Linear Algebra in $\mathbb{Z}$

Claus Fieker

RPTU Kaiserslautern-Landau

March 18, 2025

joint w/ J. Abbott

1. What Is This About?

2. Details

3. Solving

What Is This About?

Task

In ISSAC 2013, Arne Storjohann and Colton Pauderis published a paper promising a practical speedup in determinant computation over the integers - in the hard case.

I put a PhD student to the task of generalizing this to number fields.

She managed the mathematics, and it was fun, but not the performance.

Last year jointly with John Abbott we decided to re-visit this, starting with the integers - as a base case. This is what we found.

Theory

Let $A \in \mathbb{Z}^{n \times n}$ be arbitrary (later non-singular).

Then there are unimodular matrices $U \in \mathrm{Gl}(n, \mathbb{Z})$ s.th. AU is upper triangular, suitable normalized this is unique. The Hermite form of A.

There are S and $T \in \mathrm{Gl}(n, \mathbb{Z})$ s.th. SAT is diagonal with the diagonal elements $\lambda_1 | \lambda_2 | \cdots | \lambda_n$. These elementary divisors are unique. The Smith form of A.

Clearly, up to sign, the determinant of A can be read off either of those normal forms.

Overview

Let $A \in \mathbb{Z}^{n \times n}$ be non-singular.

The “standart” determinant algorithm over the integers is Abbott-Mulders, ISSAC ’99:

- For a random $b \in \mathbb{Z}^n$ solve $Ax = b$ for $x \in \mathbb{Q}^n$
- Let d be the (common) denominator of x
- For enough primes, compute $\operatorname{det}(A)d^{-1} \bmod p$
- Use CRT to obtain $\operatorname{det}(A)/d$

For “random” matrices this is close to optimal.

Facts

The denominator is, in fact, a divisor of the largest elementary divisor of A, not only of the determinant.

Solving $Ax = b$ in itself is non-trivial.

For “random” matrices, there are very few non-trivial elementary divisors and the determinant is about as large as the Hadamard bound.

No-one (outside benchmarks) is interested in large random matrices.

Applications

- Large (sparse) matrices arise in (co)homology computations. Structured Gaussian elemination produces small dense problems.
- Matrix group, characteristic polynomial
- Representation theory
- Determinants are everywhere
- Base case to understand matrices over number fields and other rings

Details

Solving - part 1

In 1992 John Dixon suggested solving $Ax = b$ using p-adic linear lifting: Let p be prime (avoid $\det A$), and $\bar{B} = \mathrm{Inv}(A) \bmod p$ the inverse.

Set $x_0 = \bar{B}b$ a lift (in $\{0, \ldots, p-1\}$) and $b_1 := (Ax_0 - b)/p$.

Now $x_1 = \bar{B}b_1$ (a lift) and $b_2 = (Ax_1 - b_1)/p$.

Then $A(x_0 + px_1) = b \bmod p^2$.

Iterate this until the desired precision, then use rational reconstruction/ Farey lifting to get the (probable) solution in $\mathbb{Q}$. Verify the solution.

This extends to non-unique solving as well.

Bounds

Precision bounds are easily obtained from Cramer's rule and the Hadamard bound. In fact $\det(A) \le n^{n/2}\|A\|_\infty^n$, assuming $\|b\|_\infty = \|A\|_\infty$ (approximately), the bound is the same in numerator and denominator, thus the precision is (essentially)

$$2\log_p(n^{n/2}\|A\|_\infty^n) = 2n\log_p(\sqrt{n}\|A\|_\infty)$$

Runtime

The Dixon iteration is

- 1 matrix × vector modulo p, so n^2 operations in $\mathbb{F}_p$ (and no hidden constants!)
- 1 matrix × vector in $\mathbb{Z}$ - in fixed size, so $\log \|A\|_\infty \cdot n^2$

