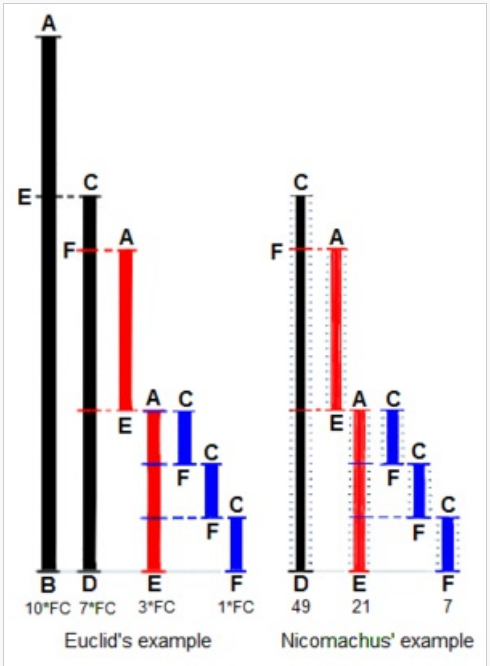
Euclidean algorithm

From Wikipedia, the free encyclopedia

This article is about an algorithm for the greatest common divisor. For the mathematics of space, see Euclidean geometry. For other uses of "Euclidean", see Euclidean (disambiguation).

In mathematics, the **Euclidean algorithm**[a], or **Euclid's algorithm**, is an efficient method for computing the greatest common divisor (GCD) of two numbers, the largest number that divides both of them without leaving a remainder. It is named after the ancient Greek mathematician Euclid, who first described it in Euclid's *Elements* (c. 300 BC). It is an example of an *algorithm*, a step-by-step procedure for performing a calculation according to well-defined rules, and is one of the oldest numerical algorithms in common use. It can be used to reduce fractions to their simplest form, and is a part of many other number-theoretic and cryptographic calculations.



Euclid's method for finding the greatest common divisor (GCD) of two starting lengths BA and DC, both defined to be multiples of a common "unit" length. The length DC being shorter, it is used to "measure" BA, but only once because remainder EA is less than DC. EA now measures (twice) the shorter length DC, with remainder FC shorter than EA. Then FC measures (three times) length EA. Because there is no remainder, the process ends with FC being the GCD. On the right Nicomachus' example with numbers 49 and 21 resulting in their GCD of 7 (derived from Heath 1908:300).

The Euclidean algorithm is based on the principle that the greatest common divisor of two numbers does not change if the larger number is replaced by its difference with the smaller number. For example, 21 is the GCD of 252 and 105 (252 = 21 × 12 and 105 = 21 × 5), and the same number 21 is also the GCD of 105 and 147 = 252 − 105. Since this replacement reduces the larger of the two numbers, repeating this process gives successively smaller pairs of numbers until one of the two numbers reaches zero. When that occurs, the other number (the one that is not zero) is the GCD of the original two numbers. By reversing the steps, the GCD can be expressed as a sum of the two original numbers each multiplied by a positive or negative integer, e.g., 21 = 5 × 105 + (−2) × 252. The fact that the GCD can always be expressed in this way is known as Bézout's identity.

The version of the Euclidean algorithm described above (and by Euclid) can take many subtraction steps to find the GCD when one of the given numbers is much bigger than the other. A more efficient version of the algorithm shortcuts these steps, instead replacing the larger of the two numbers by its remainder when divided by the smaller of the two. With this improvement, the algorithm never requires more steps than five times the number of digits (base 10) of the smaller integer. This was proven by Gabriel Lamé in 1844, and marks the beginning of computational complexity theory. Additional methods for improving the algorithm's efficiency were developed in the 20th century.

The Euclidean algorithm has many theoretical and practical applications. It is used for reducing fractions to their simplest form and for performing division in modular arithmetic. Computations using this algorithm form part of the cryptographic protocols that are used to secure internet communications, and in methods for breaking these cryptosystems by factoring large composite numbers. The Euclidean algorithm may be used to solve Diophantine equations, such as finding numbers that satisfy multiple congruences according to the Chinese remainder theorem, to construct continued fractions, and to find accurate rational approximations to real numbers. Finally, it is a basic tool for proving theorems in number theory such as Lagrange's four-square theorem and the uniqueness of prime factorizations. The original algorithm was described only for natural numbers and geometric lengths (real numbers), but the algorithm was generalized in the 19th century to other types of numbers, such as Gaussian integers and polynomials of one variable. This led to modern abstract algebraic notions such as Euclidean domains.

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
العربية
Беларуская
Български
Català
Čeština
Deutsch
Ελληνικά
Español
Esperanto
فارسی
Français
한국어
Bahasa Indonesia
Italiano
עברית
Latina
Latviešu
Lietuvių
Magyar
Монгол
Nederlands
日本語
Norsk bokmål
Norsk nynorsk
Piemontèis
Plattdüütsch
Polski
Português

Română
Русский
Scots
Simple English
Slovenčina
Slovenščina
Српски / srpski
Srpskohrvatski / српскохрватски
Suomi
Svenska
தமிழ்
ไทย
Türkçe
Українська
Tiếng Việt
中文
Edit links

Contents

Background: greatest common divisor [edit]

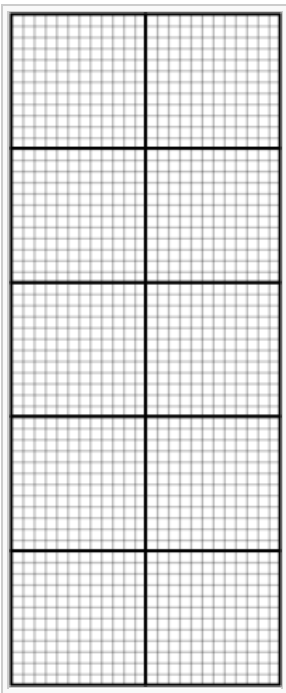
Main article: Greatest common divisor

The Euclidean algorithm calculates the greatest common divisor (GCD) of two natural numbers *a* and *b*. The greatest common divisor *g* is the largest natural number that divides both *a* and *b* without leaving a remainder. Synonyms for the GCD include the *greatest common factor* (GCF), the *highest common factor* (HCF), and the *greatest common measure* (GCM). The greatest common divisor is often written as gcd(*a*, *b*) or, more simply, as (*a*, *b*),[1] although the latter notation is also used for other mathematical concepts, such as two-dimensional vectors.

If gcd(*a*, *b*) = 1, then *a* and *b* are said to be coprime (or relatively prime).[2] This property does not imply that *a* or *b* are themselves prime numbers.[3] For example, neither 6 nor 35 is a prime number, since they both have two prime factors: 6 = 2 × 3 and 35 = 5 × 7. Nevertheless, 6 and 35 are coprime. No natural number other than 1 divides both 6 and 35, since they have no prime factors in common.

