Karatsuba Algorithm for multiplication of Linearized Polynomials

Thesis BSc Computing Science

**Author:**
Niels Feij

**Supervisor:**
Simona Samardziska

**Second reader:**
Peter Schwabe

June 2023
Abstract

This paper explores multiplication of linearized polynomials in the field $\mathbb{F}_{q^m}$ as an alternative for matrix multiplications in the field $\mathbb{F}_{n \times n}^q$. This is done by means of an implementation of the Karatsuba Algorithm for linearized polynomials. A comparison based on time complexity is made against Schoolbook multiplication. This comparison shows the Karatsuba algorithm to be faster for linearized polynomials of degree $n > 19$ independent of $q$ and for even lower degrees $n$ when $q$ gets larger. Finally, this research shows naive matrix multiplication in $\mathbb{F}_{n \times n}^q$ to yield the most efficient computation over Karatsuba linearized polynomials multiplications over $\mathbb{F}_{q^m}$.

1 Introduction

The security of the majority of our digital assets and online activities relies on the strength of the underlying cryptographic techniques. Public-key cryptography, specifically RSA [8] and Elliptic Curve Cryptography [3], plays a crucial role in establishing a secure cryptographic infrastructure.

However, with the continuous advancements in quantum computing, the long-term security of this infrastructure, including (previously) encrypted information and digital signatures, is being compromised. Once a fully functional quantum computer becomes available, all the currently standardized and widely-used public-key algorithms can be vulnerable to attacks that can be carried out in polynomial time using a quantum computer.

In response to this imminent threat to our existing public-key infrastructure, the National Institute of Standards and Technology (NIST) initiated a process in 2016 to seek, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms [6]. The goal is to have a new replacement standard in place by 2024. These algorithms are commonly referred to as post-quantum or quantum-safe algorithms.

Some of these algorithms tend to make use of matrix-heavy computations, mainly multiplications. This enforces researchers to look for better and faster alternatives. In this paper this alternative comes in the form of multiplication of linearized polynomials. Although a linearized polynomial and a matrix are different mathematical structures, they can be used to represent the same elements, thereby deeming the comparison of computational complexity in their respective 'worlds' interesting. This paper defines, explores and compares the implementation of the Karatsuba Algorithm (KA) [2], designed for the multiplication of multi-digit numbers, for the multiplication of linearized polynomials. A multiplication with lower complexity in the 'world' of linearized polynomials (i.e. $\mathbb{F}_{q^m}$) might result in reconsideration of matrices (i.e. $\mathbb{F}_{n \times n}^q$) being the industry standard 'world' of computation.

In order to simplify the problem we assume that the maximum degree of every two elements which are multiplied is identical.

The work is organized as follows. Firstly, the concepts of the KA and linearized polynomials will be made clear. Secondly, case studies for the KA for linearized polynomials of degree 1, 2 and 3 are performed. Next, the outcomes of these exploratory applications will be combined in a generalization, which will be elaborated, evaluated and compared to Schoolbook multiplication. Lastly, KA for linearized polynomials will be compared to naive matrix multiplication (MM).
1.1 Related work: Karatsuba Algorithm with polynomial multiplication

In 2012, S. Mishra and M. Pradhan [5] used polynomial multiplication in the classical KA to multiply two numbers. This lead to to a single and recursive algorithm which has better time performance over KA with regular multiplication.

1.2 Karatsuba Algorithm for polynomial multiplication

In 2016, A. Weimerskirch and C. Paar [9] generalized the classical KA for polynomial multiplication to (i) polynomials of arbitrary degree and (ii) recursive use. Their research provides tables that describe the best possible usage of the KA for polynomials up to a degree of 127. The results of the paper are especially useful for efficient implementations of computations over fixed-size fields like \( \mathbb{F}_{p^m} \).

2 Preliminaries: Schoolbook for polynomials

The naive way to multiply two polynomials is often referred to as the schoolbook method. Let \( A(x) \) and \( B(x) \) be polynomials of degree-\( d \) with \( n = d + 1 \) coefficients:

\[
A(x) = \sum_{i=0}^{d} a_i x^i, \quad B(x) = \sum_{i=0}^{d} b_i x^i
\]

Then the product \( C(x) = A(x)B(x) \) is defined as

\[
\sum_{i=0}^{d} \sum_{j=0}^{d} a_i b_j x^{i+j}
\] (1)

With \( A(x) \) and \( B(x) \) being 'simple' polynomials, this multiplication takes \( n^2 \) multiplications and \( (n-1)^2 \) additions.

2.1 Karatsuba Algorithm

The KA [2] makes uses of the reoccurring factors in a multiplication. It has a single iteration variant as well as a recursive (divide-and-conquer) variant. The classic recursive KA simplifies the multiplication of two \( n \)-digit numbers into three multiplications of \( n/2 \)-digit numbers by making use of reoccurring factors. By repeating this reduction process, it ensures a maximum of \( n^{\log_2 3} = n^{1.58} \) single-digit multiplications. This makes it significantly faster than the traditional algorithm, which performs \( n^2 \) single-digit products.

As mentioned in section 1.2, KA also has an implementation for the multiplication of polynomials which is derived by simple algebraic transformations of the naive (Schoolbook) multiplication. Multiplying two polynomials using the single iteration KA results in a \( \frac{1}{2}n^2 + \frac{1}{2}n \) multiplications, and \( \frac{5}{2}n^2 - \frac{7}{2}n + 1 \) additions [9].
2.1.1 Example Karatsuba Algorithm for polynomials

For sake of clarification, let us work out the simplest instance of the KA for polynomials. Let \( A(x) \) and \( B(x) \) be degree-1 polynomials:

\[
A(x) = a_1 x + a_0, \quad B(x) = b_1 x + b_0
\]

Let there be three auxiliary variables \( D_0, D_1 \) and \( D_{0,1} \) given by:

\[
D_0 = a_0 b_0
\]

\[
D_1 = a_1 b_1
\]

\[
D_{0,1} = (a_0 + a_1)(b_0 + b_1)
\]

Then the polynomial \( C(x) = A(x)B(x) \) is computed by:

\[
C(x) = D_1 x^2 + (D_{0,1} - D_0 - D_1) x + D_0
\]

Using this method, we need 3 multiplication and 4 additions, with the schoolbook method we needed 4 multiplication and 1 addition. Thus winning 1 multiplication at the cost of 3 additional additions.

2.2 Linearized polynomials

Linearized polynomials [7] are polynomials of the following form:

\[
a = \sum_{k=0}^{t} a_k x^{q^k}, \quad a_k \in \mathbb{F}_{q^m}
\]

Where \( q \) is a prime power, \( \mathbb{F}_q \) is a finite field with \( q \) elements and \( \mathbb{F}_{q^m} \) is a extension field over \( \mathbb{F}_q \). The notation \([k]\) is used to denote \( q^k \). We say linearized polynomial \( a \) has \( q \)-degree \( \deg_a = \max\{k \in \mathbb{N} : a_k \neq 0\} \).

Multiplication of two linearized polynomials \( A(x) \) and \( B(x) \) is defined as the composition

\[
C(x) = A(B(x))
\]

The following section derives another general equation to represent multiplication for linearized polynomials which shows similarity to the Schoolbook algorithm for all polynomials.

