Matrix and polynomial computation

9 septembre 2020

Overview

Before

≈ 55 years

Standard bases

ANNALS OF MATHEMATICS
Vol. 79, No. 2, March, 1964
Printed in Japan

RESOLUTION OF SINGULARITIES OF AN ALGEBRAIC VARIETY OVER A FIELD OF CHARACTERISTIC ZERO: II

BY HEISUKE HIRONAKA

(*Part* I *appeared in the preceding issue of this Journal*)

CHAPTER III. EFFECTS OF PERMISSIBLE MONOIDAL TRANSFORMATIONS ON SINGULARITIES.

≈ 55 years

Gröbner bases

Bruno Buchberger's PhD thesis 1965: "An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal"

1965-00-00-A

Ein Algorithmus zum Auffinden der Basisele-
mente des Restklassenringes nach einem null-
dimensionalen Polynomideal.

D i s s e r t a t i o n

zur Erlangung des Doktorgrades der philoso-
phischen Fakultät an der Leopold-Franzens-
Universität Innsbruck.

≈ 50 years

Numer. Math. 13, 354—356 (1969)

Gaussian Elimination is not Optimal

VOLKER STRASSEN*

Received December 12, 1968

1. Below we will give an algorithm which computes the coefficients of the product of two square matrices A and B of order n from the coefficients of A and B with less than $4.7 \cdot n^{\log 7}$ arithmetical operations (all logarithms in this

≈ 50 years

Actes, Congrès intern. Math., 1970. Tome 3, p. 269 à 274.

THE ANALYSIS OF ALGORITHMS

by Donald E. KNUTH

Some general aspects of algorithmic analysis are illustrated by discussing Euclid's algorithm. Euclid's method is extended in such a way that the gcd of two n digit numbers can be found in $0(n(\log n)^5 (\log \log n))$ steps as $n \to \infty$.

≈ 40 years

Math. Ann. 261, 515–534 (1982)

Mathematische Annalen

Factoring Polynomials with Rational Coefficients

A. K. Lenstra[1], H. W. Lenstra, Jr.[2], and L. Lovász[3]

1 Mathematisch Centrum, Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands
2 Mathematisch Instituut, Universiteit van Amsterdam, Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands
3 Bolyai Institute, A. József University, Aradi vértanúk tere 1, H-6720 Szeged, Hungary

In this paper we present a polynomial-time algorithm to solve the following problem: given a non-zero polynomial $f \in \mathbb{Q}[X]$ in one variable with rational coefficients, find the decomposition of f into irreducible factors in $\mathbb{Q}[X]$. It is well

≈ 30 years

Acta Informatica vol. 28, nr. 7, pp. 693–701 (1991)

ON FAST MULTIPLICATION OF POLYNOMIALS OVER ARBITRARY ALGEBRAS

David G. Cantor and Erich Kaltofen*

1. Introduction. In this paper we generalize the well-known Schönhage-Strassen algorithm for multiplying large integers to an algorithm for multiplying polynomials with coefficients from an arbitrary, not necessarily commutative, not necessarily associative, algebra $\mathcal{A}$. Our main result is an algorithm to multiply polynomials of degree $< n$ in $O(n \log n)$ algebra multiplications and $O(n \log n \operatorname{loglog} n)$ algebra

≈ 25 years

MATHEMATICS OF COMPUTATION
VOLUME 62, NUMBER 205
JANUARY 1994, PAGES 333–350

SOLVING HOMOGENEOUS LINEAR EQUATIONS OVER $GF(2)$ VIA BLOCK WIEDEMANN ALGORITHM

DON COPPERSMITH

ABSTRACT. We propose a method of solving large sparse systems of homogeneous linear equations over $GF(2)$, the field with two elements. We modify an algorithm due to Wiedemann. A block version of the algorithm allows us to perform 32 matrix-vector operations for the cost of one. The resulting algorithm is competitive with structured Gaussian elimination in terms of time and has much lower space requirements. It may be useful in the last stage of integer factorization.

