

$$\begin{aligned}
g_q(m) &= \prod_{k=1}^{m}\left(1-\frac{1}{q^k}\right)^{I_k} \ge \prod_{k=1}^{m}\left(1-\frac{1}{q^k}\right)^{\frac{q^k-1}{k}} \\
&= \prod_{k=1}^{m}\left(\left(1+\frac{1}{q^k-1}\right)^{q^k-1}\right)^{-\frac{1}{k}} \ge \prod_{k=1}^{m}\exp\left(-\frac{1}{k}\right) = \exp\left(-\sum_{k=1}^{m}\frac{1}{k}\right).
\end{aligned}$$

Let $H_m$ be the harmonic sum, i.e., $H_m = \sum_{k=1}^{m} 1/k$. Then $H_m \leq 1 + \int_1^m 1/x\, dx = 1 + \log m$, and thus $e^{-H_m} \geq 1/(em)$. When $q \to \infty$

$$g_q(m) = \prod_{k=1}^{m} \left(1 - \frac{1}{q^k}\right)^{I_k} \sim e^{-H_m}.$$

Using the well-known approximation of the harmonic sum $H_m = \log m + \gamma + O(\frac{1}{m})$, we have

$$e^{-H_m} \sim \frac{e^{-\gamma}}{m}, \quad m \to \infty. \tag{3}$$

Finally, this result is in accordance with the correspondent one of permutations with no cycles of length $m$ or less (see [29]). $\square$

We provide in Table 1 below the values of $g_q(m)$, $P_q(n,m)$ and their ratio, when $m = \log n$ and $q = 2$ for several $n < 1000$. This shows that the convergence of $P_q(n,m)$ to $g_q(m)$ is very fast. Moreover, as $n$ grows, $P_q(n,m)$ quickly decreases. For instance, for a random polynomial of degree 900, there is a probability of more than 0.9 of having a factor of degree at most 9. This is another explanation for the efficiency of Ben-Or's algorithm. Indeed, it is enough to search for irreducible polynomials of degree at most $O(\log n)$ in order to have a high probability of finding a factor.

For the remaining of this section we concentrate on the probability that a polynomial be irreducible if it has no irreducible factors of low degree. Many algorithms for factoring polynomials over finite fields comprise the following three stages: *squarefree factorization* (replace a polynomial by a squarefree one which contains all the irreducible factors of the original polynomial with exponents reduced to 1); *distinct-degree factorization* (split a squarefree polynomial into a product of polynomials whose irreducible factors have all the same degree); and *equal-degree factorization* (factor a polynomial whose irreducible factors have the same degree).

As things now stand, distinct-degree factorization is the bottleneck of the polynomial factorization problem (see [14], [18], [13]). This step of the factorization process works as follows: at any point $k$, all the irreducible factors of degree up to $k$ have been found, and all the irreducible factors of degree greater than $k$ remain to be determined from a factor $g$.

A natural way of improving the distinct-degree factorization step is by testing the irreducibility of the remaining factor $g$. Unfortunately, in the worst-case asymptotic scenario, the cost of the irreducibility test is about the same as the distinct-degree factorization algorithm. An alternative to overcome this problem is given in [12] (§6). The central idea is to run the irreducibility test and the distinct-degree factorization algorithm in parallel, feeding the former with partial information obtained by the latter (see the details in [12]).

It is clear that the probability of a monic polynomial being irreducible when it has no irreducible factors of low degree provides useful information in the above process. In the following, we derive an asymptotic formula for this probability.

