**Algebra and Computation** Course Instructor: V. Arvind

# Lecture 16 and 17: Linear Diophantine Equations

*Lecturer: V. Arvind* *Scribe: Ramprasad Saptharishi*

## 1 Overview

Our route now is towards factorization of polynomials over $\mathbb{Q}$. This require a lot of machinery to be built and we shall do it over the next few lectures.

In this class, we shall look at solving a system linear diophantine equations and its connection to lattices.

## 2 Linear Diophantine Equations

A linear diophantine equation is of the form $a_1x_1 + a_2x_2 + \cdots a_nx_n = b$ and we are interested in integer solutions $\{x_i\}$. A system of linear diophantine equations is a bunch of such equations. This can be written in a matrix notation as follows:

Given a rational $m \times n$ matrix (matrix with rational entries) $A$, and a rational $m$-vector $b$, we are looking for integral vectors $x$ that satisfy $Ax = b$.

We are looking for a polynomial time algorithm to give us all possible solutions to this equation. Getting all solutions is simple once we have a single solution $\hat{x}$. All we need to do get the solution space $\mathcal{S}$ to $Ax = 0$ and and the solutions to the diophantine system is just $\hat{x} + \mathcal{S}$.

Firstly, we can assume that $A$ is of full rank (row rank is equal to $m$) since even otherwise we can drop the other rows since they are linear combinations of the independent rows. Another thing we can assume is that the entries are integral (we can just scale the matrix up by the LCM of the denominators and rescale it in the end).

The *hermite normal form* is the key to finding solutions to the diophantine equations.

## 3 Hermite Normal Form

A full rank matrix $A$ is said to be in *hermite normal form* if

- The matrix $A$ is of the form $[B \ \ 0]$ where $B$ is a $m \times m$ matrix that is invertible.
- $B$ is lower triangular.
- The diagonal entries of $B$ are strictly greater than zero.
- Other entries are non-negative.
- For every row, the unique maximum of that row is attained at the diagonal entry.

An example is the following:

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 2 & 1 & 3 & 0 & 0 \end{bmatrix}$$

We will now see that every matrix can be converted into one in HNF with simple operations.

### 3.1 Converting to HNF

We want to start with a full rank matrix $A$ and convert it to one in HNF using simple operations called *modular column operations*. These are operations of the form

- exchange two columns
- multiply a column by $-1$
- Replace a column $C_i$ by $C_i + kC_j$ where $j \neq i$ and $k \in \mathbb{Z}$.

Note that each of the above operation just amounts to post multiplying by a matrix of determinant $\pm 1$. And any sequence of of modular column operations would just be multiplying $A$ by a single unitary matrix $U$.

**Theorem 1.** *Every full rank rational matrix can be converted into a matrix in HNF using modular column operations*

*Proof.* The proces will be row-wise. Assume we have got it to the form

$$\begin{bmatrix} B & 0 \\ C & D \end{bmatrix}$$

where $[B \ \ 0]$ is already in HNF.

First, multiply the columns of $D$ by $-1$ to make the top row of $D$ with just non-negative entries. Then, rearrange the columns to make sure that the entries are non-decreasing down the row, that is, $\delta_1 \geq \delta_2 \geq \cdots \geq 0$. Note that all of them can't be zero since we have assumed that $A$ is of full rank (thus forces $D$ to also be full rank).

Suppose the gcd of the $\delta_i$ was $d$, then implementing euclid's algorithm using modular column operations, one of the $\delta_i$ can be made equal to $d$. Once this is done, since every other element is a multiple of $d$, they can be killed. Thus we can make sure that $\delta_1 = d$ and $\delta_i = 0$ for all $i > 1$.

Now to ensure the unique maximum property, let the first row of $C$ be $c_1, c_2, \cdots, c_k$. Use the division algorithm to write $c_i = md + r$ where $r < d$ and replace $C_i$ by $C_i - mD_1$ to change every $c_i$ to its positive reminder modulo $d$. Thus $\delta_1$ would be the unique maximum in that row and every entry to its left is non-negative.

Proceeding this way, we can convert the matrix to one in HNF. □

The following characterizition is extremely powerful since it takes us from an existential quantifier to a universal quantifier.

**Theorem 2.** *Let $A$ be a full rank rational matrix and $b$ be a rational vector. The following are equivalent:*

1. *$\exists$ an integral $x$ such that $Ax = b$.*
2. *$\forall$ rational $y$, $yA$ is integral implies $yb$ is rational.*

*Proof.* One direction is clear, if $Ax = b$ has an integral solution, then $yAx = yb$. Since $x$ is integral, $yaA$ integral would clearly force $yb$ to be integral.

The converse is slightly tricky. We know that there exists a unimodular matrix that converts $A$ to the HNF. And more over, the conversion preserves the equivalence in the theorem and hence we'll work with that.

$$[B \;\; 0]\, x = b$$

Clearly, $x = \binom{B^{-1}b}{0}$ is a solution to the above equation; the only trouble we could have is that this matrix need not be integral since $B^{-1}$ could have fractional entries.

Now consider the matrix $B^{-1}\,[B \;\; 0]$ row-wise. $B_i^{-1}\,[B \;\; 0]$ is integral would imply $B_i^{-1}b$ is integral. But $B^{-1}\,[B \;\; 0] = [I \;\; 0]$ and hence forces $B^{-1}b$ to be integral. Hence the solution $x = \binom{B^{-1}b}{0}$ is indeed an integral solution. □

# 4 HNF and Lattices

Let $a_1, a_2, \cdots, a_m$ be a spanning set for $\mathsf{R}^n$. The lattice created by them is the set of all integral combinations of these vectors.

$$L(a_1, a_2, \cdots, a_m) = \left\{ \sum \lambda_i a_i \ : \ \lambda_i \in \mathbb{Z} \right\}$$

When each $a_i$ has only rational entries, this will form a discrete subset in $R^n$. The HNF of a matrix $A$ completely determines the lattice generated by the columns of $A$. An immediate corollary is that the HNF is unique.