Let *g* = gcd(*a*, *b*). Since *a* and *b* are both multiples of *g*, they can be written *a* = *mg* and *b* = *ng*, and there is no larger number *G* > *g* for which this is true. The natural numbers *m* and *n* must be coprime, since any common factor could be factored out of *m* and *n* to make *g* greater. Thus, any other number *c* that divides both *a* and *b* must also divide *g*. The greatest common divisor *g* of *a* and *b* is the unique (positive) common divisor of *a* and *b* that is divisible by any other common divisor *c*.[4]

The GCD can be visualized as follows.[5] Consider a rectangular area *a* by *b*, and any common divisor *c* that divides both *a* and *b* exactly. The sides of the rectangle can be divided into segments of length *c*, which divides the rectangle into a grid of squares of side length *c*. The greatest common divisor *g* is the largest value of *c* for which this is possible. For illustration, a 24-by-60 rectangular area can be divided into a grid of: 1-by-1 squares, 2-by-2 squares, 3-by-3 squares, 4-by-4 squares, 6-by-6 squares or 12-by-12 squares. Therefore, 12 is the greatest common divisor of 24 and 60. A 24-by-60 rectangular area can be divided into a grid of 12-by-12 squares, with two squares along one edge (24/12 = 2) and five squares along the other (60/12 = 5).



A 24-by-60 rectangle is covered with ten 12-by-12 square tiles, where 12 is the GCD of 24 and 60. More generally, an *a*-by-*b* rectangle can be covered with square tiles of side-length *c* only if *c* is a common divisor of *a* and *b*.

The GCD of two numbers *a* and *b* is the product of the prime factors shared by the two numbers, where a same prime factor can be used multiple times, but only as long as the product of these factors divides both *a* and *b*.[6] For example, since 1386 can be factored into 2 × 3 × 3 × 7 × 11, and 3213 can be factored into 3 × 3 × 3 × 7 × 17, the greatest common divisor of 1386 and 3213 equals 63 = 3 × 3 × 7, the product of their shared prime factors. If two numbers have no prime factors in common, their greatest common divisor is 1 (obtained here as an instance of the empty product), in other words they are coprime. A key advantage of the Euclidean algorithm is that it can find the GCD efficiently without having to compute the prime factors.[7][8] Factorization of large integers is believed to be a computationally very difficult problem, and the security of many modern cryptography systems is based upon its infeasibility.[9]

Another definition of the GCD is helpful in advanced mathematics, particularly ring theory.[10] The greatest common divisor *g* of two nonzero numbers *a* and *b* is also their smallest positive integral linear combination, that is, the smallest positive number of the form *ua* + *vb* where *u* and *v* are integers. The set of all integral linear combinations of *a* and *b* is actually the same as the set of all multiples of *g* (*mg*, where *m* is an integer). In modern mathematical language, the ideal generated by *a* and *b* is the ideal generated by *g* alone (an ideal generated by a single element is called a principal ideal, and all ideals of the integers are principal ideals). Some properties of the GCD are in fact easier to see with this description, for instance the fact that any common divisor of *a* and *b* also divides the GCD (it divides both terms of *ua* + *vb*). The equivalence of this GCD definition with the other definitions is described below.

The GCD of three or more numbers equals the product of the prime factors common to all the numbers,[11] but it can also be calculated by repeatedly taking the GCDs of pairs of numbers.[12] For example,

$$\gcd(a, b, c) = \gcd(a, \gcd(b, c)) = \gcd(\gcd(a, b), c) = \gcd(\gcd(a, c), b).$$

Thus, Euclid's algorithm, which computes the GCD of two integers, suffices to calculate the GCD of arbitrarily many integers.

Description [edit]

Procedure [edit]

The Euclidean algorithm proceeds in a series of steps such that the output of each step is used as an input for the next one. Let *k* be an integer that counts the steps of the algorithm, starting with zero. Thus, the initial step corresponds to *k* = 0, the next step corresponds to *k* = 1, and so on.

Each step begins with two nonnegative remainders r_{k-1} and r_{k-2}. Since the algorithm ensures that the remainders decrease steadily with every step, r_{k-1} is less than its predecessor r_{k-2}. The goal of the *k*th step is to find a quotient q_k and remainder r_k that satisfy the equation

$$r_{k-2} = q_k\, r_{k-1} + r_k$$

and that have $r_k < r_{k-1}$. In other words, multiples of the smaller number r_{k-1} are subtracted from the larger number r_{k-2} until the remainder r_k is smaller than r_{k-1}.

In the initial step (*k* = 0), the remainders r_{-2} and r_{-1} equal *a* and *b*, the numbers for which the GCD is sought. In the next step (*k* = 1), the remainders equal *b* and the remainder r_0 of the initial step, and so on. Thus, the algorithm can be written as a sequence of equations

$$a = q_0\, b + r_0$$

$$b = q_1 r_0 + r_1$$
$$r_0 = q_2 r_1 + r_2$$
$$r_1 = q_3 r_2 + r_3$$
$$\ldots$$

If a is smaller than b, the first step of the algorithm swaps the numbers. For example, if $a < b$, the initial quotient q_0 equals zero, and the remainder r_0 is a. Thus, r_k is smaller than its predecessor r_{k-1} for all $k \geq 0$.

Since the remainders decrease with every step but can never be negative, a remainder r_N must eventually equal zero, at which point the algorithm stops.[13] The final nonzero remainder r_{N-1} is the greatest common divisor of a and b. The number N cannot be infinite because there are only a finite number of nonnegative integers between the initial remainder r_0 and zero.

Proof of validity [edit]

The validity of the Euclidean algorithm can be proven by a two-step argument.[14] In the first step, the final nonzero remainder r_{N-1} is shown to divide both a and b. Since it is a common divisor, it must be less than or equal to the greatest common divisor g. In the second step, it is shown that any common divisor of a and b, including g, must divide r_{N-1}; therefore, g must be less than or equal to r_{N-1}. These two conclusions are inconsistent unless $r_{N-1} = g$.

To demonstrate that r_{N-1} divides both a and b (the first step), r_{N-1} divides its predecessor r_{N-2}

$$r_{N-2} = q_N r_{N-1}$$

since the final remainder r_N is zero. r_{N-1} also divides its next predecessor r_{N-3}

$$r_{N-3} = q_{N-1} r_{N-2} + r_{N-1}$$

because it divides both terms on the right-hand side of the equation. Iterating the same argument, r_{N-1} divides all the preceding remainders, including a and b. None of the preceding remainders r_{N-2}, r_{N-3}, etc. divide a and b, since they leave a remainder. Since r_{N-1} is a common divisor of a and b, $r_{N-1} \leq g$.

