> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

Introduction to Number Theory Chapter 2 Euclid's Algorithm and Applications

Robert C. Vaughan

January 13, 2025

イロト 不得 トイヨト イヨト ニヨー

Sac

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

Introduction to Number Theory Chapter 2 Euclid's Algorithm and Applications

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • The question arises. We know that given integers *a*, *b* not both 0, there are integers *x* and *y* so that

$$(a,b)=ax+by.$$

How do we find x and y?

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

Introduction to Number Theory Chapter 2 Euclid's Algorithm and Applications

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • The question arises. We know that given integers *a*, *b* not both 0, there are integers *x* and *y* so that

$$(a,b)=ax+by.$$

How do we find x and y?

• A method for solving this problem, known as Euclid's algorithm, first appeared in Euclid's *Elements* more than 2000 years ago.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

Introduction to Number Theory Chapter 2 Euclid's Algorithm and Applications

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • The question arises. We know that given integers *a*, *b* not both 0, there are integers *x* and *y* so that

$$(a,b)=ax+by.$$

How do we find x and y?

- A method for solving this problem, known as Euclid's algorithm, first appeared in Euclid's *Elements* more than 2000 years ago.
- Moreover this solution gives a very efficient algorithm and it is still the basis for many numerical methods in arithmetical applications.

Introduction to Number Theory Chapter 2 Euclid's Algorithm and Applications

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • The question arises. We know that given integers *a*, *b* not both 0, there are integers *x* and *y* so that

$$(a,b)=ax+by.$$

How do we find x and y?

- A method for solving this problem, known as Euclid's algorithm, first appeared in Euclid's *Elements* more than 2000 years ago.
- Moreover this solution gives a very efficient algorithm and it is still the basis for many numerical methods in arithmetical applications.
- We may certainly suppose that a and b > 0 since multiplying either by (-1) does not change the (a, b) - we can replace x by -x and y by -y.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations We can certainly suppose that b ≤ a. For convenience of notation put r₀ = b, r₋₁ = a.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We can certainly suppose that b ≤ a. For convenience of notation put r₀ = b, r₋₁ = a.
- Now apply the division algorithm iteratively as follows

$$\begin{aligned} r_{-1} &= r_0 q_1 + r_1, \quad 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, \quad 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, \quad 0 < r_3 < r_2, \\ & \dots \\ r_{s-3} &= r_{s-2} q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1} q_s. \end{aligned}$$

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ◆ ●

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We can certainly suppose that b ≤ a. For convenience of notation put r₀ = b, r₋₁ = a.
- Now apply the division algorithm iteratively as follows

 $\begin{aligned} r_{-1} &= r_0 q_1 + r_1, \quad 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, \quad 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, \quad 0 < r_3 < r_2, \\ & \dots \\ r_{s-3} &= r_{s-2} q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1} q_s. \end{aligned}$

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

• That is, we stop the moment that there is a remainder equal to 0.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We can certainly suppose that b ≤ a. For convenience of notation put r₀ = b, r₋₁ = a.
- Now apply the division algorithm iteratively as follows

 $\begin{aligned} r_{-1} &= r_0 q_1 + r_1, \quad 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, \quad 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, \quad 0 < r_3 < r_2, \\ & \dots \\ r_{s-3} &= r_{s-2} q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1} q_s. \end{aligned}$

- That is, we stop the moment that there is a remainder equal to 0.
- This could be r_1 if b|a, for example, although the way it is written out above it is as if s is at least 3.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We can certainly suppose that b ≤ a. For convenience of notation put r₀ = b, r₋₁ = a.
- Now apply the division algorithm iteratively as follows

 $\begin{aligned} r_{-1} &= r_0 q_1 + r_1, \quad 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, \quad 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, \quad 0 < r_3 < r_2, \\ & \dots \\ r_{s-3} &= r_{s-2} q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1} q_s. \end{aligned}$

- That is, we stop the moment that there is a remainder equal to 0.
- This could be r_1 if b|a, for example, although the way it is written out above it is as if s is at least 3.
- The important point is that because $r_j < r_{j-1}$, sooner or later we must have a zero remainder.

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ◆ ●

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Repeating

 $\begin{aligned} r_{-1} &= r_0 q_1 + r_1, & 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, & 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, & 0 < r_3 < r_2, \end{aligned}$

. . .

