

# MATH 25 CLASS 11 NOTES, OCT 14 2011

## CONTENTS

- |                             |   |
|-----------------------------|---|
| 1. More on congruences      | 1 |
| 2. Linear equations mod $n$ | 2 |

### 1. MORE ON CONGRUENCES

It is often useful to know that a single congruence mod  $n$  can be split up into several congruences mod prime powers, and vice versa, multiple congruences mod various relatively prime numbers can be assembled into one congruence. This is the content of the following proposition:

**Proposition 1** (Theorem 3.4 of the text). *Let  $n = p_1^{e_1} \dots p_k^{e_k}$  be the factorization of an integer  $n$ . Then  $a \equiv b \pmod{n}$  if and only if  $a \equiv b \pmod{p_i^{e_i}}$  for every  $i$ . More generally, if  $n = n_1 n_2$ , where  $n_1, n_2$  are relatively prime, then  $a \equiv b \pmod{n}$  if and only if  $a \equiv b \pmod{n_i}$  for  $i = 1, 2$ .*

*Proof.* We have basically already proved this proposition, in a slightly different language.  $a \equiv b \pmod{n}$  is true if and only if  $n|(b-a)$ , and true if and only if  $p_i^{e_i} | (b-a)$  for every  $i$ . We are using the fact that the  $p_i^{e_i}$  are relatively prime in a critical way when we go from  $p_i^{e_i} | (b-a)$  for all  $i$  to  $n|(b-a)$ ; this is the content of Corollary 1.11a of the text. The proof of the generalization is identical.  $\square$

Another fact which is obvious from what we have already proven is the following:

**Lemma 1** (Lemma 3.5 of the text). *Let  $f(x)$  be a polynomial with integer coefficients. If  $a \equiv b \pmod{n}$ , then  $f(a) \equiv f(b) \pmod{n}$ .*

*Proof.* If  $f(x) = a_n x^n + \dots + a_0$ , then  $f(a) = a_n a^n + \dots + a_0 \equiv a_n b^n + \dots + a_0 = f(b) \pmod{n}$ , because addition and multiplication are preserved mod  $n$ .  $\square$

The reason these two results are interesting is because we will think about trying to solve  $f(x) \equiv 0 \pmod{n}$ , where  $f(x)$  is some polynomial with integer coefficients. The first proposition allows us to reduce this to the problem of understanding the solutions to  $f(x) \equiv 0 \pmod{p^e}$ , for various prime powers  $p^e$ .

An interesting application of the previous lemma is a nice proof that no single-variable polynomial only takes on prime values:

**Proposition 2** (Theorem 3.6 of the text). *Let  $f(x)$  be a polynomial with integer coefficients which is non-constant. Then there is some integer  $x$  for which  $f(x)$  is not prime.*

*Proof.* Suppose  $f(x)$  were prime for every integer  $x$ . Select some integer, say  $a$ . Then  $f(a) = p$ . Now take any other  $b \equiv a \pmod{p}$ , where  $b \neq a$ . Then  $f(b) \equiv f(a) \equiv 0 \pmod{p}$ . Since  $p|f(b)$ , and  $f(b)$  is prime, then  $p = f(b)$ . This is true for every

$b \equiv a \pmod{p}$ , so the polynomial  $f(x) - p = 0$  has infinitely many roots. However, a polynomial with infinitely many roots must be the zero polynomial, so  $f(x) = p$  is a constant polynomial.  $\square$

By way of general knowledge: we know that for linear polynomials  $ax + b$ , where  $\gcd(a, b) = 1$ , there are infinitely many prime values of this polynomial, and also some non-prime values. Whether the same is true for even a quadratic polynomial is still an open question; for instance, it is unknown whether  $x^2 + 1$  takes on infinitely many prime values. Interestingly enough, there are explicit examples of multivariable polynomial which, when they take positive values, are always prime (and every prime is in the range of such a polynomial). The reason these polynomials do not work as practical prime generators is because it is hard to determine what values the variables should take to force the value of the polynomial to be positive and large.

## 2. LINEAR EQUATIONS MOD $n$

We will now systematically study the equation  $ax \equiv b \pmod{n}$ , to determine whether this equation has solutions mod  $n$ , and if so, how many there are and how to find all of them. Of course, equations mod  $n$  are finite problems, so we could simply solve this equation using brute force, by plugging in each of the  $n$  possible values of  $x \pmod{n}$ . However, this approach is not wholly satisfactory, since we want to not only know how to solve this equation, but how to determine properties of this equation without actually solving anything using brute force.

Fortunately, we've already done most of the work in understanding these equations, just in a slightly different language. The proof of the following proposition, which says a lot of what there is to be said, demonstrates what I mean.