≈ 25 years

SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 804–823, July 1994

A UNIFORM APPROACH FOR THE FAST COMPUTATION OF MATRIX-TYPE PADÉ APPROXIMANTS *

BERNHARD BECKERMANN† AND GEORGE LABAHN‡

Abstract. Recently, a uniform approach was given by B. Beckermann and G. Labahn [*Numer. Algorithms*, 3 (1992), pp. 45–54] for different concepts of matrix-type Padé approximants, such as descriptions of vector and matrix Padé approximants along with generalizations of simultaneous and

≈ 20 years

A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)

Jean-Charles Faugère
Ver 1.2
SPACES/LIP6/CNRS/Université Paris VI/INRIA
case 168, 4 pl. Jussieu, F-75252 Paris Cedex 05
E-mail: jcf@calfor.lip6.fr

ABSTRACT

This paper introduces a new efficient algorithm for computing Gröbner bases. We replace the Buchberger criteria by an optimal criteria. We give a proof that the resulting algorithm (called F_5) generates

dependence is $\sum \lambda t f = 0$ or by grouping terms: $\sum_{i=1}^{m} g_i f_i = 0$. In other words, $(g_1, \ldots, g_m)$ is a syzygy.

Several papers investigate those issues: Buchberger [4] proposes

≈ 20 years

High-order lifting and integrality certification

Arne Storjohann

School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

Abstract

Reductions to polynomial matrix multiplication are given for some classical problems involving a nonsingular input matrix over the ring of univariate polynomials with coefficients from a field. High-order lifting is used to compute the determinant,

≈ 10 years

SIAM J. COMPUT.
Vol. 40, No. 6, pp. 1767–1802

FAST POLYNOMIAL FACTORIZATION AND MODULAR COMPOSITION*

KIRAN S. KEDLAYA† AND CHRISTOPHER UMANS‡

Abstract. We obtain randomized algorithms for factoring degree n univariate polynomials over $\mathbb{F}_q$ requiring $O(n^{1.5+o(1)} \log^{1+o(1)} q + n^{1+o(1)} \log^{2+o(1)} q)$ bit operations. When $\log q < n$, this is asymptotically faster than the best previous algorithms [J. von zur Gathen and V. Shoup, *Comput. Complexity*, 2 (1992), pp. 187–224; E. Kaltofen and V. Shoup, *Math. Comp.*, 67 (1998), pp. 1179–1197]; for $\log q > n$, it matches the asymptotic running time of the best known algorithms. The

≈ **5 years**

Powers of Tensors and Fast Matrix Multiplication

François Le Gall
Department of Computer Science
Graduate School of Information Science and Technology
The University of Tokyo
legall@is.s.u-tokyo.ac.jp

30 Jan 2014

Abstract

This paper presents a method to analyze the powers of a given trilinear form (a special kind of algebraic constructions also called a tensor) and obtain upper bounds on the asymptotic complexity of matrix multiplication. Compared with existing approaches, this method is based on convex optimization, and thus has polynomial-time complexity. As an application, we use this method to study powers of the construction given by Coppersmith and Winograd [Journal of Symbolic Computation, 1990] and obtain the upper bound $\omega < 2.3728639$ on the exponent of square matrix multiplication, which slightly improves the best known upper bound.

≈ Yesterday

Integer multiplication in time $O(n \log n)$

David Harvey and Joris van der Hoeven

Abstract. We present an algorithm that computes the product of two n-bit integers in $O(n \log n)$ bit operations.