 $\begin{aligned} r_{s-3} &= r_{s-2}q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1}q_s. \end{aligned}$

▲ロト ▲周ト ▲ヨト ▲ヨト ヨー のくで

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Repeating

$$\begin{aligned} r_{-1} &= r_0 q_1 + r_1, & 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, & 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, & 0 < r_3 < r_2, \end{aligned}$$

$$r_{s-3} = r_{s-2}q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2},$$

$$r_{s-2} = r_{s-1}q_s.$$

ヘロト 人間 ト 人造 ト 人造 ト

≡ 9 < ભ

• Euclid proved that $(a, b) = r_{s-1}$.

. . .

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Repeating

$$\begin{aligned} r_{-1} &= r_0 q_1 + r_1, & 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, & 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, & 0 < r_3 < r_2, \\ & \dots \end{aligned}$$

$$\begin{aligned} r_{s-3} &= r_{s-2}q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1}q_s. \end{aligned}$$

・ロト ・ 同 ト ・ ヨ ト ・ ヨ ト

3

Sac

- Euclid proved that $(a, b) = r_{s-1}$.
- First of all (a, b)|a and (a, b)|b, and so $(a, b)|r_1$.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Repeating

 $\begin{aligned} r_{-1} &= r_0 q_1 + r_1, & 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, & 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, & 0 < r_3 < r_2, \\ & \dots \\ r_{s-3} &= r_{s-2} q_{s-1} + r_{s-1}, & 0 < r_{s-1} < r_{s-2}, \end{aligned}$

 $r_{s-2}=r_{s-1}q_s.$

- Euclid proved that $(a, b) = r_{s-1}$.
- First of all (a, b)|a and (a, b)|b, and so $(a, b)|r_1$.
- Repeating this we get $(a, b)|r_j$ for $j = 2, 3, \ldots, s 1$.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Repeating

$$\begin{aligned} r_{-1} &= r_0 q_1 + r_1, \quad 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, \quad 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, \quad 0 < r_3 < r_2, \\ \dots \\ r_{s-3} &= r_{s-2} q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1} q_s. \end{aligned}$$

- Euclid proved that $(a, b) = r_{s-1}$.
- First of all (a, b)|a and (a, b)|b, and so $(a, b)|r_1$.
- Repeating this we get $(a, b)|r_j$ for $j = 2, 3, \ldots, s 1$.
- On the other hand, starting at the bottom line $r_{s-1}|r_{s-2}$, $r_{s-1}|r_{s-3}$ and so on until we have $r_{s-1}|b$ and $r_{s-1}|a$. Recall that this means that $r_{s-1}|(a, b)$.

イロト 不得 トイヨト イヨト ニヨー

Sac

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Repeating

 $\begin{aligned} r_{-1} &= r_0 q_1 + r_1, \quad 0 < r_1 \le r_0, \\ r_0 &= r_1 q_2 + r_2, \quad 0 < r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, \quad 0 < r_3 < r_2, \\ & \dots \\ r_{s-3} &= r_{s-2} q_{s-1} + r_{s-1}, \quad 0 < r_{s-1} < r_{s-2}, \\ r_{s-2} &= r_{s-1} q_s. \end{aligned}$

- Euclid proved that $(a, b) = r_{s-1}$.
- First of all (a, b)|a and (a, b)|b, and so $(a, b)|r_1$.
- Repeating this we get $(a, b)|r_j$ for $j = 2, 3, \ldots, s 1$.
- On the other hand, starting at the bottom line $r_{s-1}|r_{s-2}$, $r_{s-1}|r_{s-3}$ and so on until we have $r_{s-1}|b$ and $r_{s-1}|a$. Recall that this means that $r_{s-1}|(a, b)$.
- Thus we have just proved that

$$|r_{s-1}|(a,b), (a,b)|r_{s-1}, r_{s-1} = (a,b)$$

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Consider.

Example 1

Let a = 10678, b = 42

$$10678 = 42 \times 254 + 10$$

$$42 = 10 \times 4 + 2$$

$$10 = 2 \times 5.$$

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ - 三 - のへぐ

Thus (10678, 42) = 2.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Consider.