| $n$ | $m$ | $P_q(n,m)$ | $g_q(m)$ | P/g |
|---:|---:|---:|---:|---:|
| 2 | 1 | .2500000000000000000000000 | .2500000000000000000000000 | 1.0000 |
| 3 | 1 | .2500000000000000000000000 | .2500000000000000000000000 | 1.0000 |
| 4 | 2 | .1875000000000000000000000 | .1875000000000000000000000 | 1.0000 |
| 5 | 2 | .1875000000000000000000000 | .1875000000000000000000000 | 1.0000 |
| 6 | 2 | .1875000000000000000000000 | .1875000000000000000000000 | 1.0000 |
| 7 | 2 | .1875000000000000000000000 | .1875000000000000000000000 | 1.0000 |
| 8 | 3 | .1406250000000000000000000 | .1435546875000000000000000 | .97963 |
| 9 | 3 | .1445312500000000000000000 | .1435546875000000000000000 | 1.0068 |
| 10 | 3 | .1435546875000000000000000 | .1435546875000000000000000 | .99997 |
| 20 | 4 | .1182861328125000000000000 | .1182854175567626953125000 | 1.0000 |
| 30 | 4 | .1182854175567626953125000 | .1182854175567626953125000 | 1.0000 |
| 40 | 5 | .09776907367631793022155762 | .09776907366723652792472876 | 1.0000 |
| 50 | 5 | .09776907366723719405854354 | .09776907366723652792472876 | 1.0000 |
| 60 | 5 | .09776907366723652792472876 | .09776907366723652792472876 | 1.0000 |
| 70 | 6 | .08484899050039278356888779 | .08484899050039278175814854 | 1.0000 |
| 80 | 6 | .08484899050039278175860054 | .08484899050039278175814854 | 1.0000 |
| 90 | 6 | .08484899050039278175814857 | .08484899050039278175814854 | 1.0000 |
| 100 | 6 | .08484899050039278175814854 | .08484899050039278175814854 | 1.0000 |
| 200 | 7 | .07367738498865927351164168 | .07367738498865927351164168 | .99999 |
| 300 | 8 | .06551498664534958936373162 | .06551498664534958936373162 | 1.0000 |
| 400 | 8 | .06551498664534958936373162 | .06551498664534958936373162 | 1.0000 |
| 500 | 8 | .06551498664534958936373162 | .06551498664534958936373162 | 1.0000 |
| 600 | 9 | .05872097388539275926570230 | .05872097388539275926570230 | 1.0000 |
| 700 | 9 | .05872097388539275926570230 | .05872097388539275926570230 | 1.0000 |
| 800 | 9 | .05872097388539275926570230 | .05872097388539275926570230 | 1.0000 |
| 900 | 9 | .05872097388539275926570230 | .05872097388539275926570230 | 1.0000 |

**Table 1.** Values of $P_q(n,m)$ and $g_q(m)$, with $m = \log n$ and $q = 2$.

**Theorem 3.3** *Let $P_q^I(n,m)$ be the probability that a polynomial of degree $n$ over $\mathbb{F}_q$ be irreducible if it has no factors of degree less than or equal to $m$, $1 \leq m \leq O(\log n)$. Then, as $n$, $m$ and $q$ approach to infinity,*

$$P_q^I(n,m) \sim e^{\gamma}\,\frac{m}{n},$$

*where $\gamma$ is the Euler's constant.*

*Proof.* This probability can be estimated considering the subset of irreducible polynomials of degree $n$ over $\mathbb{F}_q$ inside the set of polynomials of degree $n$ over $\mathbb{F}_q$ without irreducible factors of degree less than or equal to $m$, $1 \leq m \leq O(\log n)$. Using (1), Theorems 3.1 and 3.2, when $n$, $m$ and $q$ approach to infinity, we obtain

$$P_q^I(n,m) = \frac{I_n}{q^n\,P_q(n,m)} \sim \frac{\frac{1}{n}}{\frac{e^{-\gamma}}{m}} = e^{\gamma}\,\frac{m}{n}.$$

## 4 Experimental results

In this section, we describe an implementation of the algorithms discussed in Section 2. We provide a running time comparison of the algorithms for random polynomials on a Sun Sparc 20 computer. The algorithms were implemented on a C++ software due to Shoup. This package contains classes for finite fields and polynomials over finite fields with implementations for basic operations such as multiplication, taking gcd, and so on (for a description of the software see [32]).

