Gaussian lattice reduction algorithm terminates in polynomial time

Michael Soltys

November 18, 2011

Abstract

In this short note we show that the classical Gaussian reduction algorithm for finding the shortest vector in an $\mathbb{R}^2$ lattice works in polynomial time. In other words, we show that the SVP (shortest vector problem) has a polytime solution in the case of two dimensions. This has always been known, but the author could not find an explicit proof.

1 Gaussian reduction algorithm

We show that the Gaussian lattice reduction algorithm terminates in polynomial time. The algorithm takes as input two vectors v_1, v_2, and replaces the longer, say v_2, with $v_2 - mv_1$ where $m = \lfloor p \rceil = \lfloor p + \frac{1}{2} \rfloor$ where $p = (v_1 \cdot v_2)/\|v_1\|^2$, as long as $m \neq 0$, at which point it terminates. The algorithm also swaps v_1, v_2 as needed to maintain the property that $\|v_1\| \leq \|v_2\|$.

First, it follows directly from the fact that $v_2 - pv_1$ is the projection of v_2 onto the orthogonal complement of v_1, and from the Pythagorean theorem that:

$$\|v_2'\|^2 \leq \|v_2\|^2 + \left(\frac{1}{4} - p^2\right) \|v_1\|^2, \tag{1}$$

where $v_2' = v_2 - mv_1$, i.e., v_2' is the result of one iteration of the algorithm. To be more precise we prove (1):

$$\begin{aligned}
\|v_2'\|^2 = \|v_2 - mv_1\|^2 &= \|v_2 - pv_1\|^2 + \|(m-p)v_1\|^2 && \text{by Pythagorean Thm} \\
&\leq \|v_2 - pv_1\|^2 + \frac{1}{4}\|v_1\|^2 && \text{since } |m-p| \leq \tfrac{1}{2} \\
&= \|v_2\|^2 - 2p(v_1 \cdot v_2) + p^2\|v_1\|^2 + \frac{1}{4}\|v_1\|^2 \\
&= \|v_2\|^2 - p^2\|v_1\|^2 + \frac{1}{4}\|v_1\|^2 && \text{since } p\|v_1\|^2 = v_1 \cdot v_2
\end{aligned}$$

It is easy to show that for $|p| \leq 1$ the algorithm terminates in at most two more iterations, and so we assume that $|p| > 1$. With this assumption in place (1) becomes:

$$\|v_2'\|^2 \leq \|v_2\|^2 - \frac{3}{4}\|v_1\|^2, \tag{2}$$

and we consider two cases.

Case 1 $\|v_2\| \leq 2\|v_1\|$. Then we have that $-\frac{1}{4}\|v_2\|^2 \geq -\|v_1\|^2$, so from (2) we obtain the following bound: $\|v_2'\|^2 \leq \frac{13}{16}\|v_2\|^2$.

Case 2 $\|v_2\| > 2\|v_1\|$. If $\|v_2'\|^2 \leq \frac{13}{16}\|v_2\|^2$ then we are done. Otherwise we have the following two:

- $\|v_2'\|^2 \geq \frac{13}{16}\|v_2\|^2$ and
- $\|v_2\| > 2\|v_1\|$.

But with those two assumptions we obtain:

$$\|v_2'\| > \frac{\sqrt{13}}{4}\|v_2\| > \frac{\sqrt{13}}{4}2\|v_1\| = \frac{\sqrt{13}}{2}\|v_1\| > \|v_1|,$$

which means that in the next iteration $v_1'' = v_1' = v_1$, i.e., there is no swapping, and

$$|p| = \left|\frac{v_1 \cdot v_2'}{\|v_1\|^2}\right| = \frac{|v_1 \cdot v_2'|}{\|v_1\|^2} = |\cos(\theta)|\frac{\|v_2'\|}{\|v_1\|},$$

and since $|\cos(\theta)| \leq \frac{\|v_2'\|}{\frac{1}{2}\|v_1\|}$, it follows that $|p| \leq 1$, and so we have termination in at most two steps.

Therefore, putting the two cases together, we have that the algorithm terminates in at most two steps, or we have a decrease of $\|v_2'\|$ by a constant factor, i.e.,

$$\|v_2'\|^2 \leq \frac{13}{16}\|v_2\|^2.$$

Using *Hadamard's inequality*, $\det(L) \leq \|v_1\|\|v_2\|$, we can now conclude that the algorithm runs in polynomial time as follows.

Let $D = \|v_1\|\|v_2\|$ be our parameter; then $|\det(L)| \leq D$, where $\det(L) = \det(v_1, v_2)$ is fixed, and so D is bounded from below by a positive number. At the same time, after each iteration D decreases by a factor of $\frac{\sqrt{13}}{4}$. Therefore, the number of steps is bounded by n where:

$$\left(\frac{\sqrt{13}}{4}\right)^n \|v_1\|\|v_2\| \leq \det(v_1, v_2).$$

Solving for n we have that:

$$n = \log_2\left(\frac{16}{13}\right)\left[\log(\det(v_1, v_2)) - \log(\|v_1\|) - \log(\|v_2\|)\right],$$

i.e., the running time is given by a polynomial in the lengths of the binary encodings of the coordinates of the two vectors.