Example 1

Let a = 10678, b = 42

 $10678 = 42 \times 254 + 10$ $42 = 10 \times 4 + 2$ $10 = 2 \times 5.$

Thus (10678, 42) = 2.

• But how to compute the x and y in (a, b) = ax + by?

▲ロト ▲周ト ▲ヨト ▲ヨト ヨー のくで

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Consider.

Example 1

Let a = 10678, b = 42

 $10678 = 42 \times 254 + 10$ $42 = 10 \times 4 + 2$ $10 = 2 \times 5.$

Thus (10678, 42) = 2.

- But how to compute the x and y in (a, b) = ax + by?
- We could just work backwards through the algorithm using back substitution,

$$2 = 42 - 10 \times 4 = 42 - (10678 - 42 \times 254) \times 4$$

= 42 \times 1017 - 10678 \times 4.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Consider.

Example 1

Let a = 10678, b = 42

 $10678 = 42 \times 254 + 10$ $42 = 10 \times 4 + 2$ $10 = 2 \times 5.$

Thus (10678, 42) = 2.

- But how to compute the x and y in (a, b) = ax + by?
- We could just work backwards through the algorithm using back substitution,

$$2 = 42 - 10 \times 4 = 42 - (10678 - 42 \times 254) \times 4$$

Sar

 $= 42 \times 1017 - 10678 \times 4.$

 In general this is tedious and computationally wasteful since it requires all our calculations to be stored.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • A simpler way is as follows.

◆□▶ ◆□▶ ◆豆▶ ◆豆▶

Ξ 9 Q (P

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- A simpler way is as follows.
- Define $x_{-1} = 1$, $y_{-1} = 0$, $x_0 = 0$, $y_0 = 1$ and then lay the calculations out as follows.

 $\begin{array}{ll} r_{-1} = r_0 q_1 + r_1, & x_1 = x_{-1} - q_1 x_0, & y_1 = y_{-1} - q_1 y_0 \\ r_0 = r_1 q_2 + r_2, & x_2 = x_0 - q_2 x_1, & y_2 = y_0 - q_2 y_1 \\ r_1 = r_2 q_3 + r_3, & x_3 = x_1 - q_3 x_2, & y_3 = y_1 - q_3 y_2 \\ \vdots & \vdots & \vdots \\ r_{s-2} = r_{s-1} q_s. \end{array}$

▲ロト ▲周ト ▲ヨト ▲ヨト ヨー のくで

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- A simpler way is as follows.
- Define $x_{-1} = 1$, $y_{-1} = 0$, $x_0 = 0$, $y_0 = 1$ and then lay the calculations out as follows.

 $\begin{array}{ll} r_{-1} = r_0 q_1 + r_1, & x_1 = x_{-1} - q_1 x_0, & y_1 = y_{-1} - q_1 y_0 \\ r_0 = r_1 q_2 + r_2, & x_2 = x_0 - q_2 x_1, & y_2 = y_0 - q_2 y_1 \\ r_1 = r_2 q_3 + r_3, & x_3 = x_1 - q_3 x_2, & y_3 = y_1 - q_3 y_2 \\ \vdots & \vdots & \vdots \\ r_{s-2} = r_{s-1} q_s. \end{array}$

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

• The claim is that $x = x_{s-1}$, $y = y_{s-1}$. More generally $r_j = ax_j + by_j$ and this can be proved by induction.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- A simpler way is as follows.
- Define $x_{-1} = 1$, $y_{-1} = 0$, $x_0 = 0$, $y_0 = 1$ and then lay the calculations out as follows.

 $\begin{array}{ll} r_{-1} = r_0 q_1 + r_1, & x_1 = x_{-1} - q_1 x_0, & y_1 = y_{-1} - q_1 y_0 \\ r_0 = r_1 q_2 + r_2, & x_2 = x_0 - q_2 x_1, & y_2 = y_0 - q_2 y_1 \\ r_1 = r_2 q_3 + r_3, & x_3 = x_1 - q_3 x_2, & y_3 = y_1 - q_3 y_2 \\ \vdots & \vdots & \vdots \\ r_{s-2} = r_{s-1} q_s. \end{array}$

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

- The claim is that $x = x_{s-1}$, $y = y_{s-1}$. More generally $r_j = ax_j + by_j$ and this can be proved by induction.
- By construction we have $r_{-1} = ax_{-1} + by_{-1}$, $r_0 = ax_0 + by_0$.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- A simpler way is as follows.
- Define $x_{-1} = 1$, $y_{-1} = 0$, $x_0 = 0$, $y_0 = 1$ and then lay the calculations out as follows.