**Proposition 3** (Theorem 3.7 of the text). *Consider the congruence  $ax \equiv b \pmod{n}$ . Let  $d = \gcd(a, n)$ . Then this equation has a solution if and only if  $d$  divides  $b$ . If so, then there are  $d$  solutions mod  $n$ . If  $x_0 \pmod{n}$  is one of those solutions, then the remaining are given by  $x_0 + \frac{nt}{d} \pmod{n}$ , where  $t$  ranges over the integers.*

*Proof.* The key idea is to recognize that  $ax \equiv b \pmod{n}$  is a statement about a linear equation in two variables. In particular,  $ax \equiv b \pmod{n}$  if and only if  $n \mid (ax - b)$ , which is true if and only if there is an integer  $y$  such that  $ny = ax - b$ , or  $ax + ny = b$ . This is an equation which we know how to solve: it has solutions if and only if  $\gcd(a, n) = d \mid b$ , which was the first part of the proposition, and if there are solutions, the  $x$ -coordinates are given by the equation  $x = x_0 + nt/d$ , with  $t \in \mathbb{Z}$ .

To really conclude the proof, we should check that  $x_0 + nt/d \pmod{n}$  yields  $d$  congruence classes mod  $n$ . This is more or less clear, because  $nt/d$  takes on distinct congruence classes mod  $n$  when  $t = 0, 1, \dots, d-1$ , and then  $x_0 + nd/d = x_0 + n \equiv x_0 \pmod{n}$ , so that the congruence classes repeat when  $t = d$ .  $\square$

An alternate way of expressing the last claim of the proposition is that the solutions to  $ax \equiv b \pmod{n}$  not only form  $d$  congruence classes mod  $n$ , but a single congruence class mod  $n/d$ , with  $x \equiv x_0 \pmod{n/d}$ . As we will see in the near future, this point of view sometimes has its advantages.

**Examples.**

- Find all solutions to  $6x \equiv 4 \pmod{10}$ . Since  $\gcd(6, 10) = 2$ , and  $2|4$ , this equation has solutions, and has two solutions mod 10, or one solution mod 5. In this example, we simply find the solutions by brute force. A bit of observations shows that  $x = 4, 9$  both solve this equation, so the solutions are given by  $x \equiv 4, 9 \pmod{10}$ , or  $x \equiv 4 \pmod{5}$ . Notice that  $x \equiv 9 \pmod{5}$  also works.
- Find all solutions to  $5x \equiv 7 \pmod{13}$ . Since  $\gcd(5, 13) = 1$ , and  $1|7$ , this means there is exactly one solution to this equation mod 13. Again, brute force shows that  $x \equiv 4 \pmod{13}$  is a solution, and therefore the only solution.
- Find all solutions to  $94x \equiv 1 \pmod{273}$ . We used this example a few weeks ago when discussing the Euclidean algorithm and found that  $\gcd(94, 273) = 1$ . Therefore, there is exactly one solution to this equation mod 273. How do we go about finding this solution? Instead of using brute force, remember that the Euclidean algorithm tells us that  $1 = 273(-21) + 94(61)$ . Another way of writing this is  $94(61) \equiv 1 \pmod{273}$ , so  $x \equiv 61 \pmod{273}$  is the only solution mod 273 to the above equation. This is much better than the original naive approach we discussed, where we use brute force to try all 273 possible values of  $x$ !
- Find all solutions to  $ax \equiv 1 \pmod{n}$ , where  $a$  is any integer. This is a routine application of the above proposition; if  $\gcd(a, n) = 1$ , then there is exactly one solution mod  $n$ , and if not, then there are no solutions. To actually find a solution if  $\gcd(a, n) = 1$ , one can use the Euclidean algorithm to solve  $ax + by = 1$ , and then take the  $x$ -coordinate of whatever solution you find.

One nice interpretation of the last example is that it makes sense of when we are allowed to ‘divide by  $a$ ’ in a congruence equation. For instance, suppose we have an equation like  $a \equiv b \pmod{n}$ . If we try to naively ‘divide by  $a$ ’, we end up with something which looks like  $1 \equiv b/a \pmod{n}$ . As written, this makes no sense, since in general  $b/a$  is not an integer. However, if  $ax \equiv 1 \pmod{n}$  has a solution, which we might call  $a^{-1}$ , then multiplying  $a \equiv b \pmod{n}$  by  $a^{-1}$  yields  $aa^{-1} \equiv ba^{-1} \pmod{n}$ , or  $1 \equiv ba^{-1} \pmod{n}$ . As  $ba^{-1}$  is still an integer, this makes sense, and has the same effect on the left hand side as naively trying to ‘divide by  $a$ ’ does. If this  $a^{-1} \pmod{n}$  exists (ie, if  $\gcd(a, n) = 1$ ), we sometimes call  $a^{-1}$  the *multiplicative inverse* of  $a \pmod{n}$ .