**Theorem 3.** *For any two matrices $A$ and $A'$, their columns generate the same lattice if and only if the non-zero part of their HNFs are identical.*

*Proof.* Of course, if the HNFs are identical we can just invert the unitary tranformation and hence the lattices are equal. Suppose $A$ and $A'$ give the same lattice, let the invertible of the HNF be $B$ and $B'$ respectively. Let $i$ be the first row where $B$ and $B'$ differ and let the column index be $j$.

Without loss of generality, assume that $0 \geq b_{ij} < b'_{ij} \leq b'_{ii}$. Now look at the vector $b'_j - b_j$, this is cleary in the lattice formed by $B'$ since we are assuming tht both lattices are the same. And more over, the first $i-1$ coordinates of this vector is zero since we have chosen $i$ to be the first row where it differs. Hence if $b'_j - b_j = \sum \lambda_k b'_k$, then $\lambda_1, \lambda_2, \cdots, \lambda_{i-1} = 0$. But since $b'_{ij} - b_{ij} < b'_{ii}$, an integer sum of $\{b_k\}_{k \geq i}$ can never create the coordinate $b'_{ij} - b_{ij}$, giving us the contradiction.

Hence the invertible parts of the HNFs are identical. □

# 5 Bounding Sizes

In order to talk about efficient algorithms for the linear diophantine equations, we first need to see if HNFs help at all. What if the entries of the HNF are huge? What if the unitary matrix has massive numbers in it? Do we have good bounds for them?

The answer is yes!

## 5.1 Bounds on HNF size

The following fact makes the bound possible.

**Fact 1.** $\det B$ *is equal to the gcd of all $m \times m$ subdeterminants of $A$.*

*Proof.* Pretty simple, just need to show that the gcd is unaltered in the conversion process. We leave it to the reader to complete the proof. □

A geometric way to look at this is through the lattice. The determinant gives the volume of the principle parallelopiped in the lattice and that is the gcd of $m$-parallelopipeds by the columns.

With this, we then have $\det A \leq n!(\max |A_{ij}|)^n$ but this is still polynomially many bits. And since our final matrix is upper triangular with integer entries, each of the diagonal entries is bounded by this and hence the entire matrix.

Thus, the size of $B$ is polynomially bounded.

### 5.2 Bounds on U

Without loss of generality, we can assume that the first $m$ columns of $A$ are linearly independent. Hence let $A = [A' \;\; A'']$ where $A'$ is an invertible $m \times m$ matrix.

Just replace $A$ by the invertible matrix

$$\hat{A} = \begin{pmatrix} A' & A'' \\ 0 & I \end{pmatrix}$$

Suppose this had its HNF as

$$\mathcal{B} = \begin{pmatrix} B & 0 \\ B_1 & B_2 \end{pmatrix}$$

then clearly $U = \hat{A}\mathcal{B}$ and is polynomially bounded.

Hence, the size of $U$ is not too large.

## 6 Keeping Numbers Small

Though we know that the numbers at the end would be small, we need to make sure that they do not blow up in any intermediate step. The following really clever trick was given by Bachem-Kannan.

Assume that $A = [A' \;\; A'']$ where $A'$ is a non-singular square matrix. Let $|\det A'| = M$. Replace $A$ by the matrix

$$\bar{A} = \left[\begin{array}{c|ccc} & M & & \\ A & & M & \\ & & & \ddots \\ & & & & M \end{array}\right]$$

**Claim 4.** *The matrices $A$ and $\bar{A}$ generate the same lattice.*

*Proof.* We know that $A' \cdot adjA' = \det A' \cdot I$ and since $adjA'$ is an integer matrix, $M \cdot I$ is generated by the columns of $A'$. And hence, both matrices generate the same lattice. □

Since the lattices are the same, computing the HNF of this matrix would be the same as computing the HNF of $A$. The good thing in $\bar{A}$ is that you can use its columns to make sure that numbers don't blow up; whenever they do, just use the appropriate column to drive it smaller than $M$.

But notice that these columns help only till the triangulation of the matrix. How do we drive the diagonal to the unique maximum? What if numbers blow up there?

The good news is that it won't. We have made sure that every entry of the matrix is atmost $M$. The operation of converting every non-diagonal entry to it's remainder modulo $M$ can atmost blow indices by a factor of $(M+1)$. Hence, at the end of it, we would have atmost $M(M+1)^m$ and this is still not large in terms of bit complexity!

Therefore we are in good shape. Since we now know to get the HNF, this solve the linear diophantine equation.

# 7 Arithmetic circuits with bounds on final answer

Suppose we have an arithmetic circuit, a circuit with multiplication and addition gates, with inputs provided and the circuit evaluating some polynomial.

Suppose we are given the promise that the final answer is upper bounded by some $M$, we could try the following thing:

- replace every multiplication gate by a multiplication $\pmod M$ gate
- replace every addition gate by an addition $\pmod M$ gate

This modification will not change the output of the circuit at all! The modification ensures that the numbers never get too large in the middle.

This howeever cannot be directly used in the HNF setting since it is not just a circuit we are looking at. There are branches based on comparisons and $\pmod M$ gates need not preserve them.

Nevertheless, this is a great trick.