2.3 Schoolbook for linearized polynomials

Let \( A(x) \) and \( B(x) \) be linearized polynomials with \( q \)-degree 2:

\[
A = a_0 x^{[0]} + a_1 x^{[1]} + a_2 x^{[2]}
\]

\[
B = b_0 x^{[0]} + b_1 x^{[1]} + b_2 x^{[2]}
\]
Then the composition will be of the form:

\[
A(B(x)) = a_0 \left( b_0 x^{[0]} + b_1 x^{[1]} + b_2 x^{[2]} \right)^{[0]} + a_1 \left( b_0 x^{[0]} + b_1 x^{[1]} + b_2 x^{[2]} \right)^{[1]}
+ a_2 \left( b_0 x^{[0]} + b_1 x^{[1]} + b_2 x^{[2]} \right)^{[2]}
\]

\[
= a_0 b_0^{[0]} x^{[0][0]} + a_0 b_1^{[0]} x^{[1][0]} + a_0 b_2^{[0]} x^{[2][0]} + a_1 b_0^{[1]} x^{[0][1]} + a_1 b_1^{[1]} x^{[1][1]} + a_1 b_2^{[1]} x^{[2][1]}
+ a_2 b_0^{[2]} x^{[0][2]} + a_2 b_1^{[2]} x^{[1][2]} + a_2 b_2^{[2]} x^{[2][2]}
\]

\[
= a_0 b_0^{[0]} x^{[0]} + a_0 b_1^{[0]} x^{[1]} + a_0 b_2^{[0]} x^{[2]} + a_1 b_0^{[1]} x^{[1]} + a_1 b_1^{[1]} x^{[2]}
+ a_2 b_0^{[2]} x^{[2]} + a_2 b_1^{[2]} x^{[3]} + a_2 b_2^{[2]} x^{[4]}
\]

Since:

\[
x^{[1][2]} = x^{p^1 \times p^2}
= x^{p^1 + p^2}
= x^{p^3} = x^{[3]}
\]

Further simplification gives the following equation:

\[
\sum_{i=0}^{2} \left( \sum_{j=0}^{2} a_i b_j^{[i]} x^{[i+j]} \right) = \sum_{i=0}^{deg_a} \sum_{j=0}^{deg_b} a_i b_j^{[i]} x^{[i+j]}
\]

Which, since linearized polynomials are a subset of all polynomials, logically has the same general form as equation (1) and can therefore be seen as the Schoolbook multiplication of two linearized polynomials. This computation takes the same amount of multiplications \((n^2)\) and additions \(((n - 1)^2)\), however the exponents of the \(b\)-term require an additional \(n^2\) exponentiation operations over the earlier stated Schoolbook complexity. All operations are done in \(F_{q^m}\).

### 2.4 Normal bases

Normal bases facilitate calculations in finite fields and can therefore be used to reduce the computational complexity. We shortly summarize important properties of normal bases in the following [1]. A basis \(B = \{\beta_0, \beta_1, \ldots, \beta_{m-1}\} \) of \(F_{q^m}\) over \(F_q\) is a normal basis if \(\beta_i = \beta^{i}\) for all \(i\), where \(\beta \in F_{q^m}\) is called a normal element. As shown in [4], there is a normal basis for any finite extension field \(F_{q^m}\) over \(F_q\). If we represent elements of \(F_{q^m}\) in a normal basis over \(F_q\), the operation \(a \rightarrow a^{[j]}\) where \(a \in F_{q^m}\), an operation known as the Frobenius automorphism, can be accomplished in \(O(1)\) operations over \(F_{q^m}\) as follows. Let \(A = [A_1, \ldots, A_m]^T \in F_q^{m \times 1}\) be the vector representation of \(a \in F_{q^m}\) in a normal basis. Then, for any \(j\), the vector representation of \(a^{[j]}\) is given by \(\left[ A_{m-j}, A_{m-j+1}, \ldots, A_0, A_1, \ldots, A_{m-j-1} \right]^T\), which is just a cyclic shift of the representation of \(a\).

### 3 Karatsuba for linearized polynomials

In order to find the generalization of the algorithm for arbitrary degree linearized polynomials, this paper explores multiplication of linearized polynomials of \(q\)-degree 1, 2 and 3. We will encounter the transformation \(a_0 b_0^{[1]} \rightarrow a_0 b_0^{[0]}\) and explore it. Finally, a concluding generalization is defined.
3.1 Karatsuba algorithm for \( q \)-degree 1 linearized polynomial (e.g. \( a_1x^q + a_0x^q \))

Let there be two \( q \)-degree 1 linearized polynomials \( A(x) \) and \( B(x) \) given by:

\[
A(x) = a_1x^1 + a_0x^0 \quad \text{and} \quad B(x) = b_1x^1 + b_0x^0
\]

Then, following the definition of multiplication for linearized polynomials:

\[
\sum_{i=0}^{\text{deg} A} \sum_{j=0}^{\text{deg} B} a_i b_j^i x^{i+j}
\]

the product \( C(x) = A(x) B(x) \) can be determined in the following manner:

\[
C(x) = a_1b_1^1 x^2 + a_1b_0^1 x^1 + a_0b_0^0 x^1 + a_0b_0^0 x^0
\]

\[
= a_1b_1^1 x^2 + (a_1b_0^1 + a_0b_0^0) x^1 + a_0b_0^0 x^0
\]

The coefficient of \( x^1 \) in the above linearized polynomial can be written as:

\[
(a_1b_0^1 + a_0b_0^0) = ((a_0 + a_1)(b_0^1 + b_1^0) - a_0b_1^1 - a_1b_0^0)
\]

Let there be three auxiliary variables \( D_0 \), \( D_1 \), and \( D_{0,1} \) given by:

\[
D_0 = a_0b_0^0
\]
\[
D_1 = a_1b_1^0
\]
\[
D_{0,1} = (a_0 + a_1)(b_0^1 + b_1^0)
\]

Then the linearized polynomial \( C'(x) \) can be written as:

\[
C'(x) = D_1x^2 + (D_{0,1} - D_0 - D_1)x^1 + D_0x^0
\]

\[
= a_1b_1^0 x^2 + (a_0 + a_1)(b_0^1 + b_1^0) - a_0b_0^0 - a_1b_1^1 x^1 + a_0b_0^0 x^0
\]

\[
= a_1b_1^0 x^2 + ((a_0b_0^1 + a_0b_0^0 + a_1b_0^1 + a_1b_0^0) - a_0b_1^1 - a_1b_0^0) x^1 + a_0b_0^0 x^0
\]

\[
= a_1b_1^0 x^2 + (a_1b_0^0 + a_0b_0^0) x^1 + a_0b_0^0 x^0 S
\]

Which looks similar to, but is not, the result \( C(x) \) that we are looking for.

\[
C(x) = a_1b_1^1 x^2 + (a_1b_0^0 + a_0b_1^0) x^1 + a_0b_0^0 x^0
\]
\[
C'(x) = a_1b_1^0 x^2 + (a_1b_0^0 + a_0b_0^0) x^1 + a_0b_0^0 x^0
\]

In order to improve in terms of complexity, we need the following operations:

\[
a_0b_0^1 \rightarrow a_0b_0^0
\]
\[
a_1b_0^0 \rightarrow a_1b_1^1
\]

to be computational light (e.g. not factoring \( a_0b_0^1 \), lowering the exponent of \( b \) and multiply).
3.1.1 Exploring $a_0b_0^{[1]} \rightarrow a_0b_0^{[0]}$

In order to find a possible relation between $a_0b_0^{[1]}$ and $a_0b_0^{[0]}$, let us work out the terms. We will use the normal basis representation of the individual elements, with the use of a normal basis $(B, B^q, ..., B^{q^{m-1}})$. This results in the following two vector representations:

$$a_0 = (\alpha_1, \alpha_2, ..., \alpha_m)$$
$$b_0 = (\beta_1, \beta_2, ..., \beta_m)$$

Then for any $j$ the vector representation of $a^{[j]}$ is given by:

$$a^{[j]} = (\alpha_{m-j}, \alpha_{m-j+1}, ..., \alpha_{m-1})$$

(3)

