McKee's method

In this adaptation of Euler's method one uses a large value of d.

Step 1 Check that n is not a square, or a higher power. If it is we have found a factor. Otherwise choose $x_0 = \lfloor \sqrt{n - n^{2/3}} \rfloor$ and $d = n - x_0^2$. Then $n = x_0^2 + d$, with d approximately $n^{2/3}$.

Step 2 Check n for factors up to $(4d/3)^{1/4} = O(n^{1/6})$, using trial division; and stop if you find one.

Step 3 For each integer a in the interval $[\sqrt{d/12}, \sqrt{4d/3}]$, search for solutions to $an = x^2 + dy^2$ with $x, y \in \mathbb{N}$ and $y^2 \neq a$. It can be proved (though we shall not do it here) that if n is composite, and the algorithm reaches this stage, then there always will be such a solution.

For good procedures to search for solutions x, y see below.

Step 4 Having found solutions $n = x_0^2 + d$, $an = x_1^2 + dy_1^2$ it follows that $n \mid (y_1 x_0)^2 - x_1^2$. It is then not hard to check that $\gcd(n, y_1 x_0 - x_1)$ is a non-trivial factor of n.

One can show that a suitable implementation of this will run in $O(n^{1/3+\varepsilon})$ steps, for any small fixed $\varepsilon > 0$. In practice it is better to take d somewhat smaller than $n^{2/3}$. With d around $n^{1/2}$ one gets an algorithm with theoretical running time $O(n^{3/8+\varepsilon})$, which soon beats trial division.

To search for solutions of $an = x^2 + dy^2$ in Step 3 one could first factor d via trial division, and find a quadratic non-residue for each prime factor p. The algorithm of §1.11 may then be used to solve $x^2 = an (\mathrm{mod}\ p)$ for each such p, and then one can combine the solutions using the Chinese Remainder

Theorem. All this becomes far easier if we can choose d to be prime in the first place.

Example
Take $n = 1082154235955237$

Step 1 $n = 32896084^2 + 1893420181 = x_0^2 + d$, say, where in fact d is prime.
Step 2 Trial division up to 233 is a relatively tiny amount of work, and finds no prime divisors.
Step 3 Search over values for a between 12562 and 50245, finding that

$$43036n = 2591866961^2 + d.145101^2.$$

Note that with these values of d and a the congruence $x^2 = an (\text{mod } d)$ has only 7 solutions which need checking, in the range $x < \sqrt{an}$.
Step 4 $n = 12345701 \times 87654337$.

Modern speed-ups of Fermat's method

Instead of looking for solutions of $x^2 - y^2 = n$ one can try $x^2 - ny^2 = z^2$, in effect making n a difference of two rational squares, rather than integer ones. There are variants in which one examines $x^2 - any^2 = z^2$ for some small a, on the basis that if one finds prime factor of an which is suitably large (we hope), this will divide n.

There are two approaches in the literature, one is to make $x^2 - ny^2$ small, in the hope that this will increase the chance of $x^2 - ny^2$ being square. This leads to the Continued Fraction Method, and to SQUFOF (the SQUare FOrm Factoring method). The second method keeps both x and y small, and is due to McKee (1999).

These methods are expected to run in time $O(n^{1/4+\varepsilon})$, though we currently cannot prove this. In practice they are soon better than trial division.

The Continued Fraction and SQUFOF methods

Take x/y to be a continued fraction approximation to $\sqrt{n}$. Then $|x/y - \sqrt{n}| < y^{-2}$, so that

$$|x/y + \sqrt{n}| < y^{-2} + 2\sqrt{n} \leq 1 + 2\sqrt{n}.$$

Hence

$$|x^2/y^2 - n| < (1 + 2\sqrt{n})/y^2,$$

giving us $|x^2 - ny^2| < 1 + 2\sqrt{n}$. Thus the chance that $x^2 - ny^2$ is a square is $O(n^{-1/4})$. So one runs the continued fraction algorithm for $\sqrt{n}$, keeping track only of the value of $x (\mathrm{mod}\ n)$. (We have $x = p_k$ in the notation of §§4–8, and the numbers p_k grow exponentially; but fortunately it is enough to work with $x (\mathrm{mod}\ n)$.) If $x = x' (\mathrm{mod}\ n)$ with $0 \le x' < n$ then we have $x'^2 = r (\mathrm{mod}\ n)$ with a remainder r which is guaranteed to have $r < 1 + 2\sqrt{n}$. If we have $r = z^2$ then $\gcd(x' - z, n) = \gcd(x - z, n)$ will be a factor of n (if we are lucky).