**Example.** Suppose we already figured out that  $7x \equiv 1 \pmod{13}$  has solution  $x \equiv 2 \pmod{13}$ . We can use this information to find all the solutions to  $7x \equiv a \pmod{13}$ ; multiply both sides by 2 to get  $14x \equiv 2a \pmod{13}$ , or  $x \equiv 2a \pmod{13}$ . So if  $\gcd(a, n) = 1$ , we can multiply by the multiplicative inverse of  $a$  to eliminate the coefficient  $a$  from  $ax$ , much like what we would do if we were simply solving a linear equation over rationals or real numbers.

The book gives a different algorithm for solving  $ax \equiv b \pmod{n}$ . I’m not as big a fan of it, since the Euclidean algorithm does the job nicely, but the book’s method does use the following proposition which can be helpful for simplifying the process of finding a solution to  $ax \equiv b \pmod{n}$ .

**Proposition 4** (Lemma 3.9 of the text). . Consider the equation  $ax \equiv b \pmod n$ .  
 (a) If  $m|a, b, n$ , and  $a' = a/m, b' = b/m, n' = n/m$ , then  $ax \equiv b \pmod n$  if and only if  $a'x \equiv b' \pmod{n'}$ . (b) If  $\gcd(a, n) = 1$ , and  $m|a, b$ , and  $a' = a/m, b' = b/m$ , then  $ax \equiv b \pmod n$  if and only if  $a'x \equiv b' \pmod n$ .

*Proof.* (a)  $ax \equiv b \pmod n$  if and only if  $n|(ax - b)$  if and only if  $ny = (ax - b)$  for some integer  $y$ . Since  $m|a, b, n$ , this is true if and only if  $\frac{n}{m}y = \frac{a}{m}x - \frac{b}{m}$ , which is true if and only if  $n'y = a'x - b'$ . This in turn is true if and only if  $a'x \equiv b' \pmod{n'}$ .

(b) This time,  $ax \equiv b \pmod n$  if and only if  $ny = ax - b$ . Divide both sides by  $m$ ; we have  $ny/m = a'x - b'$ . Since the right hand side is an integer,  $ny/m$  is an integer. Since  $m|a$  and  $\gcd(a, n) = 1$ , we must have  $\gcd(m, n) = 1$ , so  $y/m$  is an integer. This means  $n|(a'x - b')$ , or  $a'x \equiv b' \pmod n$ . Conversely, if  $a'x \equiv b' \pmod n$ , then  $ny = a'x - b'$  for some integer  $y$ , and multiplying by  $m$  gives  $n(my) = ax - b$ , or  $ax \equiv b \pmod n$ .  $\square$

If you use this proposition, be very careful to make sure that the hypotheses are met and that you are using the correct proposition.

**Example.** Solve  $24x \equiv 12 \pmod{66}$ . Since 24, 12, 66 are all divisible by 6, this is equivalent to  $4x \equiv 2 \pmod{11}$ . We can solve this by brute force, to get  $x \equiv 6 \pmod{11}$  as the only solution. If we wanted to, we could describe the solutions to the original equation mod 66; these would be  $x \equiv 6, 6 + 11, 6 + 22, 6 + 33, 6 + 44, 6 + 55 \pmod{66}$ .

What you are not allowed to do is convert  $24x \equiv 12 \pmod{66}$  to  $4x \equiv 3 \pmod{66}$ . Even though the latter equation makes sense, since all numbers are still integers, it has fewer solutions than the original congruence, since  $\gcd(4, 66) = 2$ , which is not the same as  $\gcd(24, 66) = 6$ . So be very mindful when you are actually dividing by integers to convert a congruence to something which is more easily solved.

In summary, you have a lot of ways to find solutions to  $ax \equiv b \pmod n$ . You can do a brute force search, testing all  $n$  possibilities for  $x$ , although this might be painful if  $n$  is fairly large. If you compute  $\gcd(a, n)$ , say using the Euclidean algorithm or any other method, then you only need to find one solution to  $ax \equiv b \pmod n$  before knowing what all the solutions look like. If you used the Euclidean algorithm to solve  $ax + ny = b$ , then you immediately can read off a solution to  $ax \equiv b \pmod n$  from this. If  $a, b, n$  all have a common divisor, you can also divide through by a common divisor to reduce  $ax \equiv b \pmod n$  to another linear congruence where all numbers involved are smaller, so that it becomes easier to find a solution.