 $\begin{array}{ll} r_{-1} = r_0 q_1 + r_1, & x_1 = x_{-1} - q_1 x_0, & y_1 = y_{-1} - q_1 y_0 \\ r_0 = r_1 q_2 + r_2, & x_2 = x_0 - q_2 x_1, & y_2 = y_0 - q_2 y_1 \\ r_1 = r_2 q_3 + r_3, & x_3 = x_1 - q_3 x_2, & y_3 = y_1 - q_3 y_2 \\ \vdots & \vdots & \vdots \\ r_{s-2} = r_{s-1} q_s. \end{array}$

- The claim is that $x = x_{s-1}$, $y = y_{s-1}$. More generally $r_j = ax_j + by_j$ and this can be proved by induction.
- By construction we have $r_{-1} = ax_{-1} + by_{-1}$, $r_0 = ax_0 + by_0$.
- Suppose $r_j = ax_j + by_j$ is established for all $j \le k$. Then

$$r_{k+1} = r_{k-1} - q_{k+1}r_k$$

= $(ax_{k-1} + by_{k-1}) - q_{k+1}(ax_k + by_k)$
= $ax_{k+1} + by_{k+1}$.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- A simpler way is as follows.
- Define $x_{-1} = 1$, $y_{-1} = 0$, $x_0 = 0$, $y_0 = 1$ and then lay the calculations out as follows.

 $\begin{array}{ll} r_{-1} = r_0 q_1 + r_1, & x_1 = x_{-1} - q_1 x_0, & y_1 = y_{-1} - q_1 y_0 \\ r_0 = r_1 q_2 + r_2, & x_2 = x_0 - q_2 x_1, & y_2 = y_0 - q_2 y_1 \\ r_1 = r_2 q_3 + r_3, & x_3 = x_1 - q_3 x_2, & y_3 = y_1 - q_3 y_2 \\ \vdots & \vdots & \vdots \\ r_{s-2} = r_{s-1} q_s. \end{array}$

- The claim is that $x = x_{s-1}$, $y = y_{s-1}$. More generally $r_j = ax_j + by_j$ and this can be proved by induction.
- By construction we have $r_{-1} = ax_{-1} + by_{-1}$, $r_0 = ax_0 + by_0$.
- Suppose $r_j = ax_j + by_j$ is established for all $j \le k$. Then

$$\begin{aligned} r_{k+1} &= r_{k-1} - q_{k+1}r_k \\ &= (ax_{k-1} + by_{k-1}) - q_{k+1}(ax_k + by_k) \\ &= ax_{k+1} + by_{k+1}. \end{aligned}$$

• In particular $(a, b) = r_{s-1} = ax_{s-1} + by_{s-1} + by_{s-1$

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Hence laying out the example above in this expanded form we have

$$\begin{aligned} r_{-1} &= 10678, \ r_0 = 42, \ x_{-1} = 1, \ x_0 = 0, \ y_{-1} = 0, \ y_0 = 1, \\ 10678 &= 42 \cdot 254 + 10, \quad x_1 = 1, \quad y_1 = -254 \\ 42 &= 10 \cdot 4 + 2, \quad x_2 = -4, \quad y_2 = 1017 \\ 10 &= 2 \cdot 5. \end{aligned}$$

 $(10678, 42) = 2 = 10678 \cdot (-4) + 42 \cdot (1017).$

*ロ * * @ * * ミ * ミ * ・ ミ * の < @

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Hence laying out the example above in this expanded form we have

$$r_{-1} = 10678, r_0 = 42, x_{-1} = 1, x_0 = 0, y_{-1} = 0, y_0 = 1,$$

 $10678 = 42 \cdot 254 + 10, x_1 = 1, y_1 = -254$

$$42 = 10 \cdot 4 + 2, \qquad x_2 = -4, \quad y_2 = 1017 \\ 10 = 2 \cdot 5.$$

 $(10678, 42) = 2 = 10678 \cdot (-4) + 42 \cdot (1017).$

*ロ * * @ * * ミ * ミ * ・ ミ * の < @

• It is also possible to set this up using matrices.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Lay out the sequences in rows

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Lay out the sequences in rows

• Now proceed to compute each successive row as follows.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Lay out the sequences in rows

- Now proceed to compute each successive row as follows.
- If the s-th row is the last one to be computed, calculate $q_s = \lfloor r_{s-1}/r_s \rfloor$.