The first operation is the exponentiation to get $b_0^{[0]}$ and $b_0^{[1]}$ from $b_0$. Following the Frobenius automorphism for normal bases, we get:

$$b_0^{[0]} = (\beta_{m-0}, \beta_{m-0+1}, ..., \beta_{m-0-2}, \beta_{m-0-1})$$
$$= (\beta_m, \beta_1, ..., \beta_{m-2}, \beta_{m-1})$$

And:

$$b_0^{[1]} = (\beta_{m-1}, \beta_{m-1+1}, ..., \beta_{m-1-2}, \beta_{m-1-1})$$
$$= (\beta_{m-1}, \beta_{m}, ..., \beta_{m-3}, \beta_{m-2})$$

Let us write out $a_0b_0^{[0]}$ with simple algebraic operations for proving purposes:

$$a_0b_0^{[0]} = b_0^{[0]}a_0$$
$$= (\beta_m, \beta_1, ..., \beta_{m-2}, \beta_{m-1})(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m)$$
$$= (\beta_m(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m),$$
$$\beta_1(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m),$$
$$...,\beta_{m-2}(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m),\beta_{m-1}(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m))$$

Let us now write out $a_0b_0^{[1]}$ with simple algebraic operations for proving purposes:

$$a_0b_0^{[1]} = b_0^{[1]}a_0$$
$$= (\beta_{m-1}, \beta_m, ..., \beta_{m-3}, \beta_{m-2})(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m)$$
$$= (\beta_{m-1}(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m),$$
$$\beta_m(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m),$$
$$...,\beta_{m-3}(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m),\beta_{m-2}(\alpha_1, \alpha_2, ..., \alpha_{m-1}, \alpha_m))$$
For simplification purposes, let us now define $\gamma$ to represent the basis of $a_0$:

$$\gamma = (\alpha_1, \alpha_2, \ldots, \alpha_{m-1}, \alpha_m)$$

This now gives:

$$a_0 b_0^0 = (\beta_m \gamma, \beta_1 \gamma, \ldots, \beta_{m-2} \gamma, \beta_{m-1} \gamma)$$

And:

$$a_0 b_0^1 = (\beta_{m-1} \gamma, \beta_m \gamma, \ldots, \beta_{m-3} \gamma, \beta_{m-2} \gamma)$$

Which shows us the following equivalence:

$$a_0 b_0^0 \rightarrow a_0 b_0^1 \equiv a_0 b_0^0 \gg a_0 b_0^1$$

as well as:

$$a_0 b_0^1 \rightarrow a_0 b_0^0 \equiv a_0 b_0^1 \ll a_0 b_0^0$$

as well as:

$$a_0 b_0^0 \rightarrow a_0 b_0^2 \equiv a_0 b_0^0 \gg^2 a_0 b_0^2$$

where the $i$ in $\gg^i$ denotes the offset of the shift.

### 3.1.2 Combining

With the establishing of the triviality of the relation between $a_0 b_0^1$ and $a_0 b_0^0$, let us explore the KA once more. Let there be three auxiliary variables $D_{0,1}$, $D_{1,0}$ and $D_{0,1}$ given by:

$$D_{0,1} = a_0 b_0^1$$

$$D_{1,0} = a_1 b_0^0$$

$$D_{0,1} = (a_0 + a_1)(b_0^1 + b_0^0)$$

As well as $D_{0,0}$ and $D_{1,1}$:

$$D_{0,0} = a_0 b_0^0$$

$$D_{1,1} = a_1 b_1^1$$

Then the linearized polynomial $C(x)$ can be written as:

$$C(x) = D_{1,1} x^2 + (D_{0,1} - D_{1,0} - D_{0,0}) x^1 + D_{0,0} x^0$$

To summarize the implementation: Let there be two $q$-degree-1 linearized polynomials $A(x)$ and $B(x)$ given by:

$$A(x) = a_1 x^1 + a_0 x^0$$

$$B(x) = b_1 x^1 + b_0 x^0$$

We have the following variables:

$$a_0, a_1, b_0, b_1$$
Compute the following auxiliary variables:

\[
\begin{align*}
\hat{b}_0^{[1]} & \quad \text{by } b_0 \gg 2 \quad (4) \\
\hat{b}_1^{[0]} & \quad \text{by } b_1 \gg \quad (5) \\
D_{0^1} & = a_0 b_0^{[1]} \quad \text{by } a_0 \cdot \hat{b}_0^{[1]} \quad (6) \\
D_{1^0} & = a_1 b_1^{[0]} \quad \text{by } a_1 \cdot \hat{b}_1^{[0]} \quad (7) \\
D_{0^0} & = a_0 b_0^{[0]} \quad \text{by } a_0 b_0^{[1]} \ll \quad (8) \\
D_{1^1} & = a_1 b_1^{[1]} \quad \text{by } a_1 b_1^{[0]} \gg \quad (9) \\
D_{0.1} & = (a_0 + a_1) (\hat{b}_0^{[1]} + \hat{b}_1^{[0]}) \quad \text{by } (a_0 + a_1) \cdot (\hat{b}_0^{[1]} + \hat{b}_1^{[0]}) \quad (10)
\end{align*}
\]

### 3.1.3 Concluding Karatsuba algorithm for \(q\)-degree 1 linearized polynomials

By using the above approach for acquiring the variables, the computation of \(C(x)\):

\[
C(x) = D_{1^1} x^{[2]} + (D_{0.1} - D_{0^1} - D_{1^0}) x^{[1]} + D_{0^0} x^{[0]}
\]

requires 4 shift operations, 3 multiplications and 4 additions.

### 3.2 Karatsuba algorithm for \(q\)-degree 2 linearized polynomial (e.g. \(a_2 x^q + a_1 x^{q^1} + a_0 x^{q^0}\))

Let there be two \(q\)-degree 2 linearized polynomials \(A(x)\) and \(B(x)\) given by:

\[
A(x) = a_2 x^{[2]} + a_1 x^{[1]} + a_0 x^{[0]}, \quad B(x) = b_2 x^{[2]} + b_1 x^{[1]} + b_0 x^{[0]}
\]

Then, following the earlier given definition of multiplication for linearized polynomials (2), the product \(C(x) = A(x)B(x)\) can be determined in the following manner:

\[
C(x) = a_2 b_2^{[2]} x^{[4]} + a_2 b_1^{[2]} x^{[3]} + a_2 b_0^{[2]} x^{[2]} \\
+ a_1 b_1^{[1]} x^{[3]} + a_1 b_0^{[1]} x^{[2]} + a_1 b_0^{[0]} x^{[1]} \\
+ a_0 b_1^{[0]} x^{[2]} + a_0 b_1^{[0]} x^{[1]} + a_0 b_0^{[0]} x^{[0]} \\
= a_2 b_2^{[2]} x^{[4]} + (a_2 b_1^{[2]} + a_1 b_2^{[1]}) x^{[3]} + (a_2 b_0^{[2]} + a_1 b_1^{[1]} + a_0 b_2^{[0]}) x^{[2]} \\
+ (a_1 b_0^{[1]} + a_0 b_1^{[0]}) x^{[1]} + a_0 b_0^{[0]} x^{[0]}
\]