1. Introduction

Let $\mathsf{M}(n)$ denote the time required to multiply two n-bit integers. We work in the multitape Turing model, in which the time complexity of an algorithm refers to the number of steps performed by a deterministic Turing machine with a fixed, finite number of linear tapes [34]. The main results of this paper also hold in the Boolean circuit model [40, Sec. 9.3], with essentially the same proofs. We write

Starting point

$A \in \mathsf{K}^{n \times n}$

$$\begin{bmatrix} 1 & 0 & 3 & 3 & 5 & 6 & 8 & 8 \\ 3 & 4 & 4 & 9 & 9 & 2 & 5 & 2 \\ 8 & 10 & 9 & 3 & 6 & 5 & 6 & 5 \\ 7 & 4 & 9 & 3 & 0 & 1 & 3 & 7 \\ 2 & 9 & 7 & 5 & 3 & 6 & 8 & 2 \\ 3 & 5 & 8 & 6 & 2 & 5 & 4 & 4 \\ 5 & 2 & 1 & 1 & 7 & 10 & 6 & 5 \\ 0 & 10 & 0 & 4 & 1 & 7 & 4 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 3 & 3 & 5 & 6 & 8 & 8 \\ 0 & 4 & 6 & 0 & 5 & 6 & 3 & 0 \\ 0 & 0 & 3 & 1 & 3 & 8 & 6 & 7 \\ 0 & 0 & 0 & 10 & 0 & 1 & 2 & 4 \\ 0 & 0 & 0 & 0 & 8 & 6 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 8 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

Size $O(n^2)$ unchanged after triangularization

Model : algebraic computation, finite field, (floating point)

$A \in \mathbb{Z}[x, y, z]$, example 8×8, 2×2 submatrix:

$$\begin{bmatrix} 50\,x^2y - 22\,y^3 - 7\,y^2z + 67\,yz - 53 & -xy^2 - 43\,xyz - 39\,z^3 + 34\,yz + 56\,z^2 - 68 \\ 7\,x^2y + 97\,xyz - 78\,xy + 53\,xz - 50\,yz - 83\,y & -2\,x^2z - 26\,xz^2 - 72\,y^3 + 67\,z^3 - 69\,z^2 + 51 \end{bmatrix}$$

Some coefficient after triangularization:

$$9483760\,x^{14}y + 33303556\,x^{13}y^2 - 13474656\,x^{13}yz - 52220480\,x^{13}z^2 - 514110020\,x^{12}y^3 + \ldots$$

Increased bit size and polynomial degrees: ex from 5 ko to 360 ko

Models

Algorithms: Algebraic Random Access Machine (RAM)

- CPU, I/O medium, address and data memory, test and branching, etc.

Models

Algorithms: Algebraic Random Access Machine (RAM)

- CPU, I/O medium, address and data memory, test and branching, etc.

Arithmetic circuits

Straight-Line Program (SLP)

$P = (I, O, S, C)$

- Input and Output variables
- A set of Scalars
- Computation sequence of length l: $v_i \leftarrow v_j \circ v_k$ for $\circ \in \{+, -, \times, /\}$

Complexity bounds

Algebraic model over an abstract field (or ring)

- Algebraic operations cost 1

Complexity bounds

Algebraic model over an abstract field (or ring)

- Algebraic operations cost 1

Bit complexity

Integers a, b with at most n bits (lengths, say $\approx \log |a|$ and $\log |b|$)

- Product in $O(n^2)$ bit operations
- $O(n \log n)$ bit operations [Harvey & van der Hoeven 2019]

Polynomial multiplication

Univariate polynomial $p, q \in \mathsf{K}[x]$ with degree at most n

- Product in $O(n^2)$ arithmetic operations
- Over arbitrary algebras : $O(n \log n \log \log n)$ [Cantor & Kaltofen 1991]

Soft 0 notation for hidding polylogarithmic factors

Ex: $O(n(\log n)^i(\log\log n)^j) = \tilde{O}(n)$

NC: *decision problems decidable in* ***polylogarithmic time on a parallel computer***

The polynomial gcd problem in $K[x]$ is in NC

Hint: via structured linear algebra (see later in the course)

Open problem

Is the integer problem in NC **or P-complete?**

Hint(?): The iterated mod problem [Karloff & Ruzzo 1989]

Matrices

Algebraic complexity model

$A, B \in \mathsf{K}^{n \times n}$, compute $A \times B$?