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ◆ ●

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • Lay out the sequences in rows

- Now proceed to compute each successive row as follows.
- If the s-th row is the last one to be computed, calculate $q_s = \lfloor r_{s-1}/r_s \rfloor$.
- Then take the last two rows computed and pre multiply by $(1, -q_s)$

$$\begin{pmatrix} (1, -q_s) \begin{pmatrix} r_{s-1}, & x_{s-1}, & y_{s-1} \\ r_s, & x_s, & y_s \end{pmatrix} = (r_{s+1}, x_{s+1}, y_{s+1})$$

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ◆ ●

to obtain the s + 1-st row.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• Here is a simple example.

Example 2

Let a = 4343, b = 973. We can lay this out as follows

	4343	1	0	
4	973	0	1	
2	451	1	-4	
6	71	-2	9	
2	25	13	-58	
1	21	-28	125	
5	4	41	-183	
	1	-233	1040	
Thus $(4343, 973) = 1$	=(-23)	33)4343	+ (1040)9	73.

・ロト ・ 同ト ・ ヨト ・ ヨト

= √Q (~

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • We can use Euclid's algorithm to find the complete solution in integers to linear diophantine equations of the kind

$$ax + by = c$$
.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • We can use Euclid's algorithm to find the complete solution in integers to linear diophantine equations of the kind

$$ax + by = c.$$

• Here *a*, *b*, *c* are integers and we wish to find all integers *x* and *y* which satisfy this.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • We can use Euclid's algorithm to find the complete solution in integers to linear diophantine equations of the kind

$$ax + by = c.$$

• Here *a*, *b*, *c* are integers and we wish to find all integers *x* and *y* which satisfy this.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

• There are some obvious necessary conditions.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • We can use Euclid's algorithm to find the complete solution in integers to linear diophantine equations of the kind

$$ax + by = c.$$

- Here *a*, *b*, *c* are integers and we wish to find all integers *x* and *y* which satisfy this.
- There are some obvious necessary conditions.
- First of all if a = b = 0, then it is not soluble unless c = 0 and then it is soluble by any x and y, which is not very interesting.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • We can use Euclid's algorithm to find the complete solution in integers to linear diophantine equations of the kind

$$ax + by = c.$$

- Here *a*, *b*, *c* are integers and we wish to find all integers *x* and *y* which satisfy this.
- There are some obvious necessary conditions.
- First of all if a = b = 0, then it is not soluble unless c = 0 and then it is soluble by any x and y, which is not very interesting.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

• Thus it makes sense to suppose that one of *a* or *b* is non-zero.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • We can use Euclid's algorithm to find the complete solution in integers to linear diophantine equations of the kind

$$ax + by = c.$$

- Here *a*, *b*, *c* are integers and we wish to find all integers *x* and *y* which satisfy this.
- There are some obvious necessary conditions.
- First of all if a = b = 0, then it is not soluble unless c = 0 and then it is soluble by any x and y, which is not very interesting.
- Thus it makes sense to suppose that one of *a* or *b* is non-zero.
- Then since (a, b) divides the left hand side, we can only have solutions if (a, b)|c.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations • We are considering ax + by = c and we are assuming that a and b are not both 0 and (a, b)|c.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We are considering ax + by = c and we are assuming that a and b are not both 0 and (a, b)|c.
- If we choose x and y so that ax + by = (a, b), then we have

$$a(xc/(a,b)) + b(yc/(a,b)) = (ax + by)c/(a,b) = c$$

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

so we certainly have a solution of our equation.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We are considering ax + by = c and we are assuming that a and b are not both 0 and (a, b)|c.
- If we choose x and y so that ax + by = (a, b), then we have

a(xc/(a,b)) + b(yc/(a,b)) = (ax + by)c/(a,b) = c

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ◆ ●

so we certainly have a solution of our equation.