We tested all algorithms with the same random polynomials. A summary table for the average time in seconds of CPU in the case $\mathbb{F}_2$ is presented in Table 2 and Table 3. The degrees $n$ in Table 2 and in Table 3 were chosen such that $n$ has many prime divisors and few prime divisors, respectively. The number of polynomials tested was $10 * n$ for Table 2 and $5 * n$ for Table 3, where $n$ is the degree of the polynomials being tested for irreducibility. It can be seen from the Tables that even in the case of $n$ with few prime divisors Ben-Or has the best behavior among the three algorithms. The worst-case scenario for these algorithms happens when testing irreducible polynomials. We include a column with the number of irreducible polynomials that were tested for each degree.

| $n$ | Rabin | Rabin's Variant | Ben-Or | Number of Irreducible |
|---|---|---|---|---|
| 105 | 0.7990 | 0.6133 | 0.2000 | 9 |
| 210 | 2.7652 | 2.6942 | 0.9938 | 14 |
| 330 | 10.5672 | 17.4147 | 2.3443 | 8 |
| 420 | 10.5702 | 4.3971 | 1.5189 | 7 |

**Table 2.** Average time in seconds for testing $10 * n$ polynomials over $\mathbb{F}_2$ of degree $n$ with many prime divisors.

| $n$ | Rabin | Rabin's Variant | Ben-Or | Number of Irreducible |
|---|---|---|---|---|
| 101 | 4.1564 | 8.9188 | 0.5901 | 4 |
| 256 | 20.8451 | 46.9867 | 2.7950 | 5 |
| 331 | 32.7156 | 71.8685 | 2.7719 | 9 |

**Table 3.** Average time in seconds for testing $5 * n$ polynomials over $\mathbb{F}_2$ of degree $n$ with few prime divisors.

Similar results for testing irreducibility of polynomials over the finite field $\mathbb{F}_{1021}$ are presented in Table 4 below.

| $n$ | Rabin | Rabin's Variant | Ben-Or | Number of Irreducible |
|---|---|---|---|---|
| 101 | 186.1383 | 280.2633 | 13.0198 | 4 |
| 105 | 78.2952 | 61.8381 | 11.2629 | 5 |
| 210 | 227.5800 | 174.7400 | 23.0000 | 3 |

**Table 4.** Average time in seconds for testing $5 * n$ polynomials of degree $n$ over $\mathbb{F}_{1021}$.

We also tested the case of very large fields. For instance, the average time in seconds of CPU for testing 315 polynomials of degree 105 over $\mathbb{F}_p$, $p$ a prime with 100 bits, was:

| | |
|---|---|
| Rabin | 774.054 |
| Rabin's variant | 613.267 |
| Ben-Or | 137.565 |

In this case, 5 irreducible polynomials were found.

These timings suggest that Ben-Or's algorithm has a much better average time behavior than others, even though its worst-case complexity is the worst among them. A variant of the above algorithms that economizes gcd's computations can be given using Ben-Or's ideas up to $O(\log n)$ iterations and our Rabin's variant after that point. For this algorithm together with more experimental results, see [25].

## 5 Construction of sparse irreducible polynomials

When implementing the irreducibility tests in Section 2, we wanted to experiment our programs on various polynomials of large degrees. For reducible polynomials, they are likely to have small factors, as shown in Section 3, and the programs terminate almost immediately. However, when testing an irreducible polynomial, we do not have a priori any idea of how long it will take to complete the task. It is desirable to have some simple polynomials which we know in advance are irreducible so that we can test the correctness of our programs and know the approximate time our computer needs on various degrees. This would also help us in deciding the range of degrees to compare the tests. By "simple", we mean polynomials that can either be constructed easily (without testing for irreducibility) or have only a few nonzero terms. The problem of constructing sparse irreducible polynomials is also of independent interest.

A well-known open problem is to construct irreducible polynomials over $\mathbb{F}_2$ of degree $n$ with at most $O(\log n)$ nonzero terms in its lowest coefficients. These polynomials are useful in the discrete logarithm problem ([6], [23]). Shoup (private communication, 1994) points out that if $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/(f)$ with $f = x^n + g$ irreducible and $g \in \mathbb{F}_2[x]$ of small degree, say $\deg g \le 2 \log n$, then exponentiation in $\mathbb{F}_{2^n}$ can be achieved with $O(n^2 \log\log n)$ operations in $\mathbb{F}_2$ and using storage for $O(n/\log n)$ elements from $\mathbb{F}_{2^n}$ (see also [9]). Experimental results show that such polynomial $f$ exists for $n \le 1000$ taking $\deg g \le 2 + \log_2 n$.