Feasible exponent: there exists an algorithm using $O(n^\theta)$ arithmetic operations

Exponent of matrix multiplication: $\omega = \inf \{ \theta \mid \theta \text{ feasible} \}$

By abuse we will use the notation ω for feasible exponents

[Vassilevska Williams 2019] *Limits on All Known (and Some Unknown) Approaches to Matrix Multiplication*

[Le Gall & Urrutia 2018] *Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor*

[Le Gall 2014]

$A, B \in \mathsf{K}^{n \times n}$, compute $A \times B$?

Table 1: History of the main improvements on the exponent of square matrix multiplication

Upper bound	Year	Reference
$\omega \leq 3$		
$\omega < 2.81$	1969	Strassen [11]
$\omega < 2.79$	1979	Pan [6]
$\omega < 2.78$	1979	Bini et al. [1]
$\omega < 2.55$	1981	Schönhage [9]
$\omega < 2.53$	1981	Pan [7]
$\omega < 2.52$	1982	Romani [8]
$\omega < 2.50$	1982	Coppersmith and Winograd [2]
$\omega < 2.48$	1986	Strassen [12]
$\omega < 2.376$	1987	Coppersmith and Winograd [3]
$\omega < 2.373$	2010	Stothers [10] (see also [4])
$\omega < 2.3729$	2012	Vassilevska Williams [13]
$\omega < 2.3728639$	2014	Le Gall [5]

Dense linear algebra

$$A = \begin{bmatrix} 42 & 10 & 24 & 12 & 18 & 10 & 26 & 15 & 19 & 15 \\ 8 & 41 & 37 & 32 & 39 & 31 & 40 & 9 & 17 & 35 \\ 38 & 27 & 7 & 4 & 11 & 30 & 49 & 6 & 27 & 47 \\ 6 & 38 & 16 & 29 & 20 & 18 & 41 & 1 & 8 & 51 \\ 10 & 12 & 50 & 12 & 16 & 12 & 33 & 5 & 51 & 47 \\ 5 & 46 & 33 & 27 & 42 & 18 & 49 & 12 & 19 & 1 \\ 6 & 17 & 44 & 38 & 33 & 23 & 28 & 15 & 26 & 2 \\ 15 & 18 & 45 & 36 & 8 & 31 & 43 & 8 & 19 & 16 \\ 20 & 18 & 5 & 4 & 23 & 19 & 3 & 45 & 35 & 36 \\ 3 & 42 & 24 & 29 & 39 & 34 & 4 & 7 & 44 & 46 \end{bmatrix}$$

Triangularization

$$\begin{bmatrix}
42 & 10 & 24 & 12 & 18 & 10 & 26 & 15 & 19 & 15 \\
0 & 29 & 40 & 7 & 28 & 19 & 30 & 44 & 26 & 17 \\
0 & 0 & 11 & 9 & 43 & 42 & 17 & 51 & 24 & 33 \\
0 & 0 & 0 & 9 & 23 & 8 & 7 & 7 & 35 & 46 \\
0 & 0 & 0 & 0 & 47 & 30 & 1 & 31 & 41 & 46 \\
0 & 0 & 0 & 0 & 0 & 15 & 27 & 12 & 24 & 37 \\
0 & 0 & 0 & 0 & 0 & 0 & 7 & 49 & 7 & 37 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 45 & 40 & 41 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 17 & 29 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 44
\end{bmatrix}$$

Determinant

$$\begin{bmatrix}
\mathbf{42} & 10 & 24 & 12 & 18 & 10 & 26 & 15 & 19 & 15 \\
0 & \mathbf{29} & 40 & 7 & 28 & 19 & 30 & 44 & 26 & 17 \\
0 & 0 & \mathbf{11} & 9 & 43 & 42 & 17 & 51 & 24 & 33 \\
0 & 0 & 0 & \mathbf{9} & 23 & 8 & 7 & 7 & 35 & 46 \\
0 & 0 & 0 & 0 & \mathbf{47} & 30 & 1 & 31 & 41 & 46 \\
0 & 0 & 0 & 0 & 0 & \mathbf{15} & 27 & 12 & 24 & 37 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{7} & 49 & 7 & 37 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{45} & 40 & 41 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{17} & 29 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{44}
\end{bmatrix}$$