The coefficients of \(x^{[1]}\), (part of) \(x^{[2]}\) and \(x^{[3]}\) in the resulting polynomial \(C(x)\) can be written as:

\[
\begin{align*}
(a_1 b_0^{[1]} + a_0 b_1^{[0]}) &= ((a_0 + a_1)(\hat{b}_0^{[1]} + \hat{b}_1^{[0]}) - a_0 \hat{b}_0^{[1]} - a_1 \hat{b}_1^{[0]}) \\
(a_2 b_0^{[2]} + a_0 b_2^{[0]}) &= ((a_0 + a_2)(\hat{b}_0^{[2]} + \hat{b}_2^{[0]}) - a_0 \hat{b}_0^{[2]} - a_2 \hat{b}_2^{[0]}) \\
(a_2 b_1^{[2]} + a_1 b_2^{[1]}) &= ((a_1 + a_2)(\hat{b}_1^{[2]} + \hat{b}_2^{[1]}) - a_1 \hat{b}_1^{[2]} - a_2 \hat{b}_2^{[1]})
\end{align*}
\]
Let there be the following auxiliary variables:

\[
D_{00} = a_0 b_0^{[0]}, D_{01} = a_0 b_1^{[1]}, D_{02} = a_0 b_2^{[2]}
\]
\[
D_{10} = a_1 b_1^{[0]}, D_{11} = a_1 b_1^{[1]}, D_{12} = a_1 b_1^{[2]}
\]
\[
D_{20} = a_2 b_2^{[0]}, D_{21} = a_2 b_2^{[1]}, D_{22} = a_2 b_2^{[2]}
\]
\[
D_{0,1} = (a_0 + a_1) (b_1^{[1]} + b_0^{[0]})
\]
\[
D_{0,2} = (a_0 + a_2) (b_2^{[2]} + b_0^{[0]})
\]
\[
D_{1,2} = (a_1 + a_2) (b_2^{[2]} + b_1^{[1]})
\]

Then:

\[
C(x) = D_{22} x^{[4]} + (D_{1,2} - D_{1,2} - D_{21}) x^{[3]} + (D_{0,2} - D_{0,2} - D_{2,1}) x^{[2]} +
(D_{0,1} - D_{0,1} - D_{1,0}) x^{[1]} + D_{0,0} x^{[0]}
\]

### 3.2.1 Combining

First, we need to pre-compute all individual \( b_i \) from \( D_{0,1}, D_{0,2} \) and \( D_{1,2} \), due to the earlier mentioned need for these single terms (section 3.1.3)

\[
b_0^{[0]}, b_0^{[0]}, b_1^{[0]}, b_1^{[1]}, b_2^{[0]}, b_2^{[1]}
\]

These can be computed by shifting \( b_i \) by an offset of 1, 2 or 3 in order to obtain \( b_1^{[0]}, b_1^{[1]} \) and \( b_1^{[2]} \) respectively. Coming down to a total of 6 shift operations.

With these \( b \) variables, \( D_{0,1}, D_{0,2} \) and \( D_{1,2} \) can be computed by means of 3 multiplications and 6 additions.

Next, we need one variation of \( D_{00}, D_{11} \) and \( D_{22} \). For sake of nothing let us take the lowest possible exponent out of the above listing for \( b \) and compute the following auxiliary variables by means of 3 multiplications:

\[
D_{01} = a_0 \cdot b_0^{[1]}
\]
\[
D_{10} = a_1 \cdot b_1^{[0]}
\]
\[
D_{20} = a_2 \cdot b_2^{[0]}
\]

With the following use of shifts, all other variations of \( D_{00}, D_{11} \) and \( D_{22} \) can be computed.

\[
D_{00} = D_{00} \ll
\]
\[
D_{02} = D_{01} \gg
\]
\[
D_{11} = D_{10} \gg
\]
\[
D_{12} = D_{10} \gg^2
\]
\[
D_{21} = D_{20} \gg
\]
\[
D_{22} = D_{20} \gg^2
\]
3.2.2 Concluding Karatsuba algorithm for $q$-degree-2 linearized polynomials

Thus, with a total of 6 multiplications and 12 shifts, all auxiliary variables are computed and can be filled in in the following equation:

$$C(x) = D_{22}x^4 + (D_{1,2} - D_{12} - D_{21})x^3 + (D_{0,2} - D_{02} - D_{20} + D_{11})x^2 + (D_{0,1} - D_{01} - D_{10})x^1 + D_{00}x^0$$

Resulting in the product $C(x)$ by means of 13 addition operations.

3.3 Karatsuba algorithm for $q$-degree-3 linearized polynomial (e.g. $a_3x^{q^3} + a_2x^{q^2} + a_1x^q + a_0x^0$)

In order to make the generalization towards $q$-degree-$n$ linearized polynomials, let us explore the KA for $q$-degree-3 linearized polynomials. Let there be two $q$-degree-3 linearized polynomials $A(x)$ and $B(x)$ given by:

$$A(x) = a_3x^{q^3} + a_2x^{q^2} + a_1x^q + a_0x^0,$$
$$B(x) = b_3x^{q^3} + b_2x^{q^2} + b_1x^q + b_0x^0$$

The worked out execution of the KA can be found in Appendix A.

3.4 Generalization

With use of the insight created in the previous sections, a generalization of the KA for arbitrary $q$-degree linearized polynomials (e.g. $\sum_{i=0}^{n} a_i x^{q^i}$) is formed. Let there be two $q$-degree-$n$ linearized polynomials (note: these polynomials have $n + 1$ terms) $A(x)$ and $B(x)$ given by:

$$A(x) = \sum_{i=0}^{n} a_i x^{q^i} \text{ and } B(x) = \sum_{i=0}^{n} b_i x^{q^i} \quad (11)$$

Let $C(x)$ be the product of $A(x)$ and $B(x)$. The following auxiliary values are needed in order to perform the KA:

$$D_{i,j} = a_i b_j^q \quad [\forall i = 0, 1, 2, ..., n, \text{ and } \forall j = 0, 1, 2, ..., n]$$

$$D_{i,j} = (a_i + a_j)(b_i + b_j) \quad [\forall k = 1, 2, ..., 2(n+1) - 3, \text{ and } \forall i, j \in \{0, 1, ..., n\} \text{ such that } i + j = k \text{ and } j > i \geq 0]$$

Then $C(x)$ of the form:

$$\sum_{i=0}^{2n} c_i x^{q^i}$$

Where:

$$c_0 = D_{00}$$
$$c_{2n} = D_{nn}$$
to profit from this algorithm, the earlier stated insights should be used for obtaining the auxiliary variables. When following the above algorithm by computing the auxiliary values blatantly, e.g. exponentiate without use of shifts, it would result in a complexity similar to the schoolbook approach. In order to profit from this algorithm, the earlier stated insights should be used for obtaining the auxiliary variables in such a way that computational cost is minimal. The following achieves just that:

\[
b_{ij} = b_{i}^{[j]} \quad \forall i = 0, 1, 2, ..., n, \text{ and } \forall j = 0, 1, 2, ..., n \text{ such that } i \neq j
\]

\[
D_{0i} = a_{0}b_{0i}
\]

\[
D_{q0} = D_{0i} \ll
\]

\[
D_{0i} = D_{0i} \gg^i \quad \forall i = 2, ..., n
\]

\[
D_{q0} = a_{0}b_{0i} \quad \forall i = 1, 2, ..., n
\]

\[
D_{ij} = D_{0i} \gg^j \quad \forall i = 1, 2, ..., n \text{ and } \forall j = 1, 2, ..., n
\]

\[
D_{i,j} = (a_{i} + a_{j})(b_{i} + b_{j}) \quad \forall k = 1, 2, ..., 2(n + 1) - 3,
\]

\[
\text{and } \forall i, j \in \{0, 1, ..., n\} \text{ such that } i + j = k \text{ and } j > i \geq 0
\]

The set of exponentiations of $b_{0i}$, referenced to as $b_{ij}$, is obtained in step 13, this set has a size of $(n + 1)^2 - (n + 1) = n^2 + n$. These single term elements are part of the computation of the set $D_{ij}$ and are therefore needed. To compute all elements of $b_{ij}$, $n^2 + n$ shift operations need to be performed.