SQUFOF (see Cohen for details) was an important algorithm in its day, and in effect runs through the continued fraction algorithm, but without computing x or y.

McKee's 1999 method

Suppose that n is odd and composite, and set $b = \lceil \sqrt{n} \rceil$. Define a quadratic form

$$Q(x, y) = (x + by)^2 - ny^2 = x^2 + 2bxy + (b^2 - n)y^2,$$

so that the coefficients are $O(\sqrt{n})$. We will seek integers x, y, z such that $Q(x, y) = z^2$, in the hope that $\gcd(x + by - z, n)$ is a non-trivial factor of n. Note that $y = 1$ corresponds to Fermat's method.
McKee's method relies on the following lemma.

Lemma
Suppose that $n = pq$ with $q > p > 2n^{1/4}$, and choose a positive integer $T \le n^{1/2} - n^{1/4}$. Then there are at least T distinct integer triples (x, y, z) such that $Q(x, y) = z$ and

(1) y is even and $2 \le y \le n^{1/4} + 2(T - 1)$,

(2) $y|x| < 2T^{1/4}\sqrt{n}$,

(3) $|z| < (T^2 - 1)\sqrt{n}$.

Moreover $\gcd(x + by - z, n)$ is a non-trivial factor of n in each case.

Proof
The proof consists of unexciting technical details, which you might want to

skip!
For each integer $t = 0, \ldots, T-1$ let

$$r = \lfloor \sqrt{q/p} \rfloor < \sqrt{q/p} \leq \sqrt{(n^{3/4}/2)/(2n^{1/4})} = n^{1/4}/2$$

and put $y = 2(r+t)$, so that (1) holds. Take

$$z = q - (r+t)^2 p \quad \text{and} \quad x = q + p(r+t)^2 - by$$

so that

$$Q(x,y) = (x+by)^2 - ny^2 = (q+p(r+t)^2)^2 - 4pq(r+t)^2 = (q-(p(r+t)^2)^2 = z^2.$$

Moreover

$$z \leq q - r^2 p \leq q - (\sqrt{q/p} - 1)^2 p = 2\sqrt{qp} - p < 2\sqrt{n}$$

and

$$\begin{aligned} -z \leq -q + (\sqrt{q/p} + T - 1)^2 p &= 2(T-1)\sqrt{qp} + (T-1)^2 p \\ &\leq 2(T-1)\sqrt{n} + (T-1)^2\sqrt{n} = (T^2-1)\sqrt{n}, \end{aligned}$$

which verifies (3). For (2) we first suppose that $x \geq 0$. In this case we have

$$2bxy \leq (x+by)^2 - b^2 y^2 = (n-b^2)y^2 + z^2 \leq z^2 \leq T^4 n$$

by (3), on recalling that b was defined to be $\lceil \sqrt{n} \rceil$. Thus we have

$$xy \leq T^4 n/2b \leq T^4\sqrt{n}$$

for $x \geq 0$. If $x < 0$ we put $r = \sqrt{q/p} - \eta$, with $0 \leq \eta < 1$. Then if we set $b = \sqrt{n} - \delta$ with $0 \leq \delta < 1$ we may calculate that

$$x = q + p(r+t)^2 - by = p(t-\eta)^2 + 2\eta\delta - 2\delta\sqrt{q/p} - 2\delta t \geq -2\delta(\sqrt{q/p} + T).$$