$\det A = \prod_{i=1}^{n} a_{ii}$ if e.g row transformations of determinant 1 have been used

Row echelon form and rank

$$\begin{bmatrix}
\boxed{42} & 10 & 24 & 12 & 18 & 10 & 26 & 15 & 19 & 15 \\
0 & 0 & \boxed{40} & 7 & 28 & 19 & 30 & 44 & 26 & 17 \\
0 & 0 & 0 & 0 & 0 & \boxed{42} & 17 & 51 & 24 & 33 \\
0 & 0 & 0 & 0 & 0 & 0 & \boxed{7} & 7 & 35 & 46 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{31} & 41 & 46 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{37} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Pivot index: index of first non zero entry in the row

Echelon: pivot indices are strictly increasing

Theorem of linear algebra

$A \in \mathsf{K}^{n \times n}$

In $\tilde{O}(n^\omega)$ arithmetic operations one can compute:

- the determinant $\det A$
- the inverse A^{-1} is A is invertible ($A^{-1}A = AA^{-1} = \mathrm{I}$)
- the characteristic polynomial of A
- the rank of A and an echelon form of A
- for any $b \in \mathsf{K}^n$, a solution to $Ax = b$ or detect that no solution exist
- a kernel basis ($Ax = 0$)

$$\begin{bmatrix} a_{1,1} & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}$$

- Possibly, pivoting for having $a_{1,1} \neq 0$
- $\mathrm{row}_i \leftarrow \mathrm{row}_i - (a_{i,1}/a_{1,1})\,\mathrm{row}_1$

$$\begin{bmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & * & * & * & * & * \end{bmatrix}$$

- Possibly, next entries zero in second column

$$\begin{bmatrix} * & * & * & * & * & * \\ 0 & 0 & a_{2,3} & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \end{bmatrix}$$

- Possibly, next entries zero in second column
- $\text{row}_i \leftarrow \text{row}_i - (a_{i,3}/a_{2,3})\,\text{row}_2$

$$\begin{bmatrix} * & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- (Row) Echelon form
- $\operatorname{rank} A = 4$

$$\begin{bmatrix} * & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- (Row) Echelon form
- $\operatorname{rank} A = 4$

Matrix factorization:

$$T = U_n \times \ldots \times U_2 \times A$$

Exercise

$A \in \mathsf{K}^{n \times n}$, splitting with $(n/2) \times (n/2)$ invertible blocks

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$\begin{bmatrix} \mathrm{I} & 0 \\ -CA^{-1} & \mathrm{I} \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathrm{I} & -A^{-1}B \\ & \mathrm{I} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix}$$

$$M^{-1} = R \cdot \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \cdot L$$

Show that an algorithm in $O(n^\omega)$ for the multiplication gives an algorithm in $O(n^\omega)$ for the inversion ($2 \leq \omega \leq 3$)

Matrix multiplication $\rightsquigarrow$ Basic linear algebra

Matrix multiplication $\rightsquigarrow$ Basic linear algebra

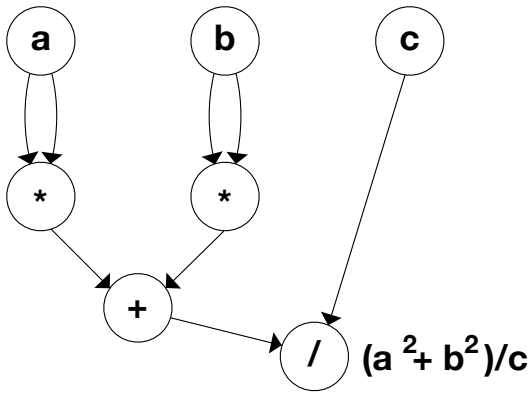
Matrix multiplication $\overset{?}{\leftrightsquigarrow}$ Basic linear algebra

Determinant versus matrix multiplication ?