Since the precision is $O(n)$, Dixon has a total complexity of $O(n^3)$ with small constants $(2 + \log \|A\|_\infty)$ - and no chance of ω via fast matrix multiplication. (B can be obtained in $O(n^\omega)$)

Back to determinant

In the generic case, Abbott and Mulders obtain

- $O(n^3)$ for the Dixon
- $O(1)$ additional primes at cost of $O(n^\omega)$ for the mod-p determinants (as Hadamard is “sharp”)

However, worst case is much worse:

- $\mathtt{det}(A)$ can be 1 or
- all elementary divisors are the same

in either case the runtime then goes up to $O(n^4)$ - the complexity for pure CRT determinant.

Practical n^3?

$$Ax = b$$

Storjohann suggested using not only using the denominator - but also the numerator. So $x = c/d$ for $d \in \mathbb{Z}$ and $c \in \mathbb{Z}^n$, lowest terms.

Setting

$$C = \begin{bmatrix} d & 0 \\ c^t & I_n \end{bmatrix} \in \mathbb{Z}^{(n+1)\times(n+1)}$$

Then (row) HNF is

$$\tilde{H} = \begin{bmatrix} 1 & \tilde{c} \\ 0 & H \end{bmatrix}$$

and $A_1 := H^{-1}A \in \mathbb{Z}^{n\times n}$ is a basis for the col span of $(A|b)$.

Moreover $\det A_i$ is smaller by a factor of d and solving $A_1 x = b_1$ can recover a new denominator.

Practical n^3?

If the number of elementary divisors is $O(1)$ then this computes the determinant in time $O(n^3)$ - but how do you prove the correctness?

Two options:

- close the gap between the computed $\det A$ and the Hadamard bound using CRT
- prove that the final matrix A_i has det ± 1: unimodularity certification.

(Storjohann and Pauderis gave a procedure for writing down H in time $O(n^2)$ without actually doing an HNF)

Unimodularity

In ISSAC 2012, S&P gave a straight line program for the inverse of an integer matrix in base X (for suitable X):

$$\mathrm{Inv}(A) = (B_0(I + R_0X) + M_0X^2)(I + R_1X^{2^2-1}) + M_1X^? \dots$$

by alternating between linear and quadratic steps.

All matrices here have entries $\leq X$ and $X \approx 3.6n^2\|A\|_\infty$.

Magic.

A is unimodular iff R_i becomes 0 eventually (after at most $O(\log n)$ steps). So $\log n$ steps of cost n^ω will (dis-)proof unimodularity.

Algorithm

```
$B_0 = \mathrm{Inv}(A) \bmod X$
$R_0 = (I - AB_0)/X$
for $i = 0, \ldots$
    $S = R_i^2$
    $M_i = B_0 S \bmod X$
    $R_{i+1} = (S - AM_i)/X$
```

(To test unimodularity we do not need to store the M_i and R_i)

HNF

When run to completion S&P have, at the end

$$\prod \mathrm{Inv}(H_i)A = U$$

or

$$A = \prod H_i U$$

so $\prod H_i$ is upper triangular - and almost the HNF.

This is (can be) *much* faster than a direct computation.

But if there are too many elementary divisors, this is still too slow.

F.-Abbott

After the solving we have (a large part of) the largest elementary divisor. Remove the divisor from the determinant using S&P and repeat.

The next denominator wil be

- much smaller
- (almost) the 2nd elementary divisor
- a multiple of all the other elementary divisors as well.

Compute H the HNF of $(A|dI)$ for the current (left-over A and denominator d).

This is a basis for the span of $[A] + [d\mathbb{Z}^n] \supset [A]$.

So $\mathrm{Inv}(H)A$ is integral - and will have *all* elementary divisors removed.

(OK: not really, we might miss small primes, so we might have to repeat this)

(or compute the missing parts using CRT)

F.-Abbott

Assume we use CRT and modulo p determinants a bit and suspect that the determinant is in fact small, d. How can we prove that?