In the second step, any natural number c that divides both a and b (in other words, any common divisor of a and b) divides the remainders r_k. By definition, a and b can be written as multiples of c: $a = mc$ and $b = nc$, where m and n are natural numbers. Therefore, c divides the initial remainder r_0, since $r_0 = a - q_0 b = mc - q_0 nc = (m - q_0 n)c$. An analogous argument shows that c also divides the subsequent remainders r_1, r_2, etc. Therefore, the greatest common divisor g must divide r_{N-1}, which implies that $g \leq r_{N-1}$. Since the first part of the argument showed the reverse ($r_{N-1} \leq g$), it follows that $g = r_{N-1}$. Thus, g is the greatest common divisor of all the succeeding pairs:[15][16]

$$g = \gcd(a, b) = \gcd(b, r_0) = \gcd(r_0, r_1) = \ldots = \gcd(r_{N-2}, r_{N-1}) = r_{N-1}.$$

Worked example [edit]

For illustration, the Euclidean algorithm can be used to find the greatest common divisor of $a = 1071$ and $b = 462$. To begin, multiples of 462 are subtracted from 1071 until the remainder is less than 462. Two such multiples can be subtracted ($q_0 = 2$), leaving a remainder of 147:

$$1071 = 2 \times 462 + 147.$$

Then multiples of 147 are subtracted from 462 until the remainder is less than 147. Three multiples can be subtracted ($q_1 = 3$), leaving a remainder of 21:

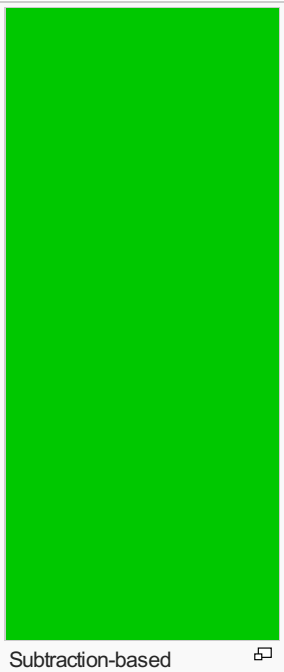
$$462 = 3 \times 147 + 21.$$

Then multiples of 21 are subtracted from 147 until the remainder is less than 21. Seven multiples can be subtracted ($q_2 = 7$), leaving no remainder:

$$147 = 7 \times 21 + 0.$$

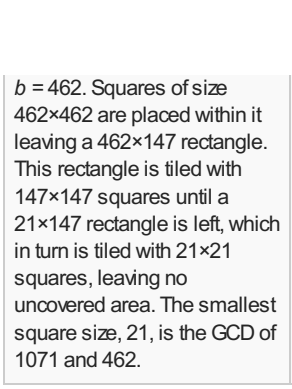
Since the last remainder is zero, the algorithm ends with 21 as the greatest common divisor of 1071 and 462. This agrees with the gcd(1071, 462) found by prime factorization above. In tabular form, the steps are

Step k	Equation	Quotient and remainder
0	$1071 = q_0\,462 + r_0$	$q_0 = 2$ and $r_0 = 147$
1	$462 = q_1\,147 + r_1$	$q_1 = 3$ and $r_1 = 21$
2	$147 = q_2\,21 + r_2$	$q_2 = 7$ and $r_2 = 0$; algorithm ends



Subtraction-based animation of the Euclidean algorithm. The initial rectangle has dimensions $a = 1071$ and

Visualization [edit]



b = 462. Squares of size 462×462 are placed within it leaving a 462×147 rectangle. This rectangle is tiled with 147×147 squares until a 21×147 rectangle is left, which in turn is tiled with 21×21 squares, leaving no uncovered area. The smallest square size, 21, is the GCD of 1071 and 462.

The Euclidean algorithm can be visualized in terms of the tiling analogy given above for the greatest common divisor.[17] Assume that we wish to cover an a-by-b rectangle with square tiles exactly, where a is the larger of the two numbers. We first attempt to tile the rectangle using b-by-b square tiles; however, this leaves an r_0-by-b residual rectangle untiled, where $r_0 < b$. We then attempt to tile the residual rectangle with r_0-by-r_0 square tiles. This leaves a second residual rectangle r_1-by-r_0, which we attempt to tile using r_1-by-r_1 square tiles, and so on. The sequence ends when there is no residual rectangle, i.e., when the square tiles cover the previous residual rectangle exactly. The length of the sides of the smallest square tile is the GCD of the dimensions of the original rectangle. For example, the smallest square tile in the adjacent figure is 21-by-21 (shown in red), and 21 is the GCD of 1071 and 462, the dimensions of the original rectangle (shown in green).

Euclidean division [edit]

Main article: Euclidean division

At every step k, the Euclidean algorithm computes a quotient q_k and remainder r_k from two numbers r_{k-1} and r_{k-2}

$$r_{k-2} = q_k\, r_{k-1} + r_k$$

where the magnitude of r_k is strictly less than that of r_{k-1}. The theorem which underlies the definition of the Euclidean division ensures that such a quotient and remainder always exist and are unique.[18]

In Euclid's original version of the algorithm, the quotient and remainder are found by repeated subtraction; that is, r_{k-1} is subtracted from r_{k-2} repeatedly until the remainder r_k is smaller than r_{k-1}. After that r_k and r_{k-1} are exchanged and the process is iterated. Euclidean division reduces all the steps between two exchanges into a single step, which is thus more efficient. Moreover, the quotients are not needed, thus one may replace Euclidean division by the modulo operation, which gives only the remainder. Thus the iteration of the Euclidean algorithm becomes simply

$$r_k = r_{k-2} \bmod r_{k-1}.$$

Implementations [edit]

Implementations of the algorithm may be expressed in pseudocode. For example, the division-based version may be programmed as[19]

```
function gcd(a, b)
    while b ≠ 0
       t := b
       b := a mod b
       a := t
    return a
```

At the beginning of the kth iteration, the variable b holds the latest remainder r_{k-1}, whereas the variable a holds its predecessor, r_{k-2}. The step $b := a \bmod b$ is equivalent to the above recursion formula $r_k \equiv r_{k-2} \bmod r_{k-1}$. The temporary variable t holds the value of r_{k-1} while the next remainder r_k is being calculated. At the end of the loop iteration, the variable b holds the remainder r_k, whereas the variable a holds its predecessor, r_{k-1}.

In the subtraction-based version which was Euclid's original version, the remainder calculation ($b = a \bmod b$) is replaced by repeated subtraction.[20] Contrary to the division-based version, which works with arbitrary integers as input, the subtraction-based version supposes that the input consists of positive integers and stops when $a = b$:

```
function gcd(a, b)
    while a ≠ b
        if a > b
           a := a − b
        else
           b := b − a
    return a
```

The variables *a* and *b* alternate holding the previous remainders r_{k-1} and r_{k-2}. Assume that *a* is larger than *b* at the beginning of an iteration; then *a* equals r_{k-2}, since $r_{k-2} > r_{k-1}$. During the loop iteration, *a* is reduced by multiples of the previous remainder *b* until *a* is smaller than *b*. Then *a* is the next remainder r_k. Then *b* is reduced by multiples of *a* until it is again smaller than *a*, giving the next remainder r_{k+1}, and so on.