Next, the variable $D_{0i}$ is computed instead of $D_{q0}$ because $b_{0i}$ is in $b_{ij}$ whereas $b_{q0}$ is not. This is the only element of $D_{ij}$ that is initially computed with $j = 1$ instead of $j = 0$. Thereby justifying step 15 and 16. All other $D_{ij}$ are computed by step 17 and 18. Combining step 14 - 18, there is a total of $n + 1$ multiplications (to obtain $D_{0i}$ and all $D_{q0}$) and $n(n + 1)$ shift operations.

Lastly, $D_{i,j}$ is computed. This set consists of $\frac{n^2 + n}{2}$ elements and is computed with the use of tuples of elements of $b_{ij}$ by means of $\frac{n^2 + n}{2}$ multiplications and $n^2 + n$ additions. All auxiliary variables are thus computed by means of:

\[
(n^2 + n) + n(n + 1) = n(2n + 2)
\]

\[\text{exponentiations}\]

\[n + 1 + \frac{n^2 + n}{2} = \frac{n^2 + 3n}{2} + 1\]

\[\text{multiplications}\]

\[n^2 + n\]

\[\text{additions}\]

3.4.1 Obtaining the auxiliary variables

In order to conclude the overall complexity, we need to determine the number of additions used in the computation of all $c_i$ (excluding $c_0$ and $c_{2n}$). Let us denote that number by $A$.

Looking at this equation (12) together with the worked out example in section 3.3, we see the following. We require (at least) 2 additions for computing $c_1$, $c_2$, $c_{2n-1}$ and $c_{2n-2}$, 5 additions for...
computing $c_3$, $c_4$, $c_{2n-3}$ and $c_{2n-4}$. This increase of 3 additions goes on. In order to simplify the determination of the number of additions, let us consider the set of tuples:

$$\theta = \{(c_1, c_2, c_{2n-2}, c_{2n-1}), (c_3, c_4, c_{2n-4}, c_{2n-3}), \ldots \}$$

Where $|\theta| = \frac{n}{2}$. Note that $c_n$ is the only $c_i$ that appears twice in this set. The elements of every tuple in $\theta$ share an equal number of additions in their computation (not yet taking the additional addition for even $i$ into account). Thus, we can compute the following:

$$A' = 4 \sum_{i=1}^{\frac{n}{2}} (3i - 1)$$

(21)

However, the computation of $c_n$ is counted twice by $i = \frac{n}{2}$ in the above equation, despite it only appearing once (see 3.3). The number of additions for $c_n$, we therefore have to subtract, is equal to:

$$A(c_n) = 3 \frac{n}{2} - 1$$

And the number of additional additions due to the even $i$ statement, we have to add, is equal to:

$$\frac{2n - 2}{2} = n - 1$$

Combining gives $A$ to be:

$$A = 4 \sum_{i=1}^{\frac{n}{2}} (3i - 1) - (3 \frac{n}{2} - 1) + (n - 1) = \frac{3n^2}{2} + \frac{n}{2}$$

(22)

Which lets us conclude the number of operations for the generalized KA for linearized polynomials of $q$-degree-$n$:

$$\begin{align*}
n(2n + 2) & \quad \text{exponentiations} \\
\frac{n^2 + 3n}{2} + 1 & \quad \text{multiplications} \\
\frac{5n^2}{2} + \frac{3n}{2} & \quad \text{additions}
\end{align*}$$

Table 1 below shows the number of operations for small prime $n$ for both KA and Schoolbook multiplication. Note that with increase of $n$ the number of multiplications needed for the KA gradually become less then for Schoolbook. Also note that the significant difference in number of exponentiations.
4 Karatsuba vs Schoolbook

Next, we consider the complexity of the newly found implementation of the KA for linearized polynomials and compare that to the complexity of the Schoolbook multiplication. We define the operations to have the following complexities:

addition in $\mathbb{F}_q$:
$$O(m \log_2(q))$$
multiplication in $\mathbb{F}_q$:
$$O(m \log_2(q) \log_2(m \log_2(q)))$$
exponentiation in $\mathbb{F}_q$:
$$O(1)$$

Note that the complexity of the exponentiation is constant since we assume the elements to be in their normal base representation. These complexities in combination with the number of executions established in 3.4.1 gives the following complexity for the KA for $q$-degree-$n$ linearized polynomials:

additions:
$$O((\frac{5n^2}{2} + \frac{3n}{2} - 1) \cdot m \log_2(q))$$
$$= O((\frac{5n^2}{2} + \frac{3n}{2} - 1)m \log_2(q))$$
multiplication:
$$O(\frac{n^2 + 3n}{2} + 1 \cdot \log_2(q) \log_2(m \log_2(q)))$$
$$= O(\frac{n^2 + 3n}{2} + 1(m \log_2(q) \log_2(m \log_2(q))))$$
shift:
$$O(2n + 2) \cdot O(1)$$
$$= O(2n^2 + 2n)$$

Combining into an overall complexity for the KA for $q$-degree-$n$ linearized polynomials:

$$O((\frac{5n^2}{2} + \frac{3n}{2} - 1)m \log_2(q)) +$$

$$O(\frac{n^2 + 3n}{2} + 1(m \log_2(q) \log_2(m \log_2(q))) + 2n^2 + 2n)$$

Table 1: comparison for KA and Schoolbook for small primes
And for Schoolbook:

additions:
\[(n - 1)^2 \cdot O(m(\log_2(q)))\]
\[= O((n - 1)^2 m(\log_2(q)))\]
multiplication:
\[n^2 \cdot O(\log_2(q) \log_2(m \log_2(q)))\]
\[= O(n^2 (m \log_2(q) \log_2(m \log_2(q))))\]
Exponentiations:
\[n^2 \cdot O(1)\]
\[= O(n^2)\]

Combining into an overall complexity for Schoolbook for \(q\)-degree-\(n\) linearized polynomials:
\[O \left((n - 1)^2 m(\log_2(q)) + n^2 (m \log_2(q) \log_2(m \log_2(q))) + n^2 \right)\] (24)

Since this research aims for an efficient alternative for MM in \(F_{q^n}^{\times n}\) we need to be able to represent matrices in the alternative field. We say \(m\) to be equal to \(n\) in all comparisons. This way all elements can be represented in both \(F_{q^n}^{\times n}\) and \(F_{q^m}\) and thus be compared. Note that not all elements of \(F_{q^n}^{\times n}\) require the degree of the linearized polynomial that represents it in \(F_{q^m}\) to be \(m\). Also note that by letting \(m = n\), quadratic asymptotic complexities with respect to \(n\) become cubic.
4.1 Exploring Karatsuba vs Schoolbook

Both complexities, (23) and (24), share the same asymptotic theoretical complexity of $O(n^2)$. In order to determine which multiplication algorithm is most suitable for competing with MM in $F_{q}^{m \times m}$, let us explore the quantified complexities of both KA and SB in $F_{q}^{m}$ for various values of $q$, and $n$. Appendix B contains an elaborate table that covers more values of $q$ and $n$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>KA</th>
<th>SB</th>
<th>Diff</th>
<th>%Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>50</td>
<td>14</td>
<td>-36</td>
<td>72.0</td>
</tr>
<tr>
<td>4</td>
<td>344</td>
<td>180</td>
<td>-164</td>
<td>-47.7</td>
</tr>
<tr>
<td>8</td>
<td>2600</td>
<td>1992</td>
<td>-608</td>
<td>-30.4</td>
</tr>
<tr>
<td>16</td>
<td>20860</td>
<td>20240</td>
<td>-620</td>
<td>-3.0</td>
</tr>
<tr>
<td>32</td>
<td>178328</td>
<td>190616</td>
<td>22288</td>
<td>11.6</td>
</tr>
<tr>
<td>64</td>
<td>1493504</td>
<td>1830976</td>
<td>337472</td>
<td>22.6</td>
</tr>
</tbody>
</table>