As above, compute H, the HNF of $(A|dI_n)$, then $\mathrm{Inv}(H)A$ and use unimodularity certification to prove the result.

Strategy

- compute $\det$ modulo a few primes. If $\det$ looks small: use HNF to prove it.
- Dixon and S&P to remove the big denominator
- if it looks like $\det$ is now small, use HNF and prove it.
- Proof: if the gap to Hadamard is small, use CRT, else unimodularity

Magma is using Lübeck as well: if small p divides the determinant, he gave a method of computing the p-part of the determinant (well the valuations of the elementary divisors).

Solving

Solving - 2

Given that we have the X-adic inverse of A, we can use this to solve $Ax = b$ this way!

Attempt 1: multiply b by the inverse using the SLP. Runtime is terrible, we we have many multiplications with big integers.

Attempt 2: Storjohann again (2005, The shifted number system for fast linear algebra on integer matrices.): compute a X-adic expansion of $\mathrm{Inv}(A)b$: put each X-adic digit of b and the solution into a column of a matrix.

- multiplication of matrices with entries of size X
- cumulative: can use $O(n^\omega)$ matrix multiplication!
- in total: solving in time $O(n^\omega)$ or so...

(There is some fine print)

Solving - 2 - fine print

- it works and is competitive
- there is a renormalization missing
- the asymptotic complexity is $O(n^3)$ vs $O(\log n n^\omega)$ for $\mathbb{Z}$-operations. For $\log \|A\|_\infty = O(1)$, the cross over is past $n = 10^6$, for $\log \|A\|_\infty$ large (enough) it wins. ($n = 200$, $\|A\|_\infty = 10^{1000}$ factor of 3)
- Storjohann suggests a double modular method (RNS) and using BLAS directly - that we couldn't verify any improvement there. Our code is faster than IML.
- Storjohann suggests removing the even part of the determinant and then choose X as a power of 2.

Solving - 3

Ala Kim-Manuel Klein, Janina Reuter (arXiv:2408.06685, Faster Lattice Basis Computation via a Natural Generalization of the Euclidean Algorithm)

Task: solve $Ax = b$ with $A \in \mathbb{Z}^{n\times(n+1)}$, $b \in \mathbb{Z}^n$ and $x \in \mathbb{Z}^{n+1}$

Idea: find unimodular C s.th. $CA = (\tilde{A}|0)$ and $\tilde{A}$ is a basis for the col. span, then solve $\tilde{A}\tilde{x} = b$ and recover x from $\tilde{x}$.

By above, C can be found in time $O(n^\omega)$, $\tilde{x}$ as well... so this gives the fastest know algorithm for this!

Times - HNF

Fix p and choose $A \in \mathbb{Z}^{p \times p}$ for $A_{i,j} = i^{j-1} \bmod p$. They are “known” to be nasty. Timings are for HNF.

p	131	233	331	431	541	631	733	839	937
O	0.2	3.8	6.6	28.0	37.3	65.2	114.5	200	256
M	0.13	1.77	18.1	56.0	152	281.2	530	340	1615

M = Magma, O = Oscar (julia)

Times - Solving

For $n = 50, 100$ we chose $A \in \mathbb{Z}^{n \times n}$ and $b \in \mathbb{Z}^n$ with entries uniform in $[-10^{100i}, 10^{100i}]$. Times are for solving using Dixon and DoublePlusOne:

$n = 50$:

i	1	2	3	4	5	6	7	8	9	10
D	0.09	0.24	0.46	0.62	0.84	1.1	1.3	1.5	2.3	2.3
+	1.2	1.4	1.9	2.0	2.2	2.6	3.0	3.6	4.1	4.7

$n = 100$:

i	1	2	3	4	5	6	7	8	9	10
D	0.6	1.8	3.1	4.3	7.2	11.4	17.1	21.3	23.7	27.1
+	1.9	2.7	2.8	5.1	6.7	9.5	11.7	15.2	17.2	20.1