The recursive version[21] is based on the equality of the GCDs of successive remainders and the stopping condition $\gcd(r_{N-1}, 0) = r_{N-1}$.

```
function gcd(a, b)
    if b = 0
       return a
    else
       return gcd(b, a mod b)
```

For illustration, the gcd(1071, 462) is calculated from the equivalent gcd(462, 1071 mod 462) = gcd(462, 147). The latter GCD is calculated from the gcd(147, 462 mod 147) = gcd(147, 21), which in turn is calculated from the gcd(21, 147 mod 21) = gcd(21, 0) = 21.

Method of least absolute remainders [edit]

In another version of Euclid's algorithm, the quotient at each step is increased by one if the resulting negative remainder is smaller in magnitude than the typical positive remainder.[22][23] Previously, the equation

$$r_{k-2} = q_k r_{k-1} + r_k$$

assumed that $|r_{k-1}| > r_k > 0$. However, an alternative negative remainder e_k can be computed:

$$r_{k-2} = (q_k + 1) r_{k-1} + e_k$$

if $r_{k-1} > 0$ or

$$r_{k-2} = (q_k - 1) r_{k-1} + e_k$$

if $r_{k-1} < 0$.

If r_k is replaced by e_k. when $|e_k| < |r_k|$, then one gets a variant of Euclidean algorithm such that

$$|r_k| \leq |r_{k-1}| / 2$$

at each step.

Leopold Kronecker has shown that this version requires the least number of steps of any version of Euclid's algorithm.[22][23] More generally, it has been proven that, for every input numbers *a* and *b*, the number of steps is minimal if and only if q_k is chosen in order that $\left|\frac{r_{k+1}}{r_k}\right| < \frac{1}{\varphi} \sim 0.618,$ where φ is the golden ratio.[24]

Historical development [edit]

The Euclidean algorithm is one of the oldest algorithms in common use.[25] It appears in Euclid's *Elements* (c. 300 BC), specifically in Book 7 (Propositions 1–2) and Book 10 (Propositions 2–3). In Book 7, the algorithm is formulated for integers, whereas in Book 10, it is formulated for lengths of line segments. (In modern usage, one would say it was formulated there for real numbers. But lengths, areas, and volumes, represented as real numbers in modern usage, are not measured in the same units and there is no natural unit of length, area, or volume; the concept of real numbers was unknown at that time.) The latter algorithm is geometrical. The GCD of two lengths *a* and *b* corresponds to the greatest length *g* that measures *a* and *b* evenly; in other words, the lengths *a* and *b* are both integer multiples of the length *g*.

The Euclidean algorithm was probably invented centuries before Euclid, shown here holding a

The algorithm was probably not discovered by Euclid, who compiled results from earlier mathematicians in his *Elements*.[26][27] The mathematician and historian B. L. van der Waerden suggests that Book VII derives from a textbook on number theory written by mathematicians in the school of Pythagoras.[28] The algorithm was probably known by Eudoxus of Cnidus (about 375 BC).[25][29] The algorithm may even pre-date Eudoxus,[30][31] judging from the use of the technical term

ἀνθυφαίρεσις (*anthyphairesis*, reciprocal subtraction) in works by Euclid and Aristotle.[32]

compass.

Centuries later, Euclid's algorithm was discovered independently both in India and in China,[33] primarily to solve Diophantine equations that arise in astronomy and making accurate calendars. In the late 5th century, the Indian mathematician and astronomer Aryabhata described the algorithm as the "pulverizer",[34] perhaps because of its effectiveness in solving Diophantine equations.[35] Although a special case of the Chinese remainder theorem had already been described by Chinese mathematician and astronomer Sun Tzu,[36] the general solution was published by Qin Jiushao in his 1247 book *Shushu Jiuzhang* (數書九章 *Mathematical Treatise in Nine Sections*).[37] The Euclidean algorithm was first described in Europe in the second edition of Bachet's *Problèmes plaisants et délectables* (*Pleasant and enjoyable problems*, 1624).[34] In Europe, it was likewise used to solve Diophantine equations and in developing continued fractions. The extended Euclidean algorithm was published by the English mathematician Nicholas Saunderson, who attributed it to Roger Cotes as a method for computing continued fractions efficiently.[38]

In the 19th century, the Euclidean algorithm led to the development of new number systems, such as Gaussian integers and Eisenstein integers. In 1815, Carl Gauss used the Euclidean algorithm to demonstrate unique factorization of Gaussian integers, although his work was first published in 1832.[39] Gauss mentioned the algorithm in his *Disquisitiones Arithmeticae* (published 1801), but only as a method for continued fractions.[33] Peter Gustav Lejeune Dirichlet seems to have been the first to describe the Euclidean algorithm as the basis for much of number theory.[40] Lejeune Dirichlet noted that many results of number theory, such as unique factorization, would hold true for any other system of numbers to which the Euclidean algorithm could be applied.[41] Lejeune Dirichlet's lectures on number theory were edited and extended by Richard Dedekind, who used Euclid's algorithm to study algebraic integers, a new general type of number. For example, Dedekind was the first to prove Fermat's two-square theorem using the unique factorization of Gaussian integers.[42] Dedekind also defined the concept of a Euclidean domain, a number system in which a generalized version of the Euclidean algorithm can be defined (as described below). In the closing decades of the 19th century, the Euclidean algorithm gradually became eclipsed by Dedekind's more general theory of ideals.[43]

> "[The Euclidean algorithm] is the granddaddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day."
>
> Donald Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 2nd edition (1981), p. 318.

Other applications of Euclid's algorithm were developed in the 19th century. In 1829, Charles Sturm showed that the algorithm was useful in the Sturm chain method for counting the real roots of polynomials in any given interval.[44]

The Euclidean algorithm was the first integer relation algorithm, which is a method for finding integer relations between commensurate real numbers. Several novel integer relation algorithms have been developed, such as the algorithm of Helaman Ferguson and R.W. Forcade (1979)[45] and the LLL algorithm.[46][47]

In 1969, Cole and Davie developed a two-player game based on the Euclidean algorithm, called *The Game of Euclid*,[48] which has an optimal strategy.[49] The players begin with two piles of *a* and *b* stones. The players take turns removing *m* multiples of the smaller pile from the larger. Thus, if the two piles consist of *x* and *y* stones, where *x* is larger than *y*, the next player can reduce the larger pile from *x* stones to $x - my$ stones, as long as the latter is a nonnegative integer. The winner is the first player to reduce one pile to zero stones.[50][51]

Mathematical applications [edit]

Bézout's identity [edit]