Shparlinski ([33]) gives a construction of irreducible polynomials with degrees of the form $4 \cdot 3^k \cdot 5^\ell$ over $\mathbb{F}_2$, $4 \cdot 2^k \cdot 5^\ell$ over $\mathbb{F}_3$, and $2 \cdot 2^k \cdot 3^\ell$ over $\mathbb{F}_p$, for any prime $p > 3$, and $k, l$ nonnegative integers. In the following, we first generalize his construction and then construct explicitly several infinite families of irreducible polynomials.

**Theorem 5.1** *Let $p_1, \ldots, p_k$ be any fixed primes and $n = mp_1^{e_1} \ldots p_k^{e_k}$ where $m$ is the multiplicative order of $q$ modulo $p_1 \cdots p_k$ and $e_1, \ldots, e_k$ are arbitrary nonnegative integers. Then, over any finite field $\mathbb{F}_q$ whose characteristic is distinct from $p_1, \ldots, p_k$, there is an irreducible polynomial of degree $n$ with at most $2m + 1 \le 2(p_1 - 1) \cdots (p_k - 1) + 1$ nonzero terms.*

*Proof.* Let $\ell = m$ if all $p_i$ are odd. If one of $p_i$, say $p_1$, is 2 and $q \equiv 3 \bmod 4$, let $\ell = \mathrm{lcm}(2, m)$. Then, $p_i | (q^\ell - 1)$ for $1 \le i \le \ell$, and $4 | (q^\ell - 1)$ if $p_1$ is even.

Let $\beta$ be an element in $\mathbb{F}_{q^\ell}$ that is not a $p_i$th power in $\mathbb{F}_{q^\ell}$ for $1 \le i \le k$. The minimal polynomial $\zeta(x)$ of $\beta$ over $\mathbb{F}_q$ has degree $\ell$. By Lidl and Niederreiter ([21], p. 124, Theorem 3.75),

$$x^{p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}} - \beta$$

is irreducible over $\mathbb{F}_{q^\ell}$ for all nonnegative integers $e_1, \ldots, e_k$. Then

$$\zeta\left(x^{p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}}\right) \tag{4}$$

is irreducible over $\mathbb{F}_q$ of degree $\ell p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$. The polynomial (4) has at most $\ell + 1 \le 2m + 1$ nonzero terms.

Irreducible polynomials from the construction of Theorem 5.1 are usually very sparse. In fact, the number of nonzero terms depends only on the prime factors of $n$, and if we fix them and let the exponents grow arbitrarily then these polynomials have only $O(1)$ nonzero terms (even though not necessarily the lowest coefficients). This can be seen from the following examples.

**Example 5.2** *$q \equiv 1 \bmod 4$ and $n = 2^k$. Take $a \in \mathbb{F}_q$ to be any quadratic nonresidue. Then $x^{2^k} - a$ is irreducible over $\mathbb{F}_q$ for all $k \ge 0$. For instance,*

*i. if $q = p \equiv \pm 3 \bmod 8$ is a prime, then take $a = 2$;*
*ii. if $q = p \equiv \pm 5 \bmod 12$ is a prime, then take $a = 3$;*
*iii. if $q = p \equiv \pm 2 \bmod 5$ is a prime, then take $a = 5$.*

*These results are from [17] (Proposition 5.1.3 for the first two, and Theorem 2, p. 54, for the third).*