(a) $q = 2$

<table>
<thead>
<tr>
<th>$n$</th>
<th>KA</th>
<th>SB</th>
<th>Diff</th>
<th>%Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>260</td>
<td>108</td>
<td>-152</td>
<td>-58.5</td>
</tr>
<tr>
<td>4</td>
<td>1736</td>
<td>1184</td>
<td>-552</td>
<td>-31.8</td>
</tr>
<tr>
<td>8</td>
<td>12848</td>
<td>11572</td>
<td>-276</td>
<td>-2.1</td>
</tr>
<tr>
<td>16</td>
<td>101792</td>
<td>112960</td>
<td>10168</td>
<td>10.0</td>
</tr>
<tr>
<td>32</td>
<td>838592</td>
<td>1041536</td>
<td>102944</td>
<td>12.7</td>
</tr>
<tr>
<td>64</td>
<td>673392</td>
<td>946696</td>
<td>273304</td>
<td>39.9</td>
</tr>
</tbody>
</table>

(b) $q = 4$

<table>
<thead>
<tr>
<th>$n$</th>
<th>KA</th>
<th>SB</th>
<th>Diff</th>
<th>%Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>984</td>
<td>468</td>
<td>-516</td>
<td>-52.4</td>
</tr>
<tr>
<td>4</td>
<td>6288</td>
<td>4736</td>
<td>-552</td>
<td>-23.4</td>
</tr>
<tr>
<td>8</td>
<td>45088</td>
<td>45221</td>
<td>133</td>
<td>0.3</td>
</tr>
<tr>
<td>16</td>
<td>350813</td>
<td>416229</td>
<td>65416</td>
<td>18.6</td>
</tr>
<tr>
<td>32</td>
<td>2852712</td>
<td>3745412</td>
<td>892700</td>
<td>31.3</td>
</tr>
<tr>
<td>64</td>
<td>23733952</td>
<td>33200648</td>
<td>9466696</td>
<td>39.9</td>
</tr>
</tbody>
</table>

(c) $q = 16$

<table>
<thead>
<tr>
<th>$n$</th>
<th>KA</th>
<th>SB</th>
<th>Diff</th>
<th>%Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1183</td>
<td>570</td>
<td>-613</td>
<td>-51.8</td>
</tr>
<tr>
<td>4</td>
<td>7494</td>
<td>5723</td>
<td>-771</td>
<td>-23.6</td>
</tr>
<tr>
<td>8</td>
<td>53718</td>
<td>54449</td>
<td>731</td>
<td>1.3</td>
</tr>
<tr>
<td>16</td>
<td>416868</td>
<td>498379</td>
<td>81511</td>
<td>19.6</td>
</tr>
<tr>
<td>32</td>
<td>3384148</td>
<td>4372105</td>
<td>987957</td>
<td>32.1</td>
</tr>
<tr>
<td>64</td>
<td>28119298</td>
<td>39554879</td>
<td>11435581</td>
<td>40.7</td>
</tr>
</tbody>
</table>

(d) $q = 4093$

<table>
<thead>
<tr>
<th>$n$</th>
<th>KA</th>
<th>SB</th>
<th>Diff</th>
<th>%Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1387</td>
<td>675</td>
<td>-712</td>
<td>-51.3</td>
</tr>
<tr>
<td>4</td>
<td>8743</td>
<td>6735</td>
<td>-2008</td>
<td>-23.0</td>
</tr>
<tr>
<td>8</td>
<td>62478</td>
<td>63678</td>
<td>1200</td>
<td>1.9</td>
</tr>
<tr>
<td>16</td>
<td>483860</td>
<td>582130</td>
<td>98270</td>
<td>20.3</td>
</tr>
<tr>
<td>32</td>
<td>3922406</td>
<td>5211524</td>
<td>1289118</td>
<td>32.9</td>
</tr>
<tr>
<td>64</td>
<td>32556446</td>
<td>46904317</td>
<td>1435871</td>
<td>41.3</td>
</tr>
</tbody>
</table>

Table 2: quantified complexities for KA and Schoolbook for various $n$ and $q$ common in the field of cryptography

4.2 Concluding Karatsuba vs Schoolbook

We see that SB is faster for small $n$, regardless of $q$. We also see that with increasing $n$, KA becomes increasingly superior over SB. This increase is relatively bigger for larger $q$. We can conclude KA to be the preferred algorithm for multiplication in $F_{q}^{m}$ for the comparison with MM.

5 Karatsuba vs Matrix multiplication

As mentioned, a linearized polynomial and a matrix can be used to represent the same element:

$$A^{m \times m} \in F_{q}^{m \times m} \equiv f(x) \in F_{q^{m}}$$

where $\deg_q(f(x)) < m$

Before comparing computational complexities, let us define the scope of the comparison. Although most cryptographic algorithms, as of today, make use of MM, we are not considering KA for
linearized polynomials as a multiplication algorithm to plug in place of MM by means of transformations back and forth. We do therefore not explore transformations between the two discussed 'worlds'. The scope of this research is the comparison of:

\[ f_1 \cdot f_2, \text{ where } f \in F_{q^m} \text{ by means of } \text{KA} \]  \hspace{1cm} (25)

and

\[ A_1 \cdot A_2, \text{ where } A \in F_{q^m} \times m \text{ by means of naive matrix multiplication} \]  \hspace{1cm} (26)

## 5.1 Complexity of naive Matrix multiplication

In order to determine the complexity of naive MM, we define the following complexities:

- **addition in** \( F_q \)
  \[ \mathcal{O}(\log_2(q)) \]

- **multiplication in** \( F_q \)
  \[ \mathcal{O}(\log_2(q) \log_2(\log_2(q))) \]

And the following number of operations:

\[ n^3 \] \hspace{1cm} multiplications

\[ n^3 - n^2 \] \hspace{1cm} additions

Combining these definitions we can state the complexity of naive MM to be:

\[ \mathcal{O}(n^3 \log_2(q) + n^3 \log_2(q) \log_2(\log_2(q)) - n^2 \log_2(q)) \]  \hspace{1cm} (27)

With the earlier found complexity of KA for linearized polynomials (23), we now can compare.

## 5.2 Exploring Karatsuba vs Matrix multiplication

As can be seen in table 3, with increasing \( n \), the number of operations significantly start to differ. In order for sensible conclusions to be drawn, the operation complexity must to be taken into account.
Table 3: comparison for number of operations KA and MM for small primes

Table 4 shows the quantified complexity of both algorithms for 6 different primes \( q \) commonly used in the field of cryptography. Three of which are on the low side and the other three are on the high side. Note that for all values of \( q, n \) and \( m \), MM is superior over KA. Note that this superiority decreases with increase of \( n \). This decrease is faster for higher \( n \). However, even on the high end of the spectrum (\( q = 65521, n = 64 \)), KA is 35.8% slower than naive MM. Appendix B contains an elaborate table that covers more values of \( q \) and \( n \).

<table>
<thead>
<tr>
<th>n</th>
<th>KA in ( \mathbb{F}_3^n )</th>
<th>Naive in ( \mathbb{F}_3^m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6 MUL</td>
<td>12 MUL</td>
</tr>
<tr>
<td>3</td>
<td>10 ADD</td>
<td>26 ADD</td>
</tr>
<tr>
<td>5</td>
<td>21 EXP</td>
<td>69 EXP</td>
</tr>
<tr>
<td>7</td>
<td>36 MUL</td>
<td>132 MUL</td>
</tr>
<tr>
<td>11</td>
<td>78 MUL</td>
<td>318 MUL</td>
</tr>
</tbody>
</table>

Table 4: quantified complexities for KA and MM for various \( n \) and \( q \) common in the field of cryptography
6 Conclusion

In this research we implemented the Karatsuba Algorithm for same degree linearized polynomials. We analyzed the complexity and compared that to Schoolbook multiplication for same degree linearized polynomials. This comparison showed the Karatsuba algorithm to be faster for linearized polynomials of degree $n > 19$ independent of $q$ and for even lower degrees $n$ when $q$ gets larger. We also compared to naive matrix multiplication. Matrix multiplication is superior over the Karatsuba Algorithm in all cases.