Bézout's identity states that the greatest common divisor *g* of two integers *a* and *b* can be represented as a linear sum of the original two numbers *a* and *b*.[52] In other words, it is always possible to find integers *s* and *t* such that $g = sa + tb$.[53][54]

The integers *s* and *t* can be calculated from the quotients q_0, q_1, etc. by reversing the order of equations in Euclid's algorithm.[55] Beginning with the next-to-last equation, *g* can be expressed in terms of the quotient q_{N-1} and the two preceding remainders, r_{N-2} and r_{N-3}:

$$g = r_{N-1} = r_{N-3} - q_{N-1} r_{N-2} .$$

Those two remainders can be likewise expressed in terms of their quotients and preceding remainders,

$$r_{N-2} = r_{N-4} - q_{N-2} r_{N-3} \text{ and}$$
$$r_{N-3} = r_{N-5} - q_{N-3} r_{N-4} .$$

Substituting these formulae for r_{N-2} and r_{N-3} into the first equation yields *g* as a linear sum of the remainders

r_{N-4} and r_{N-5}. The process of substituting remainders by formulae involving their predecessors can be continued until the original numbers a and b are reached:

$$r_2 = r_0 - q_2 r_1$$
$$r_1 = b - q_1 r_0$$
$$r_0 = a - q_0 b.$$

After all the remainders r_0, r_1, etc. have been substituted, the final equation expresses g as a linear sum of a and b: $g = sa + tb$. Bézout's identity, and therefore the previous algorithm, can both be generalized to the context of Euclidean domains.

Principal ideals and related problems [edit]

Bézout's identity provides yet another definition of the greatest common divisor g of two numbers a and b.[10] Consider the set of all numbers $ua + vb$, where u and v are any two integers. Since a and b are both divisible by g, every number in the set is divisible by g. In other words, every number of the set is an integer multiple of g. This is true for every common divisor of a and b. However, unlike other common divisors, the greatest common divisor is a member of the set; by Bézout's identity, choosing $u = s$ and $v = t$ gives g. A smaller common divisor cannot be a member of the set, since every member of the set must be divisible by g. Conversely, any multiple m of g can be obtained by choosing $u = ms$ and $v = mt$, where s and t are the integers of Bézout's identity. This may be seen by multiplying Bézout's identity by m,

$$mg = msa + mtb.$$

Therefore, the set of all numbers $ua + vb$ is equivalent to the set of multiples m of g. In other words, the set of all possible sums of integer multiples of two numbers (a and b) is equivalent to the set of multiples of gcd(a, b). The GCD is said to be the generator of the ideal of a and b. This GCD definition led to the modern abstract algebraic concepts of a principal ideal (an ideal generated by a single element) and a principal ideal domain (a domain in which every ideal is a principal ideal).

Certain problems can be solved using this result.[56] For example, consider two measuring cups of volume a and b. By adding/subtracting u multiples of the first cup and v multiples of the second cup, any volume $ua + vb$ can be measured out. These volumes are all multiples of g = gcd(a, b).

Extended Euclidean algorithm [edit]

Main article: Extended Euclidean algorithm

The integers s and t of Bézout's identity can be computed efficiently using the extended Euclidean algorithm. This extension adds two recursive equations to Euclid's algorithm[57]

$$s_k = s_{k-2} - q_k s_{k-1}$$
$$t_k = t_{k-2} - q_k t_{k-1}$$

with the starting values

$$s_{-2} = 1, t_{-2} = 0$$
$$s_{-1} = 0, t_{-1} = 1.$$

Using this recursion, Bézout's integers s and t are given by $s = s_N$ and $t = t_N$, where $N+1$ is the step on which the algorithm terminates with $r_{N+1} = 0$.

The validity of this approach can be shown by induction. Assume that the recursion formula is correct up to step $k - 1$ of the algorithm; in other words, assume that

$$r_j = s_j a + t_j b$$

for all j less than k. The kth step of the algorithm gives the equation

$$r_k = r_{k-2} - q_k r_{k-1}.$$

Since the recursion formula has been assumed to be correct for r_{k-2} and r_{k-1}, they may be expressed in terms of the corresponding s and t variables

$$r_k = (s_{k-2} a + t_{k-2} b) - q_k (s_{k-1} a + t_{k-1} b).$$

Rearranging this equation yields the recursion formula for step k, as required

$$r_k = s_k a + t_k b = (s_{k-2} - q_k s_{k-1}) a + (t_{k-2} - q_k t_{k-1}) b.$$

Matrix method [edit]

The integers s and t can also be found using an equivalent matrix method.[58] The sequence of equations of Euclid's algorithm

$$a = q_0\, b + r_0$$
$$b = q_1\, r_0 + r_1$$
$$\cdots$$
$$r_{N-2} = q_N\, r_{N-1} + 0$$

can be written as a product of 2-by-2 quotient matrices multiplying a two-dimensional remainder vector

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} q_0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} b \\ r_0 \end{pmatrix} = \begin{pmatrix} q_0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q_1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} r_0 \\ r_1 \end{pmatrix} = \cdots = \prod_{i=0}^{N} \begin{pmatrix} q_i & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} r_{N-1} \\ 0 \end{pmatrix}.$$

Let **M** represent the product of all the quotient matrices

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} = \prod_{i=0}^{N} \begin{pmatrix} q_i & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} q_0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q_1 & 1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} q_N & 1 \\ 1 & 0 \end{pmatrix}.$$

This simplifies the Euclidean algorithm to the form

$$\begin{pmatrix} a \\ b \end{pmatrix} = \mathbf{M} \begin{pmatrix} r_{N-1} \\ 0 \end{pmatrix} = \mathbf{M} \begin{pmatrix} g \\ 0 \end{pmatrix}.$$

To express g as a linear sum of a and b, both sides of this equation can be multiplied by the inverse of the matrix **M**.[58][59] The determinant of **M** equals $(-1)^{N+1}$, since it equals the product of the determinants of the quotient matrices, each of which is negative one. Since the determinant of **M** is never zero, the vector of the final remainders can be solved using the inverse of **M**

$$\begin{pmatrix} g \\ 0 \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} a \\ b \end{pmatrix} = (-1)^{N+1} \begin{pmatrix} m_{22} & -m_{12} \\ -m_{21} & m_{11} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}.$$

Since the top equation gives

$$g = (-1)^{N+1} \left(m_{22}\, a - m_{12}\, b \right),$$

the two integers of Bézout's identity are $s = (-1)^{N+1} m_{22}$ and $t = (-1)^{N} m_{12}$. The matrix method is as efficient as the equivalent recursion, with two multiplications and two additions per step of the Euclidean algorithm.