**Example 5.3** *$q \equiv 3 \bmod 4$ and $n = 2^k$. In this case, $\ell = 2$ in the proof of Theorem 5.1. We need to find a quadratic nonresidue in $\mathbb{F}_{q^2}$ and compute its minimal polynomial. The following elegant solution is from [3]. Suppose that $q = p^m$ where $m$ is odd and $p \equiv 3 \bmod 4$ is a prime. Let $2^v | (p+1)$, $2^{v+1} \not| (p+1)$. Then $v \geq 2$. Construct $u \in \mathbb{F}_p$ iteratively as follows:*

$$\begin{aligned} u_1 &= 0, \\ u_i &= \pm \left( \frac{u_{i-1}+1}{2} \right)^{\frac{p+1}{4}} \pmod p, \quad \textit{for } 1 < i < v, \\ u_v &= \pm \left( \frac{u_{v-1}-1}{2} \right)^{\frac{p+1}{4}} \pmod p, \end{aligned}$$

*where, at each step, one can take any of the signs arbitrarily. Let $u = u_v$. Then $x^2 - 2ux - 1$ is the minimal polynomial of some quadratic nonresidue in $\mathbb{F}_{p^2}$. Therefore*

$$x^{2^k} - 2ux^{2^{k-1}} - 1$$

*is irreducible over $\mathbb{F}_p$, and over $\mathbb{F}_q$ as well, for all $k \geq 1$.*

**Example 5.4** *$q = 2$. Theorem 5.1 yields the following families of irreducible polynomials over $\mathbb{F}_2$ for all $k, \ell, m, n \geq 0$:*

$$\begin{aligned} &x^{2\cdot 3^k} + x^{3^k} + 1 \\ &x^{3\cdot 7^k} + x^{7^k} + 1 \\ &x^{4\cdot 3^k\cdot 5^\ell} + x^{3^k\cdot 5^\ell} + 1 \\ &x^{6\cdot 3^k\cdot 7^\ell} + x^{3^k\cdot 7^\ell} + 1 \\ &x^{10\cdot 3^k\cdot 11^\ell\cdot 31^m} + x^{3\cdot 3^k\cdot 11^\ell\cdot 31^m} + 1 \\ &x^{12\cdot 3^k\cdot 5^\ell\cdot 7^m\cdot 13^n} + x^{8\cdot 3^k\cdot 5^\ell\cdot 7^m\cdot 13^n} + x^{2\cdot 3^k\cdot 5^\ell\cdot 7^m\cdot 13^n} + x^{3^k\cdot 5^\ell\cdot 7^m\cdot 13^n} + 1. \end{aligned}$$

For other explicit constructions of irreducible polynomials, see [22] (Chapter 3), and [10].

**Acknowledgment.** We thank Bruce Richmond for helpful discussions and Victor Shoup for the use of his package. Part of this work was done by the first author while visiting the University of Waterloo whose hospitality and support are gratefully acknowledged.

## References

1. Ben-Or, M. Probabilistic algorithms in finite fields. In *Proc. 22nd IEEE Symp. Foundations Computer Science* (1981), pp. 394–398.
2. Blake, I., Gao, S., and Lambert, R. Constructive problems for irreducible polynomials over finite fields. In *Information Theory and Applications*, A. Gulliver and N. Secord, Eds., vol. 793 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994, pp. 1–23.