If $x < 0$ this yields

$$|x| \leq 2\sqrt{q/p} + 2T \leq 2p^{-1}\sqrt{n} + 2T \leq n^{1/4} + 2T,$$

since $p > 2n^{1/4}$. On the other hand

$$y = 2(r+t) \leq 2(\sqrt{q/p} + T) \leq n^{1/4} + 2T$$

by the same argument, so that

$$y|x| \leq (n^{1/4} + 2T)^2 \leq 2T^4\sqrt{n}$$

if $n \geq 1000$, say. This completes the proof of (2).
Finally we observe that $x + by - z = 2p(r + t)^2$. This is divisible by p, but not by q, since $0 < q + r < n^{1/4} + T \leq \sqrt{n}$.

The clever bit in the algorithm, which is adaptable to other problems, is as follows. We are looking for a solution to $Q(x, y) = z^2$ with the variables in certain ranges given by the lemma. Given a range $X < \ell \leq 2X$, the probability that z will have a prime factor ℓ in this range is of order 1 in $\log X$. So if we take T a bit bigger than $\log X$, there is a pretty good chance that there will be a solution in which z has a prime divisor $\ell \in (X, 2X]$. So we run through the various possible p, looking for solutions in which $\ell \mid z$.
How can we solve $Q(x, y) = 0 (\mathrm{mod}\ \ell^2)$? We begin by finding the solutions x_0 of $Q(x_0, 1) = 0 (\mathrm{mod}\ \ell^2)$. There will normally be either two solutions or none. Then $x = x_0 y (\mathrm{mod}\ \ell^2)$ for one of the solutions, so that $x = x_0 y - \mu\ell^2$, say. However if we choose $P \geq 2T^2 n^{1/4}$ then

$$\left|\frac{x_0}{\ell^2} - \frac{\mu}{y}\right| = \left|\frac{x}{\ell^2 y}\right| = \frac{2|x|y}{\ell^2}\frac{1}{2y^2} \leq \frac{4T^4\sqrt{n}}{P^2}\frac{1}{2y^2} \leq \frac{1}{2y^2}$$

by part (2) of the lemma, and this means that μ/y will be one of the convergents in the continued fraction expansion of x_0/ℓ^2.
Hence for each prime ℓ we will be able to find any solution with $\ell \mid z$ in polynomial time. Taking P to be of order $T^2 n^{1/4}$ then gives us a running time $O(n^{1/4+\varepsilon})$, but without a complete guarantee of success.

Other $O(n^{1/4+\varepsilon})$ algorithms

There are in the literature a large number of algorithms with expected running time $O(n^{1/4+\varepsilon})$. Indeed the fastest known algorithms which are both deterministic and provably correct have this running time. (They are therefore "better" than those described above, being both deterministic and provably correct.) One of these, due to Strassen (1977) merely computes $\lfloor\sqrt{n}\rfloor!$ modulo n in a very efficient manner. Taking the gcd with n gives a factor of n.

One of the simplest algorithms is Pollard's ρ-algorithm.
Step 1 Choose an integer n other than 0 or -2, and define $f(x) = x^2 + n$. Set $a = b = 2$.
Step 2 Compute $a' = f(a) (\mathrm{mod}\ n)$ and $b' = f(f(b)) (\mathrm{mod}\ n)$. Find the gcd of $|a' - b'|$ and n. If the value is n give up. If the value is between 1 and n we have found a proper factor of n.
Step 3 If the gcd is 1 replace a by a' and b by b', and return to Step 2.

Example (From Wikipedia)
Take $n = 8051$ and $f(x) = x^2 + 1$. Then

(i) a=2 and $b = 2$ become 5 and 26, and $\gcd(21, 8051) = 1$.

(ii) a=5 and $b = 26$ become 26 and 7474 (mod 8051), and $\gcd(7448, 8051) = 1$.

(iii) a=26 and $b = 7474$ become 677 and 871 (mod 8051), and $\gcd(194, 8051) = 97$.