Euclid's lemma and unique factorization [edit]

Bézout's identity is essential to many applications of Euclid's algorithm, such as demonstrating the unique factorization of numbers into prime factors.[60] To illustrate this, suppose that a number L can be written as a product of two factors u and v, that is, $L = uv$. If another number w also divides L but is coprime with u, then w must divide v, by the following argument: If the greatest common divisor of u and w is 1, then integers s and t can be found such that

$$1 = su + tw\,.$$

by Bézout's identity. Multiplying both sides by v gives the relation

$$v = suv + twv = sL + twv\,.$$

Since w divides both terms on the right-hand side, it must also divide the left-hand side, v. This result is known as Euclid's lemma.[61] Specifically, if a prime number divides L, then it must divide at least one factor of L. Conversely, if a number w is coprime to each of a series of numbers a_1, a_2, ..., a_n, then w is also coprime to their product, $a_1 \times a_2 \times \ldots \times a_n$.[61]

Euclid's lemma suffices to prove that every number has a unique factorization into prime numbers.[62] To see this, assume the contrary, that there are two independent factorizations of L into m and n prime factors, respectively

$$L = p_1 p_2 \ldots p_m = q_1 q_2 \ldots q_n\,.$$

Since each prime p divides L by assumption, it must also divide one of the q factors; since each q is prime as well, it must be that $p = q$. Iteratively dividing by the p factors shows that each p has an equal counterpart q; the two prime factorizations are identical except for their order. The unique factorization of numbers into primes has many applications in mathematical proofs, as shown below.

Linear Diophantine equations [edit]

Diophantine equations are equations in which the solutions are restricted to integers; they are named after the 3rd-century Alexandrian mathematician Diophantus.[63] A typical *linear* Diophantine equation seeks integers x and y such that[64]

$$ax + by = c$$

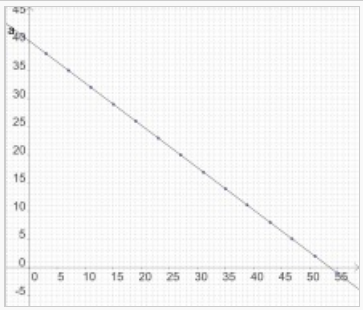
where *a*, *b* and *c* are given integers. This can be written as an equation for *x* in modular arithmetic:

$$ax \equiv c \bmod b.$$

Let *g* be the greatest common divisor of *a* and *b*. Both terms in *ax* + *by* are divisible by *g*; therefore, *c* must also be divisible by *g*, or the equation has no solutions. By dividing both sides by *c*/*g*, the equation can be reduced to Bezout's identity

$$sa + tb = g$$

where *s* and *t* can be found by the extended Euclidean algorithm.[65] This provides one solution to the Diophantine equation, $x_1 = s\ (c/g)$ and $y_1 = t\ (c/g)$.



Plot of a linear Diophantine equation, 9*x* + 12*y* = 483. The solutions are shown as blue circles.

In general, a linear Diophantine equation has no solutions, or an infinite number of solutions.[66] To find the latter, consider two solutions, (x_1, y_1) and (x_2, y_2), where

$$ax_1 + by_1 = c = ax_2 + by_2$$

or equivalently

$$a(x_1 - x_2) = b(y_2 - y_1).$$

Therefore, the smallest difference between two *x* solutions is *b*/*g*, whereas the smallest difference between two *y* solutions is *a*/*g*. Thus, the solutions may be expressed as

$$x = x_1 - bu/g$$
$$y = y_1 + au/g.$$

By allowing *u* to vary over all possible integers, an infinite family of solutions can be generated from a single solution (x_1, y_1). If the solutions are required to be *positive* integers ($x > 0$, $y > 0$), only a finite number of solutions may be possible. This restriction on the acceptable solutions allows some systems of Diophantine equations with more unknowns than equations to have a finite number of solutions;[67] this is impossible for a system of linear equations when the solutions can be any real number (see Underdetermined system).

Multiplicative inverses and the RSA algorithm [edit]

A finite field is a set of numbers with four generalized operations. The operations are called addition, subtraction, multiplication and division and have their usual properties, such as commutativity, associativity and distributivity. An example of a finite field is the set of 13 numbers {0, 1, 2, …, 12} using modular arithmetic. In this field, the results of any mathematical operation (addition, subtraction, multiplication, or division) is reduced modulo 13; that is, multiples of 13 are added or subtracted until the result is brought within the range 0–12. For example, the result of 5 × 7 = 35 mod 13 = 9. Such finite fields can be defined for any prime *p*; using more sophisticated definitions, they can also be defined for any power *m* of a prime p^m. Finite fields are often called Galois fields, and are abbreviated as GF(*p*) or GF(p^m).

In such a field with *m* numbers, every nonzero element *a* has a unique modular multiplicative inverse, a^{-1} such that $aa^{-1} = a^{-1}a \equiv 1 \bmod m$. This inverse can be found by solving the congruence equation $ax \equiv 1 \bmod m$,[68] or the equivalent linear Diophantine equation[69]

$$ax + my = 1.$$

This equation can be solved by the Euclidean algorithm, as described above. Finding multiplicative inverses is an essential step in the RSA algorithm, which is widely used in electronic commerce; specifically, the equation determines the integer used to decrypt the message.[70] Note that although the RSA algorithm uses rings rather than fields, the Euclidean algorithm can still be used to find a multiplicative inverse where one exists. The Euclidean algorithm also has other applications in error-correcting codes; for example, it can be used as an alternative to the Berlekamp–Massey algorithm for decoding BCH and Reed–Solomon codes, which are based on Galois fields.[71]

Chinese remainder theorem [edit]

Euclid's algorithm can also be used to solve multiple linear Diophantine equations.[72] Such equations arise in the Chinese remainder theorem, which describes a novel method to represent an integer *x*. Instead of representing an integer by its digits, it may be represented by its remainders x_i modulo a set of *N* coprime numbers m_i:[73]

$$\begin{aligned} x_1 &\equiv x \bmod m_1 \\ x_2 &\equiv x \bmod m_2 \\ &\vdots \\ x_N &\equiv x \bmod m_N . \end{aligned}$$

The goal is to determine x from its N remainders x_i. The solution is to combine the multiple equations into a single linear Diophantine equation with a much larger modulus M that is the product of all the individual moduli m_i, and define M_i as

$$M_i = \frac{M}{m_i}.$$

Thus, each M_i is the product of all the moduli *except* m_i. The solution depends on finding N new numbers h_i such that

$$M_i h_i \equiv 1 \bmod m_i .$$

With these numbers h_i, any integer x can be reconstructed from its remainders x_i by the equation

$$x \equiv (x_1 M_1 h_1 + x_2 M_2 h_2 + \cdots + x_N M_N h_N) \bmod M .$$

Since these numbers h_i are the multiplicative inverses of the M_i, they may be found using Euclid's algorithm as described in the previous subsection.