3. Blake, I., Gao, S., and Mullin, R. Explicit factorization of $x^{2^k}+1$ over $\mathbb{F}_p$ with prime $p \equiv 3 \pmod 4$. *Appl. Alg. Eng. Comm. Comp. 4* (1993), 89–94.
4. Butler, M. On the reducibility of polynomials over a finite field. *Quart. J. Math. Oxford 5* (1954), 102–107.
5. Cantor, D., and Kaltofen, E. On fast multiplication of polynomials over arbitrary algebras. *Acta. Inform. 28* (1991), 693–701.
6. Coppersmith, D. Fast evaluation of logarithms in fields of characteristic two. *IEEE Trans. Info. Theory 30* (1984), 587–594.
7. Flajolet, P., Gourdon, X., and Panario, D. Random polynomials and polynomial factorization. In *Proc. 23rd ICALP Symp.* (1996), vol. 1099 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 232–243.
8. Flajolet, P., and Odlyzko, A. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics 3 2* (1990), 216–240.
9. Gao, S., von zur Gathen, J., and Panario, D. Gauss periods and efficient arithmetic in finite fields. Submitted to *Journal of Symbolic Computation* (extended abstract in *Proc. LATIN'95*, vol. 911 of *Lecture Notes in Computer Science*, 311–322), 1995.
10. Gao, S., and Mullen, G. Dickson polynomials and irreducible polynomials over finite fields. *J. Number Theory 49* (1994), 118–132.
11. Gao, S., and Panario, D. Density of normal elements. Submitted to *Finite Fields and their Applications* (abstract in *AMS Abstracts* **102** Fall 1995, # 904-68-227, p. 798), 1995.
12. von zur Gathen, J., and Gerhard, J. Arithmetic and factorization of polynomials over $\mathbb{F}_2$. In *Proc. ISSAC'96, Zürich, Switzerland* (1996), L. Y.N., Ed., ACM press, pp. 1–9.
13. von zur Gathen, J., and Panario, D. A survey on factoring polynomials over finite fields. Submitted to the special issue of the MAGMA conference in *J. Symb. Comp.*, 1996.
14. von zur Gathen, J., and Shoup, V. Computing Frobenius maps and factoring polynomials. *Comput complexity 2* (1992), 187–224.
15. Golomb, S. W. *Shift register sequences.* Aegean Park Press, Laguna Hills, California, 1982.
16. Graham, R., Knuth, D., and Patashnik, O. *Concrete Mathematics*, 2nd ed. Addison-Wesley, Reading, MA, 1994.
17. Ireland, K., and Rosen, M. *A Classical Introduction to Modern Number Theory*, 2nd ed. Springer-Verlag, Berlin, 1990.
18. Kaltofen, E., and Shoup, V. Subquadratic-time factoring of polynomials over finite fields. In *Proc. 27th ACM Symp. Theory of Computing* (1995), pp. 398–406.
19. Knopfmacher, J., and Knopfmacher, A. Counting irreducible factors of polynomials over a finite field. *SIAM Journal on Discrete Mathematics 112* (1993), 103–118.
20. Knuth, D. *The art of computer programming, vol.2: seminumerical algorithms*, 2nd ed. Addison-Wesley, Reading MA, 1981.
21. Lidl, R., and Niederreiter, H. *Finite fields*, vol. 20 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley, Reading MA, 1983.
22. Menezes, A., Blake, I., Gao, X., Mullin, R., Vanstone, S., and Yaghoobian, T. *Applications of Finite Fields.* Kluwer Academic Publishers, Boston, Dordrecht, Lancaster, 1993.

23. Odlyzko, A. Discrete logarithms and their cryptographic significance. In *Advances in Cryptology, Proceedings of Eurocrypt 1984* (1985), vol. 209 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 224–314.
24. Odlyzko, A. Asymptotic enumeration methods. In *Handbook of Combinatorics*, R. Graham, M. Grötschel, and L. Lovász, Eds. Elsevier, 1996.
25. Panario, D. Combinatorial and algebraic aspects of polynomials over finite fields. PhD Thesis, in preparation, 1996.
26. Rabin, M. O. Probabilistic algorithms in finite fields. *SIAM J. Comp. 9* (1980), 273–280.
27. Schönhage, A. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Inf. 7* (1977), 395–398.
28. Schönhage, A., and Strassen, V. Schnelle Multiplikation großer Zahlen. *Computing 7* (1971), 281–292.
29. Sedgewick, R., and Flajolet, P. *An Introduction to the Analysis of Algorithms.* Addison-Wesley, Reading MA, 1996.
30. Shepp, L., and Lloyd, S. Ordered cycle lengths in a random permutation. *Trans. Amer. Math. Soc. 121* (1966), 340–357.
31. Shoup, V. Fast construction of irreducible polynomials over finite fields. *J. Symb. Comp. 17* (1995), 371–391.
32. Shoup, V. A new polynomial factorization algorithm and its implementation. *J. Symb. Comp. 20* (1996), 363–397.
33. Shparlinski, I. Finding irreducible and primitive polynomials. *Appl. Alg. Eng. Comm. Comp. 4* (1993), 263–268.

**Note**. This paper has appeared in *Foundations of Computational Mathematics*, F. Cucker and M. Shub (Eds.), Springer 1997, 346–361.