Tonelli's Method for Square Roots

We will not have a proof that Lehmer's Algorithm 8.3 in Bressoud's book actually works. Therefore I give here a short explanation of Tonelli's Method based instead on primitive roots. This is an adaptation of the discussion of the approach in the book *Algorithmic Number Theory* by E. Bach and J. Shalit, MIT, 1996, page 156. In our proof of correctness of the method, we use the following central theorem.

Theorem (Gauss). *If p is a prime, then p admits a primitive root, i.e. an element of $\mathbb{Z}_p^\times$ of multiplicative order $p-1$.*

Surprisingly enough, primitive roots do not appear in the algorithm itself! They serve only as signposts in understanding what is going on.

There is no problem taking square roots modulo 2, so let p be an odd prime, say $p-1 = 2^s t$, t odd. Fix b, a quadratic non-residue modulo p, $1 < b \leq p-1$, and g, a primitive root modulo p.

Lemma. *Then for every number a, $1 \leq a < p$, there is a unique pair of integers (i, j), $0 \leq i < 2^s$, $0 \leq j < t$, such that $a \equiv b^i \cdot g^{j\cdot 2^s} \mod p$.*

Proof: To see this, notice that the products on the right give $2^s t = p-1$ integers not divisible by p. If two of them were congruent modulo p, then cancelling the smaller powers of b and g^{2^s} from both sides would give a congruence of the form

$$b^i \equiv g^{j\cdot 2^s} \mod p, \quad i, j \geq 0, \quad i < 2^s, j < t.$$

Write $b = g^l \mod p$, with l odd, so that $il \equiv j \cdot 2^s \mod 2^s t$, as $p-1 = 2^s t$ is the order of g. But l is odd, $i < 2^s$, and $il \equiv j \cdot 2^s \equiv 0 \mod 2^s$. Thus $i = 0$. Now $t \mid j < t$, so $j = 0$ as well. In other words, the orginal pairs of exponents (i, j) were identical. □

Now if we are to have any chance of finding a square root for a modulo p, a must be a quadratic residue modulo p: $(a/p) = 1$. In particular, i must be even in the above representation, and the main part of the algorithm is dedicated to determining i such that

- $i < 2^s$ is even and
- $(a/b^i)^t \equiv 1 \mod p$.

.

Since $i < 2^s$, by our Lemma, $a = b^i g^{j\cdot 2^s}$, $j < t$, odd. Then

$$\left(g^{j\cdot 2^s}\right)^{t+1} = \left(g^{j\cdot 2^s}\right)^t \cdot g^{j\cdot 2^s} = \left(g^{2^s\cdot t}\right)^j \cdot g^{j\cdot 2^s} \equiv g^{j\cdot 2^s} \mod p.$$

Therefore, since t is odd, $t+1$ is even, as is i. So

$$(b^{i/2} \cdot g^{j\cdot 2^s(t+1)/2})^2 = b^i g^{j\cdot 2^s(t+1)} = (g^{2^s\cdot t}) b^i g^{j\cdot 2^s} \equiv a \mod p.$$

That is, we have found a square root of a modulo p. For convenience in the algorithm, we write this as

$$\sqrt{a} = b^{i/2} \cdot (a/b^i)^{(t+1)/2} \operatorname{Mod} p.$$

Now we turn our attention to the determination of i, which is carried out recursively so that $i = i_s$ below.

Lemma. *The congruences*

$$\text{(Cong)} \qquad \left(a/b^{i_k}\right)^{2^{s-k}t} \equiv 1 \mod p, \quad k = 1, \ldots, s,$$

have a solution with $i_1 = 2$ *and* $i_{k+1} \equiv i_k \mod 2^k$, $k = 1, \ldots, s-1$.

Proof. (Base Case) As $i_1 = 2$ and a is a quadratic residue, $(a/b^{i_1})^{2^{s-1}t} \equiv a^{2^{s-1}t}/b^{p-1} \equiv 1 \mod p$, by Euler's Criterion and Fermat's Little Theorem. So the congruence (Cong) is satisfied for $k = 1$.

(Induction Step) Assume the claimed congruence satisfied for k, $1 \leq k < s$. But

$$(a/b^{i_k})^{2^{s-k}t} \equiv 1 \mod p \quad \Longrightarrow \quad (a/b^{i_k})^{2^{s-k-1}t} \equiv \pm 1 \mod p,$$

as $x^2 \equiv 1 \mod p$ has only the solutions ± 1 modulo p.

Moreover, as b is a quadratic non-residue modulo p, by Euler's Criterion,

$$b^{(p-1)/2} = b^{2^{s-1}t} \equiv -1 \mod p.$$

Therefore

$$(a/b^{i_k+2^k})^{2^{s-k-1}t} \equiv -(a/b^{i_k})^{2^{s-k-1}t} \mod p.$$

Consequently, exactly one of the choices

$$i_{k+1} = i_k \quad \text{or} \quad i_{k+1} = i_k + 2^k$$

will satisfy the congruence for $k+1$:

$$(a/b^{i_{k+1}})^{2^{s-k-1}t} \equiv 1 \mod p.$$

□

Summary of Tonelli's Method for Computation of Square Roots Mod p.

0. Verify that $(a/p) = 1$.
1. Write $p - 1 = 2^s t$, t odd.
2. Find b s.t. $(b/p) = -1$.
3. Set $i_1 = 2$ and recursively determine $i_k = i_{k-1}$ or $i_{k-1} + 2^{k-1}$, $k = 2, \ldots, s$, s.t.

$$\left(a/b^{i_k}\right)^{2^{s-k}t} \equiv 1 \mod p.$$

4. With $i := i_s$, set

$$\sqrt{a} := b^{i/2} \cdot (a/b^i)^{(t+1)/2} \operatorname{Mod} p;$$

the other square root is $p - \sqrt{a}$.

Remarks: Now Tonelli's Method can be thought of in two ways. If we just search with $b = 2, 3, \ldots$, it is deterministic, but with no certainty of how long the search will take for an unspecified p. If we choose b randomly, then the likelihood is 50% that the first choice will work. So a random search finds a quadratic nonresidue quickly with high probability. Having chosen b, Tonelli's algorithm finishes off the job in $\mathcal{O}((\log p)^{2+\log_2 3})$ bit operations.

W. Dale Brownawell
21 August 2015