After k steps we have $a = f^k(2) (\mathrm{mod}\ n)$ and $b = f^{2k}(2) (\mathrm{mod}\ n)$. If p is a prime factor of n we expect the values $f^k(2) (\mathrm{mod}\ p)$ to form a random sequence, which eventually has a repetition, so that $f^k(2) = f^j(2) (\mathrm{mod}\ p)$ with $k < j$. As in the birthday paradox, we expect this to happen with $k, j = O(\sqrt{p})$. Write $\ell = j - k$, so that the sequence has an initial section followed by a cycle of period ℓ. (Pictorially one can think of this as forming a letter "rho" as in the name of the algorithm.) Now, if $h = \ell\lceil k/\ell \rceil$ then $h \geq k$ and $f^h(2) = f^{2h}(2)$. Thus after h steps we will have $p \mid a - b$. We expect $h = 0(\sqrt{p})$. Moreover we will have $n \mid a - b$ only if the cycle lengths for all other prime factors of n happen to divide h.

Thus failures of the algorithm may occur, but are uncommon. Moreover the expected running time is $O(n^{\varepsilon}\sqrt{p})$ where p is the smallest prime factor of n. This is certainly $O(n^{1/4+\varepsilon})$. For comparison, trial division takes time roughly $O(p)$, while some of the other algorithms discussed have running times $O(n^{1/4+\varepsilon})$ even when one of the prime factors is small.

Factor bases

There are many situations where "factor bases" are useful, but we will consider here the problem of finding integers such that $x^2 = y^2 (\mathrm{mod}\ n)$, in the hope of using this to factor n. The basic idea is to find many congruences $x_i^2 = y_i (\mathrm{mod}\ n)$, with $1 \leq i \leq K$, say, and then to look for a subset of indices I such that $\prod_{i \in I} y_i$ is a square y^2. We will then be able to take $x = \prod_{i \in I} x_i$.

Example (From Koblitz)
Take $n = 1829$. We have

$$42^2 = -65, \quad 43^2 = 20, \quad 61^2 = 63, \quad 85^2 = -91, \quad \text{all (mod 1829)}.$$

The product of the four right-hand sides is $2^2 3^2 5^2 7^2 13^2 = 2730^2$. On the other hand $42.43.61.85 = 1459 (\mathrm{mod}\ 1829)$, so that $1459^2 = 2730^2 (\mathrm{mod}\ 1829)$. We

therefore compute $\gcd(2730 - 1459, 1829) = \gcd(1271, 1829) = 31$, giving a factor of 1829.

Generally, how do we find a product of the y_i's which is a square? A **factor base** is a set B of relatively small primes, together with the "prime" -1. For a simple model we will use the primes p up to a certain bound b. We then generate lots of pairs (x, y) such that $x^2 = y(\text{mod } n)$, and keep only those, (x_i, y_i) (with $1 \leq i \leq N$ say), for which y_i is a product of factors in B. We can then write

$$y_i = \prod_{p \in B} p^{e_{p,i}},$$

and in order to make $\prod_{i \in I} y_i$ a square it suffices to find exponents $f_i = 0$ or 1, such that the simultaneous congruences

$$\sum_{i=1}^{N} f_i e_{p,i} = 0(\text{mod } 2) \quad (\forall p \in B)$$

hold. This is a set of equations over $\mathbb{F}_2$ which can be solved by Gaussian elimination. We will have a non-trivial solution if $N > \#B$, so we will need to have more suitable relations than primes in the factor base.

One feature of factor base methods is clear. There will be a non-trivial memory requirement, since we will have to store the various pairs (x_i, y_i) and manipulate the matrix of exponents $e_{p,i}$. In contrast, the algorithms in the course before this point have had relatively minor memory requirements.

Large-prime variants

There are many variations on this line of attack, but one important one retains pairs (x, y) for which $y = y'p$ with y' a product of primes in the factor base and p a medium size prime — outside the factor base but not too enormously large. If two such pairs (x, y) are found which have the same extra factor p, then we can eliminate p, modulo squares, by multiplying the two relations.