Stern–Brocot tree [edit]

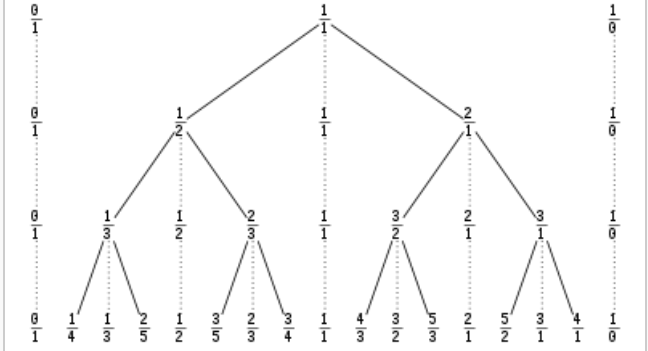
Main article: Stern–Brocot tree

The Euclidean algorithm can be used to arrange the set of all positive rational numbers into an infinite binary search tree, called the Stern–Brocot tree. The number 1 (expressed as a fraction 1/1) is placed at the root of the tree, and the location of any other number a/b can be found by computing gcd(a,b) using the original form of the Euclidean algorithm, in which each step replaces the larger of the two given numbers by its difference with the smaller number (not its remainder), stopping when two equal numbers are reached. A step of the Euclidean algorithm that replaces the first of the two numbers corresponds to a step in the tree from a node to its right child, and a step that replaces the second of the two numbers corresponds to a step in the tree from a node to its left child. The sequence of steps constructed in this way does not depend on whether a/b is given in lowest terms, and forms a path from the root to a node containing the number a/b.[74] This fact can be used to prove that each positive rational number appears exactly once in this tree.

For example, 3/4 can be found by starting at the root, going to the left once, then to the right twice:

$$\begin{aligned} &\gcd(3,4) && \leftarrow \\ =&\gcd(3,1) && \rightarrow \\ =&\gcd(2,1) && \rightarrow \\ =&\gcd(1,1). \end{aligned}$$

The Euclidean algorithm has almost the same relationship to another binary tree on the rational numbers called the Calkin–Wilf tree. The difference is that the path is reversed: instead of producing a path from the root of the tree to a target, it produces a path from the target to the root.



The Stern–Brocot tree, and the Stern–Brocot sequences of order i for $i = 1, 2, 3, 4$

Continued fractions [edit]

The Euclidean algorithm has a close relationship with continued fractions.[75] The sequence of equations can be written in the form

$$\begin{aligned}
\frac{a}{b} &= q_0 + \frac{r_0}{b} \\
\frac{b}{r_0} &= q_1 + \frac{r_1}{r_0} \\
\frac{r_0}{r_1} &= q_2 + \frac{r_2}{r_1} \\
&\vdots \\
\frac{r_{k-2}}{r_{k-1}} &= q_k + \frac{r_k}{r_{k-1}} \\
&\vdots \\
\frac{r_{N-2}}{r_{N-1}} &= q_N \,.
\end{aligned}$$

The last term on the right-hand side always equals the inverse of the left-hand side of the next equation. Thus, the first two equations may be combined to form

$$\frac{a}{b} = q_0 + \cfrac{1}{q_1 + \cfrac{r_1}{r_0}}.$$

The third equation may be used to substitute the denominator term r_1/r_0, yielding

$$\frac{a}{b} = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{r_2}{r_1}}}.$$

The final ratio of remainders r_k/r_{k-1} can always be replaced using the next equation in the series, up to the final equation. The result is a continued fraction

$$\frac{a}{b} = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots + \cfrac{1}{q_N}}}} = [q_0; q_1, q_2, \ldots, q_N]\,.$$

In the worked example above, the gcd(1071, 462) was calculated, and the quotients q_k were 2, 3 and 7, respectively. Therefore, the fraction 1071/462 may be written

$$\frac{1071}{462} = 2 + \cfrac{1}{3 + \cfrac{1}{7}} = [2; 3, 7]$$

as can be confirmed by calculation.

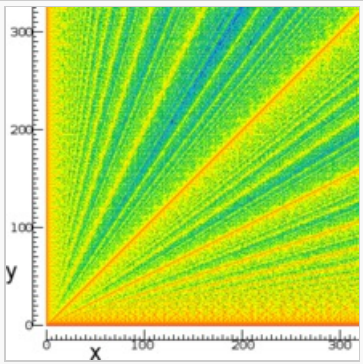
Factorization algorithms [edit]

Calculating a greatest common divisor is an essential step in several integer factorization algorithms,[76] such as Pollard's rho algorithm,[77] Shor's algorithm,[78] Dixon's factorization method[79] and the Lenstra elliptic curve factorization.[80] The Euclidean algorithm may be used to find this GCD efficiently. Continued fraction factorization uses continued fractions, which are determined using Euclid's algorithm.[81]

Algorithmic efficiency [edit]

The computational efficiency of Euclid's algorithm has been studied thoroughly.[82] This efficiency can be described by the number of division steps the algorithm requires, multiplied by the computational expense of each step. The first known analysis of Euclid's algorithm is due to A.-A.-L. Reynaud in 1811,[83] who showed that the number of division steps on input (u, v) is bounded by v; later he improved this to $v/2 + 2$. Later, in 1841, P.-J.-E. Finck showed[84] that the number of division steps is at most $2 \log_2 v + 1$, and hence Euclid's algorithm runs in time polynomial in the size of the input; also see.[85] His analysis was refined by Gabriel Lamé in 1844,[86] who showed that the number of steps required for completion is never more than five times the number h of base-10 digits of the smaller number b.[87][88]

In the uniform cost model (suitable for analyzing the complexity of gcd calculation on numbers that fit into a single machine word), each step of the algorithm takes constant time, and Lamé's analysis implies that the total running time is also $O(h)$. However, in a model of computation suitable for computation with larger numbers, the computational expense of a single remainder computation in the algorithm can be as large as $O(h^2)$.[89] In this case the total time for all of the steps of the algorithm can be analyzed using a telescoping series, showing that it is also $O(h^2)$. Modern algorithmic techniques based on the Schönhage–Strassen algorithm for fast integer multiplication can be used to speed this up, leading to subquadratic algorithms for the GCD.[90][91]



Number of steps in the Euclidean algorithm for gcd(*x*,*y*). Lighter (red and yellow) points indicate relatively few steps, whereas darker (green and blue) points indicate more steps. The largest dark area follows the line $y = \Phi x$, where Φ represents the Golden ratio.

Number of steps [edit]

The number of steps to calculate the GCD of two natural numbers, a and b, may be denoted by $T(a, b)$.[92] If g is the GCD of a and b, then $a = mg$ and $b = ng$ for two coprime numbers m and n. Then

$$T(a, b) = T(m, n)$$

as may be seen by dividing all the steps in the Euclidean algorithm by g.[93] By the same argument, the number of steps remains the same if a and b are multiplied by a common factor w: $T(a, b) = T(wa, wb)$. Therefore, the number of steps T may vary dramatically between neighboring pairs of numbers, such as $T(a, b)$ and $T(a, b + 1)$, depending on the size of the two GCDs.