Times - Solving

For $n = 150, 200$ we chose $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^n$ with entries uniform in $[-10^{100i}, 10^{100i}]$. Times are for solving using Dixon and DoublePlusOne:

$n = 150$:

i	1	2	3	4	5	6	7	8	9	10
D	3.0	14.9	17.7	27.7	42	49	63	77	94	113
+	3.3	6.8	9.9	13.7	18.6	24	31	38	46	53

$n = 200$:

i	1	2	3	4	5	6	7	8	9	10
D	11	24	53	80	110	129	155	202	232	272
+	6	11	16	24	32	41	53	65	78	94

Matrices

Let $Ac = db$ a solution entirely in integers, and in lowest terms (i.e. $\gcd(\mathrm{Cont}(c), d) = 1$). Then there is $T \in \mathrm{Gl}(n+1, \mathbb{Z})$ s.th.

$$T \begin{bmatrix} d & 0 \\ c & I \end{bmatrix} = \begin{bmatrix} 1 & \tilde{c} \\ 0 & H \end{bmatrix}$$

is in Hermite form. Solving for T and using

$$\begin{bmatrix} d & 0 \\ c & I \end{bmatrix} = \begin{bmatrix} d & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & I \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ c & I \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -c & I \end{bmatrix}$$

We get

$$T = \begin{bmatrix} d & 0 \\ c & I \end{bmatrix}^{-1} \begin{bmatrix} 1 & \tilde{c} \\ 0 & H \end{bmatrix} = \begin{bmatrix} (1 - \tilde{c}c)/d & \tilde{c} \\ -Hc/d & H \end{bmatrix}$$

and

$$T^{-1} = \begin{bmatrix} d & -d\tilde{c}H^{-1} \\ c & (I - c\tilde{c})H^{-1} \end{bmatrix}$$

ctnd.

Applying this

$$\begin{aligned}(-b|A)T^{-1} &= (-db + Ac|db\tilde{c}H^{-1} + A(I - c\tilde{c})H^{-1})\\ &= (0|((db - Ac)\tilde{c}H^{-1} + AH^{-1})\\ &= (0|AH^{-1})\end{aligned}$$

so AH^{-1} is a basis for $(A|b)$

H is “for free” by S&P (from c and d), $\tilde{c}$ comes from any Bezout relations for c and d reduced by H. Multiplication of a vector by H^{-1} is solving with a triangular matrix, so at cost $O(n^2)$. If need, both T and T^{-1} can thus be obtained at cost of $O(n^2)$ as well.

now solving

Applying T to solving:

$$(-b|A)\begin{bmatrix}x_1\\x_2\end{bmatrix}=\mu$$

$$(-b|A)T^{-1}T\begin{bmatrix}x_1\\x_2\end{bmatrix}=\mu$$

so

$$\begin{bmatrix}y_1\\y_2\end{bmatrix}:=T\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

and $(-b|A)T^{-1}=(0|AH^{-1})$, so $y_1=0$ and $AH^{-1}y_2=\mu$

$$T^{-1}\begin{bmatrix}0\\y_2\end{bmatrix}=\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

Back to Number Fields

ala Suranimalee:

- “All” also works for number fields
- The hidden constants are much larger
- Should the denominator be in $\mathbb{Z}$ (unique) or in $\mathbb{Z}_K$ (small)?
- This approach computes the ideal generated by $\det$
- In $\mathbb{Z}$ the gap between the (principal) ideal and the correct generator is only a sign (up to units)
- In $\mathbb{Z}_K$ is the unit much much harder!
- Nevertheless, the “plain” Dixon is very competitive.

Back to Number Fields - 2

ala Suranimalee:

- Now that we understand the $\mathbb{Z}$ case, we should be able to lower the constants in the number field case!
- In many applications, the determinant ideal is enough.
- The unimodularity certification can also be done over $\mathbb{Z}$ via restriction-of-scalars. What is better?
- Play with vector reconstruction?