7 Discussion

This research implemented single iteration Karatsuba as contender for matrix multiplication. Recursive Karatsuba (divide & conquer) fell out of the scope of this research. Although this paper did not conclude in an enticing alternative for matrix multiplication, perhaps the exploration of recursive Karatsuba could.

As mentioned, this research assumes $m$ to be equal to $n$. Using cases in which the ’dimension’ of the field of linearized polynomials $m$ is strictly smaller than matrix size $n$ has potency of being in favor of Karatsuba.

This research assumed and compared to naive matrix multiplication. There are known to be more efficient matrix multiplication algorithms, of which the Strassen algorithm is best known. Future research could incorporate other algorithms.

Lastly, this research did not consider real-world implementations of the algorithms, solely theory. Actual running time might show different results, especially when utilizing hardware in favorable ways.
References


A Karatsuba algorithm for $q$-degree 3 linearized polynomial

Then, following the earlier given definition of multiplication for linearized polynomials (2), the product $C(x) = A(x)B(x)$ can be determined in the following manner:

$$C(x) = a_3b_3^3x^6 + a_3b_2^3x^5 + a_3b_1^3x^4 + a_3b_0^3x^3$$
$$+ a_2b_3^2x^5 + a_2b_2^2x^4 + a_2b_1^2x^3 + a_2b_0^2x^2$$
$$+ a_1b_3^1x^4 + a_1b_2^1x^3 + a_1b_1^1x^2 + a_1b_0^1x^1$$
$$+ a_0b_3^0x^3 + a_0b_2^0x^2 + a_0b_1^0x^1 + a_0b_0^0x^0$$

$$= a_3b_3^3x^6 + (a_3b_2^3 + a_2b_3^2)x^5 + (a_3b_1^3 + a_2b_2^2 + a_1b_3^1)x^4$$
$$+ (a_3b_0^3 + a_2b_1^2 + a_1b_2^1 + a_0b_3^0)x^3 + (a_2b_0^2 + a_1b_1^1 + a_0b_2^0)x^2$$
$$+ (a_1b_0^1 + a_0b_1^0)x^1 + a_0b_0^0x^0$$

The coefficients of $x^1$, (part of) $x^2$, (part of) $x^3$, (part of) $x^4$, and $x^5$ in the resulting polynomial $C(x)$ can be written as:

$$(a_1b_0^1 + a_0b_1^0) = ((a_0 + a_1)(b_0^1 + b_1^0) - a_0b_1^1 - a_1b_0^1)$$
$$= (a_2b_0^2 + a_0b_2^0) = ((a_0 + a_2)(b_0^2 + b_2^0) - a_0b_2^0 - a_2b_0^2)$$
$$= (a_3b_0^3 + a_0b_3^0) = ((a_0 + a_3)(b_0^3 + b_3^0) - a_0b_3^0 - a_3b_0^3)$$
$$= (a_2b_1^2 + a_1b_2^1) = ((a_1 + a_2)(b_1^2 + b_2^1) - a_1b_2^1 - a_2b_1^1)$$
$$= (a_3b_1^3 + a_1b_3^1) = ((a_1 + a_3)(b_1^3 + b_3^1) - a_1b_3^1 - a_3b_1^3)$$
$$= (a_3b_2^2 + a_2b_3^2) = ((a_2 + a_3)(b_2^2 + b_3^2) - a_2b_3^2 - a_3b_2^2)$$
Let there be the following auxiliary variables:

\[ D_{00} = a_0 b_0^0, \quad D_{01} = a_0 b_0^1, \quad D_{02} = a_0 b_0^2, \quad D_{03} = a_0 b_0^3 \]
\[ D_{10} = a_1 b_1^0, \quad D_{11} = a_1 b_1^1, \quad D_{12} = a_1 b_1^2, \quad D_{13} = a_1 b_1^3 \]
\[ D_{20} = a_2 b_2^0, \quad D_{21} = a_2 b_2^1, \quad D_{22} = a_2 b_2^2, \quad \ldots \]
\[ D_{30} = a_3 b_3^0, \quad D_{31} = a_3 b_3^1, \quad D_{32} = a_3 b_3^2, \quad D_{33} = a_3 b_3^3 \]
\[ D_{0,1} = (a_0 + a_1)(b_0^1 + b_1^0) \]
\[ D_{0,2} = (a_0 + a_2)(b_0^2 + b_2^0) \]
\[ D_{0,3} = (a_0 + a_3)(b_0^3 + b_3^0) \]
\[ D_{1,2} = (a_1 + a_2)(b_1^2 + b_2^1) \]
\[ D_{1,3} = (a_1 + a_3)(b_1^3 + b_3^1) \]
\[ D_{2,3} = (a_2 + a_3)(b_2^3 + b_3^2) \]

Then:

\[ C(x) = D_{3,3} x^6 + 
\quad (D_{2,3} - D_{2,3} - D_{3,2}) x^5 + 
\quad (D_{1,3} - D_{1,3} - D_{3,1} + D_{2,2}) x^4 + 
\quad ((D_{0,3} - D_{0,3} - D_{3,0}) + (D_{1,2} - D_{1,2} - D_{2,1})) x^3 + 
\quad (D_{0,2} - D_{2,0} - D_{0,2} + D_{1,1}) x^2 + 
\quad (D_{0,1} - D_{0,1} - D_{1,0}) x^1 + 
\quad D_{0,0} x^0 \]
### B Complexities