• Call it x₀, y₀.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We are considering ax + by = c and we are assuming that a and b are not both 0 and (a, b)|c.
- If we choose x and y so that ax + by = (a, b), then we have

a(xc/(a,b)) + b(yc/(a,b)) = (ax + by)c/(a,b) = c

so we certainly have a solution of our equation.

- Call it *x*₀, *y*₀.
- Now consider any other solution. Then

$$ax + by - ax_0 - by_0 = c - c = 0, \ a(x - x_0) = b(y_0 - y).$$

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ◆ ●

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We are considering ax + by = c and we are assuming that a and b are not both 0 and (a, b)|c.
- If we choose x and y so that ax + by = (a, b), then we have

a(xc/(a,b)) + b(yc/(a,b)) = (ax + by)c/(a,b) = c

so we certainly have a solution of our equation.

- Call it *x*₀, *y*₀.
- Now consider any other solution. Then

 $ax + by - ax_0 - by_0 = c - c = 0, \ a(x - x_0) = b(y_0 - y).$

• Hence

$$\frac{a}{(a,b)}(x-x_0) = \frac{b}{(a,b)}(y_0 - y).$$

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ◆ ●

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We are considering ax + by = c and we are assuming that a and b are not both 0 and (a, b)|c.
- If we choose x and y so that ax + by = (a, b), then we have

a(xc/(a,b)) + b(yc/(a,b)) = (ax + by)c/(a,b) = c

so we certainly have a solution of our equation.

- Call it *x*₀, *y*₀.
- Now consider any other solution. Then

 $ax + by - ax_0 - by_0 = c - c = 0, \ a(x - x_0) = b(y_0 - y).$

Hence

$$\frac{a}{(a,b)}(x-x_0) = \frac{b}{(a,b)}(y_0 - y).$$

• Then as $\left(\frac{a}{(a,b)}, \frac{b}{(a,b)}\right) = 1$ we have by an earlier example that $y_0 - y = z \frac{a}{(a,b)}$ and $x - x_0 = z \frac{b}{(a,b)}$ for some z.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

- We are considering ax + by = c and we are assuming that a and b are not both 0 and (a, b)|c.
- If we choose x and y so that ax + by = (a, b), then we have

a(xc/(a,b)) + b(yc/(a,b)) = (ax + by)c/(a,b) = c

so we certainly have a solution of our equation.

- Call it *x*₀, *y*₀.
- Now consider any other solution. Then

 $ax + by - ax_0 - by_0 = c - c = 0, \ a(x - x_0) = b(y_0 - y).$

Hence

$$\frac{a}{(a,b)}(x-x_0) = \frac{b}{(a,b)}(y_0-y).$$

- Then as $\left(\frac{a}{(a,b)},\frac{b}{(a,b)}\right) = 1$ we have by an earlier example
 - that $y_0 y = z \frac{a}{(a,b)}$ and $x x_0 = z \frac{b}{(a,b)}$ for some z.
- But any x and y of this form give a solution, so we have found the complete solution set.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• We have

Theorem 3

Suppose that a and b are not both 0 and (a, b)|c. Suppose further that $ax_0 + by_0 = c$. Then every solution of

ax + by = c

is given by

$$x = x_0 + z \frac{b}{(a,b)}, \quad y = y_0 - z \frac{a}{(a,b)}$$

▲ロト ▲周ト ▲ヨト ▲ヨト ヨー のくで

where z is any integer.

> Robert C. Vaughan

Euclid's algorithm

Linear Diophantine Equations

• We have

Theorem 3

Suppose that a and b are not both 0 and (a, b)|c. Suppose further that $ax_0 + by_0 = c$. Then every solution of

ax + by = c

is given by

$$x = x_0 + z \frac{b}{(a,b)}, \quad y = y_0 - z \frac{a}{(a,b)}$$

where z is any integer.

• One can see here that the solutions x all leave the same remainder on division by $\frac{b}{(a,b)}$ and likewise for y on division by $\frac{a}{(a,b)}$. This suggests that there may be a useful way of classifying integers.