ARTICLE IN PRESS YJSCO:1812

Journal of Symbolic Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

# On the complexity of integer matrix multiplication

David Harvey [a], Joris van der Hoeven [b]

[a] School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia
[b] CNRS, LIX, École polytechnique, 91128 Palaiseau Cedex, France

ARTICLE INFO

*Article history:*
Received 12 November 2014
Accepted 9 October 2016
Available online xxxx

*MSC:*
68W30
68Q17
68W40

*Keywords:*
Matrix multiplication
Complexity
Algorithm
FFT
Bluestein reduction

ABSTRACT

Let $\mathsf{M}(n)$ denote the bit complexity of multiplying $n$-bit integers, let $\omega \in (2, 3]$ be an exponent for matrix multiplication, and let $\lg^* n$ be the iterated logarithm. Assuming that $\log d = O(n)$ and that $\mathsf{M}(n)/(n \log n)$ is increasing, we prove that $d \times d$ matrices with $n$-bit integer entries may be multiplied in

$$O(d^2 \mathsf{M}(n) + d^\omega n\, 2^{O(\lg^* n - \lg^* d)} \mathsf{M}(\lg d)/\lg d)$$

bit operations. In particular, if $n$ is large compared to $d$, say $d = O(\log n)$, then the complexity is only $O(d^2 \mathsf{M}(n))$.



## 1. Introduction

In this paper we study the complexity of multiplying $d \times d$ matrices whose entries are integers with at most $n$ bits. We are particularly interested in the case that $n$ is very large compared to $d$, say $d = O(\log n)$. All complexity bounds refer to deterministic bit complexity, in the sense of the multi-tape Turing model (Papadimitriou, 1994).

Matrices with large integer coefficients appear naturally in several areas. One first application is to the efficient high precision evaluation of so-called holonomic functions (such as exp, log, sin,

*E-mail addresses:* d.harvey@unsw.edu.au (D. Harvey), vdhoeven@lix.polytechnique.fr (J. van der Hoeven).

https://doi.org/10.1016/j.jsc.2017.11.001
0747-7171/© 2017 Elsevier Ltd. All rights reserved.

Please cite this article in press as: Harvey, D., van der Hoeven, J. On the complexity of integer matrix multiplication. J. Symb. Comput. (2017), https://doi.org/10.1016/j.jsc.2017.11.001

Bessel functions, and hypergeometric functions) using a divide and conquer technique (Chudnovsky and Chudnovsky, 1990; Haible and Papanikolaou, 1997; van der Hoeven, 1999, 2001, 2007). Another application concerns recent algorithms for computing the $L$-series of algebraic varieties (Harvey, 2014, 2015; Harvey and Sutherland, 2014, 2016; Harvey et al., 2016a). The practical running time in these applications is dominated by the multiplication of matrices with large integer entries, and it is vital to have a highly efficient implementation of this fundamental operation. Typical parameters for these applications are $n$ around $10^8$ bits, and $d$ around 10.

In this paper, we focus mainly on theoretical bounds. We write $\mathsf{M}_d(n)$ for the cost of multiplying $d \times d$ matrices with $n$-bit integer entries, and $\mathsf{M}(n) := \mathsf{M}_1(n)$ for the cost of multiplying $n$-bit integers. We will also write $\mathsf{M}_{R,d}(n)$ for the algebraic complexity of multiplying $d \times d$ matrices whose entries are polynomials of degree $< n$ over an abstract effective ring $R$, and $\mathsf{M}_R(n) := \mathsf{M}_{R,1}(n)$.

Schönhage and Strassen (1971) used fast Fourier transforms (FFTs) to prove that $\mathsf{M}(n) = O(n \log n \log \log n)$ for large $n$. Fürer (2009) improved this to $\mathsf{M}(n) = O(n \log n\, 2^{O(\lg^* n)})$ where $\lg^*$ is the iterated logarithm, i.e.,

$$\begin{aligned} \lg n &:= \lceil \log_2 n \rceil, \\ \lg^* n &:= \min\{k \in \mathbb{N} : \lg^{\circ k} n \leqslant 1\}, \\ \lg^{\circ k} &:= \underbrace{\lg \circ \cdots \circ \lg}_{k \times}, \end{aligned}$$

and this was recently sharpened to $\mathsf{M}(n) = O(n \log n\, 8^{\lg^* n})$ (Harvey et al., 2016b). The best currently known bound (Cantor and Kaltofen, 1991) for $\mathsf{M}_R(n)$ is $\mathsf{M}_R(n) = O(n \log n \log \log n)$; if $R$ is a ring of finite characteristic this may be improved to $\mathsf{M}_R(n) = O(n \log n\, 8^{\lg^* n})$ (Harvey et al., 2017).

The algebraic complexity of $d \times d$ matrix multiplication is usually assumed to be of the form $O(d^\omega)$, where $\omega$ is a so-called exponent of matrix multiplication (von zur Gathen and Gerhard, 2003, Ch. 12). Classical matrix multiplication yields $\omega = 3$, and Strassen's algorithm (Strassen, 1969) achieves $\omega = \log 7 / \log 2 \approx 2.807$. The best currently known exponent $\omega < 2.3728639$ was found by Le Gall (Le Gall, 2014; Coppersmith and Winograd, 1987).

When working over the integers and taking into account the growth of coefficients, the general bound for matrix multiplication specialises to

$$\mathsf{M}_d(n) = O(d^\omega \mathsf{M}(n + \lg d)).$$

Throughout this paper we will enforce the very mild restriction that $\log d = O(n)$. Under this assumption the above bound simplifies to

$$\mathsf{M}_d(n) = O(d^\omega \mathsf{M}(n)).$$

The main result of this paper is the following improvement.

**Theorem 1.** *Assume that* $\mathsf{M}(n)/(n \log n)$ *is increasing. Let* $C > 1$ *be a constant. Then*

$$\mathsf{M}_d(n) = O(d^2 \mathsf{M}(n) + d^\omega n\, 2^{O(\lg^* n - \lg^* d)} \mathsf{M}(\lg d)/\lg d), \tag{1}$$

*uniformly for* $n \geqslant 2$ *and* $d \geqslant 1$, *under the condition that* $\lg d \leqslant Cn$.

In particular, if $n$ is large compared to $d$, say $d = O(\log n)$, then (1) simplifies to

$$\mathsf{M}_d(n) = O(d^2 \mathsf{M}(n)). \tag{2}$$

This bound is essentially optimal (up to constant factors), in the sense that we cannot expect to do better for $d = 1$, and the bound grows proportionally to the input and output size as a function of $d$.

The new algorithm has its roots in studies of analogous problems in the algebraic complexity setting. When working over an arbitrary effective ring $R$, a classical technique for multiplying polynomial matrices is to use an evaluation-interpolation scheme. There are many different evaluation-interpolation strategies (van der Hoeven, 2010, Sections 2.1–2.3) such as Karatsuba, Toom–Cook, FFT,