The following table contains the complexity for different values of $q$ and $n$. The KA and SB columns display the complexity of Karatsuba and Schoolbook respectively. The SB-KA and the % column that follows that, show the difference between SB and KA with respect to KA, absolute and percentage-wise. The MM column displays the complexity of naive matrix multiplication. The MM-KA and the % column that follows that show the difference between MM and KA with respect to KA, again, absolute and percentage-wise.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$n$</th>
<th>KA</th>
<th>SB</th>
<th>SB-KA</th>
<th>%</th>
<th>MM</th>
<th>MM-KA</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>14</td>
<td>-36</td>
<td>-72.0</td>
<td>4</td>
<td>-46</td>
<td>-92.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>344</td>
<td>180</td>
<td>-164</td>
<td>-47.7</td>
<td>48</td>
<td>-296</td>
<td>-86.0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2600</td>
<td>1992</td>
<td>-608</td>
<td>-23.4</td>
<td>448</td>
<td>-2152</td>
<td>-82.8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>20960</td>
<td>20240</td>
<td>-720</td>
<td>-3.4</td>
<td>3840</td>
<td>-17120</td>
<td>-81.7</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>175382</td>
<td>195616</td>
<td>20288</td>
<td>11.6</td>
<td>31744</td>
<td>-143584</td>
<td>-81.9</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>1493564</td>
<td>1830976</td>
<td>337472</td>
<td>22.6</td>
<td>238048</td>
<td>-1235456</td>
<td>-82.7</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>12813440</td>
<td>16769960</td>
<td>3947520</td>
<td>30.8</td>
<td>2080708</td>
<td>-10726272</td>
<td>-83.8</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>110870272</td>
<td>150929664</td>
<td>40959392</td>
<td>37.1</td>
<td>16711680</td>
<td>-93588592</td>
<td>-84.8</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>943896176</td>
<td>134195648</td>
<td>397929472</td>
<td>42.2</td>
<td>139355584</td>
<td>-81003092</td>
<td>-85.8</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>8672291056</td>
<td>12698621885</td>
<td>4026330833</td>
<td>46.4</td>
<td>1149924288</td>
<td>-7522867674</td>
<td>-86.7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>112</td>
<td>40</td>
<td>-72</td>
<td>-64.3</td>
<td>24</td>
<td>-88</td>
<td>-78.6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>768</td>
<td>472</td>
<td>-296</td>
<td>-38.5</td>
<td>224</td>
<td>-544</td>
<td>-70.8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>5776</td>
<td>4944</td>
<td>-832</td>
<td>-14.4</td>
<td>1920</td>
<td>-3856</td>
<td>-66.8</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>46272</td>
<td>4816</td>
<td>2144</td>
<td>4.6</td>
<td>15872</td>
<td>-30400</td>
<td>-65.7</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>384148</td>
<td>455744</td>
<td>71296</td>
<td>18.5</td>
<td>129024</td>
<td>-556424</td>
<td>-66.4</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>3253248</td>
<td>4182144</td>
<td>928896</td>
<td>28.6</td>
<td>1040384</td>
<td>-2212864</td>
<td>-68.0</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>27740416</td>
<td>37699840</td>
<td>9959424</td>
<td>35.9</td>
<td>8355840</td>
<td>-19384576</td>
<td>-69.9</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>236983296</td>
<td>335348224</td>
<td>98364928</td>
<td>41.5</td>
<td>66977792</td>
<td>-170005504</td>
<td>-71.7</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>184567030801</td>
<td>27098190651</td>
<td>9201487650</td>
<td>49.7</td>
<td>4601893760</td>
<td>-13894809241</td>
<td>-75.5</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>260</td>
<td>108</td>
<td>-152</td>
<td>-58.5</td>
<td>80</td>
<td>-180</td>
<td>-69.2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1736</td>
<td>1182</td>
<td>-976</td>
<td>-7.6</td>
<td>5888</td>
<td>-6960</td>
<td>-54.2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>12848</td>
<td>11872</td>
<td>-976</td>
<td>-7.6</td>
<td>5888</td>
<td>-6960</td>
<td>-54.2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>101792</td>
<td>112960</td>
<td>11168</td>
<td>11.0</td>
<td>48128</td>
<td>-53664</td>
<td>-52.7</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>838592</td>
<td>1014536</td>
<td>202944</td>
<td>24.2</td>
<td>389120</td>
<td>-449472</td>
<td>-53.6</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6474296</td>
<td>9408768</td>
<td>2361472</td>
<td>33.5</td>
<td>3129344</td>
<td>-3917952</td>
<td>-55.6</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>50740928</td>
<td>83771904</td>
<td>24039976</td>
<td>40.2</td>
<td>25100288</td>
<td>-34646640</td>
<td>-58.0</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>507783680</td>
<td>737739776</td>
<td>229956096</td>
<td>45.3</td>
<td>201064448</td>
<td>-306719232</td>
<td>-60.4</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>4314389504</td>
<td>640617984</td>
<td>2126228480</td>
<td>49.3</td>
<td>1609564160</td>
<td>-2704825344</td>
<td>-62.7</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>39299836499</td>
<td>59999373367</td>
<td>206095268688</td>
<td>52.7</td>
<td>13807877888</td>
<td>-25491968611</td>
<td>-64.9</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>287</td>
<td>145</td>
<td>-192</td>
<td>-57.0</td>
<td>111</td>
<td>-226</td>
<td>-61.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2323</td>
<td>1560</td>
<td>-672</td>
<td>-30.1</td>
<td>909</td>
<td>-1263</td>
<td>-56.6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>14629</td>
<td>17471</td>
<td>-958</td>
<td>5.8</td>
<td>8575</td>
<td>-8354</td>
<td>-50.8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>120687</td>
<td>141018</td>
<td>16421</td>
<td>12.7</td>
<td>65872</td>
<td>-63815</td>
<td>-49.2</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>1005584</td>
<td>1339855</td>
<td>274271</td>
<td>25.7</td>
<td>532050</td>
<td>-533534</td>
<td>-50.1</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>8936354</td>
<td>12053092</td>
<td>3116738</td>
<td>34.9</td>
<td>4276695</td>
<td>-4659659</td>
<td>-52.1</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>75625365</td>
<td>106958498</td>
<td>31333133</td>
<td>41.4</td>
<td>34294731</td>
<td>-41330634</td>
<td>-54.7</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>641861499</td>
<td>939365625</td>
<td>297504126</td>
<td>46.4</td>
<td>274682526</td>
<td>-306718973</td>
<td>-57.2</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>5446677244</td>
<td>8182193649</td>
<td>2735516405</td>
<td>50.2</td>
<td>2198758926</td>
<td>-3247918318</td>
<td>-59.6</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>49565899471</td>
<td>76071955705</td>
<td>26551462324</td>
<td>53.5</td>
<td>18861783586</td>
<td>-3069525885</td>
<td>-61.9</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>2</td>
<td>512</td>
<td>230</td>
<td>-282</td>
<td>-55.1</td>
<td>184</td>
<td>-328</td>
<td>-61.4</td>
</tr>
<tr>
<td>4</td>
<td>3340</td>
<td>2416</td>
<td>-924</td>
<td>-27.7</td>
<td>1590</td>
<td>-1790</td>
<td>-52.4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>24365</td>
<td>23575</td>
<td>-790</td>
<td>-3.2</td>
<td>13167</td>
<td>-11198</td>
<td>-46.0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>191214</td>
<td>220213</td>
<td>28999</td>
<td>15.2</td>
<td>107132</td>
<td>-84082</td>
<td>-44.0</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>1564584</td>
<td>2065330</td>
<td>438746</td>
<td>28.0</td>
<td>864213</td>
<td>-700317</td>
<td>-44.8</td>
<td></td>
</tr>
<tr>
<td>65521</td>
<td>2</td>
<td>1387</td>
<td>675</td>
<td>-712</td>
<td>-51.3</td>
<td>575</td>
<td>-812</td>
<td>-58.5</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>4</td>
<td>8743</td>
<td>6735</td>
<td>-2008</td>
<td>-23.0</td>
<td>4863</td>
<td>-3880</td>
<td>-44.4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>62418</td>
<td>63678</td>
<td>1200</td>
<td>1.3</td>
<td>39934</td>
<td>-22544</td>
<td>-36.1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>483860</td>
<td>582130</td>
<td>98270</td>
<td>20.3</td>
<td>323575</td>
<td>-169285</td>
<td>-33.1</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>3922406</td>
<td>5211524</td>
<td>1289118</td>
<td>32.9</td>
<td>2604986</td>
<td>-1317420</td>
<td>-33.6</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>32556446</td>
<td>46010317</td>
<td>13453871</td>
<td>41.3</td>
<td>20905427</td>
<td>-11051019</td>
<td>-35.8</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>273203450</td>
<td>402138028</td>
<td>125934578</td>
<td>47.2</td>
<td>167505559</td>
<td>-105697891</td>
<td>-38.7</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>2302277782</td>
<td>3487553430</td>
<td>1186275648</td>
<td>51.5</td>
<td>1341093029</td>
<td>-961184753</td>
<td>-41.7</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>19415632121</td>
<td>30055968481</td>
<td>10640336360</td>
<td>54.8</td>
<td>10732938452</td>
<td>-8682693669</td>
<td>-44.7</td>
<td></td>
</tr>
<tr>
<td>1048</td>
<td>175658002840</td>
<td>276820659349</td>
<td>101162656509</td>
<td>57.6</td>
<td>92061785874</td>
<td>-83596216966</td>
<td>-47.6</td>
<td></td>
</tr>
</tbody>
</table>