[Baur & Strassen 1983]

[Chen, Kayal & Wigderson 2010] *Partial Derivatives in Arithmetic Complexity and Beyond*

SLP or arithmetic circuit model



With input gates $x_1, \ldots, x_n$, computes a rational function $f \in \mathsf{K}(x_1, \ldots, x_n)$

Theorem: Given a circuit of size s for $f \in \mathsf{K}(x_1, \ldots, x_n)$, one can compute a circuit of size $O(s)$ for the n first-order partial derivatives of f

Hint: proceed inductively backwards from the output using Leibniz's rule

Ex: $p(x) = (x-1)(x-2)\ldots(x-n)$, evaluate $p'(x) = \sum_{i=1}^{n} \prod_{j \neq i} (x-j)$?

$p(x) \leftarrow q(x) \times (x-n)$ gives $p'(x) \leftarrow q'(x) \times (x-n) + q(x)$

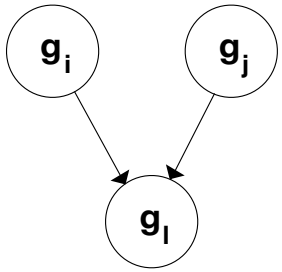
Rough sketch

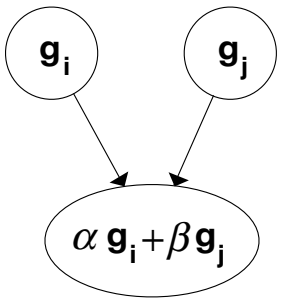
Circuit gates: $x_1, \ldots, x_n, g_1, \ldots, g_l$

$\Delta_l(x_1, \ldots, x_n, g_1, \ldots, g_l)$

$$\frac{\partial \Delta_l}{\partial g_l} = 1$$

$g_l = \alpha g_i + \beta g_j$


g_i
g_j
g_l


$\mathbf{g}_i$
$\mathbf{g}_j$
$\alpha\,\mathbf{g}_i+\beta\,\mathbf{g}_j$

Rough sketch

Circuit gates: $x_1, \ldots, x_n, g_1, \ldots, g_l$

$\Delta_l(x_1, \ldots, x_n, g_1, \ldots, g_l)$

$$\frac{\partial \Delta_l}{\partial g_l} = 1$$

$g_l = \alpha g_i + \beta g_j$

$\Delta_{l-1} = (x_1, \ldots, x_n, g_1, \ldots, \alpha g_i + \beta g_j)$

$$k \neq i, j : \frac{\partial \Delta_{l-1}}{\partial g_k} \equiv \frac{\partial \Delta_l}{\partial g_k}$$

$$\frac{\partial \Delta_{l-1}}{\partial g_i} \equiv \frac{\partial \Delta_l}{\partial g_i} + \alpha \frac{\partial \Delta_l}{\partial g_l}$$

1. Determinant $\Longrightarrow$ Inversion

[Cramer]

$$A^{-1} = \frac{1}{\det A} C^{\mathsf{T}}, \quad \text{with the comatrix } C_{ij} = (-1)^{i+j} \det A \downarrow_{i,j}$$

[Laplace]

$$\det A = \sum_{i=1}^{n} a_{ij} C_{ij}$$

1. Determinant $\Longrightarrow$ Inversion

[Cramer]

$$A^{-1} = \frac{1}{\det A} C^{\mathsf{T}}, \quad \text{with the comatrix } C_{ij} = (-1)^{i+j} \det A \downarrow_{i,j}$$

[Laplace]

$$\det A = \sum_{i=1}^{n} a_{ij} C_{ij}$$

$$\rightsquigarrow \quad \frac{\partial \det A}{\partial a_{ij}} = C_{ij}$$

2. In an algorithmic sense, inversion is as hard as multiplication:

[Winograd]

$$\begin{bmatrix} I & A & 0 \\ 0 & I & B \\ 0 & 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} I & -A & A \times B \\ 0 & I & -B \\ 0 & 0 & I \end{bmatrix}$$

[Baur & Strassen 1983]

Open problem

"... shows that the determinant has roughly the same complexity as matrix multiplication or inversion. It would be interesting to have a similar result for **solving a system of linear equations.**"

Key topic

Fundamental problems in linear algebra over a field are in some way equivalent.

What about polynomial and integer matrices?

Formal power series

Formal power series

Over a commutative* ring R: R$[[x]]$

$$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots = \sum_{i \geq 0} a_i x^i$$

- addition, multiplication
- multiplicative inverse
  if $a(x) = 1 + x\,b(x)$, then $a(x) \times \left(1 - x\,b(x) + x^2\,b^2(x) - x^3\,b^3(x)\right) = 1$

R$[[x]]$ **is a commutative ring, an element is invertible iff** a_0 **is invertible**

Formal power series

- **Truncation**
  $a(x) \bmod x^n = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$

Costs

Series operations modulo x^n: $O(\mathsf{M}(n)) = \tilde{O}(n)$ operations in R

Note: the division by $1 + x\,b(x)$ requires no division

Can division help when computing polynomials?

- For matrix multiplication ?
- For computing the determinant ?

Can division help when computing polynomials?

[Strassen 1973] [Kaltofen 1988]

Theorem: If $f \in \mathsf{K}[x_1, \ldots, x_m]$ of degree n can be computed by a program of length L, then f can be computed by a program of length $O(L\,\mathsf{M}(n)) = \tilde{O}(nL)$ without divisions.

(Here: SLP model, K infinite)

Consequence: divisions do not help for matrix multiplication.

Note: ω can depend only (if at all) on the characteristic of the field

Elimination of divisions

$$\begin{bmatrix} 1 + x\,a(x) & x\,b(x) \\ x\,c(x) & 1 + x\,d(x) \end{bmatrix}$$

$$\begin{bmatrix} 1 + x\,a(x) & x\,b(x) \\ x\,c(x) & 1 + x\,d(x) \end{bmatrix} \rightarrow \begin{bmatrix} 1 + x\,a(x) & x\,b(x) \\ 0 & 1 + x\,d(x) + x^2\,e(x) \end{bmatrix}$$

$$\text{row}_2 \leftarrow \text{row}_2 - \Big((x\,c(x)) \cdot (1 + x\,a(x))^{-1} \Big)\ \text{row}_1$$

$$\text{row}_2 \leftarrow \text{row}_2 - x f(x)\,\text{row}_1$$

Gaussian elimination without divisions

- Elimination with input $I + x\,B(x)$ modulo x^k requires no divisions

Determinant of A?

- Determinant of $M(x) = I + x\,(A - I)$ modulo x^{n+1} requires no division
- $\det M(x)$ has degree at most n: $\quad \delta(x) = \det M(x) = \left(\det M(x) \bmod x^{n+1}\right)$
- $\delta(1)$ gives $\det A$

Theorem: The determinant can be computed in $O(n^\omega\,\mathsf{M}(n)) = \tilde{O}(n^{\omega+1})$ ring operations.

Gaussian elimination without divisions

- Elimination with input $I + x\,B(x)$ modulo x^k requires no divisions

Determinant of A?

- Determinant of $M(x) = I + x\,(A - I)$ modulo x^{n+1} requires no division
- $\det M(x)$ has degree at most n: $\quad \delta(x) = \det M(x) = \left(\det M(x) \bmod x^{n+1}\right)$
- $\delta(1)$ gives $\det A$

Theorem: The determinant can be computed in $O(n^\omega\,\mathsf{M}(n)) = \tilde{O}(n^{\omega+1})$ ring operations.

Open problem

Can the determinant be computed in $\tilde{O}(n^\omega)$ without divisions?

Note: best known exponent $\eta(\omega, \zeta) \approx 2.7 < 3.373$ [Kaltofen & Villard 2005]

Rule of thumb:

$$\text{cost} \leq \text{arithmetic cost} \times \text{output degree}$$