The recursive nature of the Euclidean algorithm gives another equation

$$T(a, b) = 1 + T(b, r_0) = 2 + T(r_0, r_1) = \ldots = N + T(r_{N-2}, r_{N-1}) = N + 1$$

where $T(x, 0) = 0$ by assumption.[92]

Worst-case [edit]

If the Euclidean algorithm requires N steps for a pair of natural numbers $a > b > 0$, the smallest values of a and b for which this is true are the Fibonacci numbers F_{N+2} and F_{N+1}, respectively.[94] This can be shown by induction.[95] If $N = 1$, b divides a with no remainder; the smallest natural numbers for which this is true is $b = 1$ and $a = 2$, which are F_2 and F_3, respectively. Now assume that the result holds for all values of N up to $M - 1$. The first step of the M-step algorithm is $a = q_0 b + r_0$, and the second step is $b = q_1 r_0 + r_1$. Since the algorithm is recursive, it required $M - 1$ steps to find $\gcd(b, r_0)$ and their smallest values are F_{M+1} and F_M. The smallest value of a is therefore when $q_0 = 1$, which gives $a = b + r_0 = F_{M+1} + F_M = F_{M+2}$. This proof, published by Gabriel Lamé in 1844, represents the beginning of computational complexity theory,[96] and also the first practical application of the Fibonacci numbers.[94]

This result suffices to show that the number of steps in Euclid's algorithm can never be more than five times the number of its digits (base 10).[97] For if the algorithm requires N steps, then b is greater than or equal to F_{N+1} which in turn is greater than or equal to φ^{N-1}, where φ is the golden ratio. Since $b \geq \varphi^{N-1}$, then $N - 1 \leq \log_\varphi b$. Since $\log_{10}\varphi > 1/5$, $(N - 1)/5 < \log_{10}\varphi \log_\varphi b = \log_{10} b$. Thus, $N \leq 5 \log_{10} b$. Thus, the Euclidean algorithm always needs less than $O(h)$ divisions, where h is the number of digits in the smaller number b.

Average [edit]

The average number of steps taken by the Euclidean algorithm has been defined in three different ways. The first definition is the average time $T(a)$ required to calculate the GCD of a given number a and a smaller natural number b chosen with equal probability from the integers 0 to $a - 1$[92]

$$T(a) = \frac{1}{a} \sum_{0 \leq b < a} T(a, b).$$

However, since $T(a, b)$ fluctuates dramatically with the GCD of the two numbers, the averaged function $T(a)$ is likewise "noisy".[98]

To reduce this noise, a second average $\tau(a)$ is taken over all numbers coprime with a

$$\tau(a) = \frac{1}{\varphi(a)} \sum_{\substack{0 \leq b < a \\ \gcd(a,b)=1}} T(a, b).$$

There are $\varphi(a)$ coprime integers less than a, where φ is Euler's totient function. This tau average grows smoothly with a[99][100]

$$\tau(a) = \frac{12}{\pi^2} \ln 2 \ln a + C + O(a^{-1/6-\epsilon})$$

with the residual error being of order $a^{-(1/6)+\varepsilon}$, where ε is infinitesimal. The constant *C* (*Porter's Constant*[101])in this formula equals

$$C = -\frac{1}{2} + \frac{6 \ln 2}{\pi^2}(4\gamma - 24\pi^2\zeta'(2) + 3\ln 2 - 2) \approx 1.467$$

where γ is the Euler–Mascheroni constant and ζ' is the derivative of the Riemann zeta function.[102][103] The leading coefficient $(12/\pi^2) \ln 2$ was determined by two independent methods.[104][105]

Since the first average can be calculated from the tau average by summing over the divisors *d* of *a*[106]

$$T(a) = \frac{1}{a} \sum_{d|a} \varphi(d)\tau(d)$$

it can be approximated by the formula[107]

$$T(a) \approx C + \frac{12}{\pi^2} \ln 2 \left(\ln a - \sum_{d|a} \frac{\Lambda(d)}{d} \right)$$

where Λ(*d*) is the Mangoldt function.[108]

A third average *Y*(*n*) is defined as the mean number of steps required when both *a* and *b* are chosen randomly (with uniform distribution) from 1 to *n*[107]

$$Y(n) = \frac{1}{n^2} \sum_{a=1}^{n} \sum_{b=1}^{n} T(a,b) = \frac{1}{n} \sum_{a=1}^{n} T(a).$$

Substituting the approximate formula for *T*(*a*) into this equation yields an estimate for *Y*(*n*)[109]

$$Y(n) \approx \frac{12}{\pi^2} \ln 2 \ln n + 0.06.$$

Computational expense per step [edit]

In each step *k* of the Euclidean algorithm, the quotient q_k and remainder r_k are computed for a given pair of integers r_{k-2} and r_{k-1}

$$r_{k-2} = q_k\, r_{k-1} + r_k.$$

The computational expense per step is associated chiefly with finding q_k, since the remainder r_k can be calculated quickly from r_{k-2}, r_{k-1}, and q_k

$$r_k = r_{k-2} - q_k\, r_{k-1}.$$

The computational expense of dividing *h*-bit numbers scales as $O(h(\ell+1))$, where ℓ is the length of the quotient.[89]

For comparison, Euclid's original subtraction-based algorithm can be much slower. A single integer division is equivalent to the quotient *q* number of subtractions. If the ratio of *a* and *b* is very large, the quotient is large and many subtractions will be required. On the other hand, it has been shown that the quotients are very likely to be small integers. The probability of a given quotient *q* is approximately $\ln|u/(u-1)|$ where $u = (q+1)^2$.[110] For illustration, the probability of a quotient of 1, 2, 3, or 4 is roughly 41.5%, 17.0%, 9.3%, and 5.9%, respectively. Since the operation of subtraction is faster than division, particularly for large numbers,[111] the subtraction-based Euclid's algorithm is competitive with the division-based version.[112] This is exploited in the binary version of Euclid's algorithm.[113]

Combining the estimated number of steps with the estimated computational expense per step shows that the Euclid's algorithm grows quadratically (h^2) with the average number of digits *h* in the initial two numbers *a* and *b*. Let $h_0, h_1, \ldots, h_{N-1}$ represent the number of digits in the successive remainders $r_0, r_1, \ldots, r_{N-1}$. Since the number of steps *N* grows linearly with *h*, the running time is bounded by

$$O\Big(\sum_{i<N} h_i(h_i - h_{i+1} + 2)\Big) \subseteq O\Big(h \sum_{i<N} (h_i - h_{i+1} + 2)\Big) \subseteq O(h(h_0 + 2N)) \subseteq O(h^2).$$

Alternative methods [edit]

Euclid's algorithm is widely used in practice, especially for small numbers, due to its simplicity.[114] For comparison, the efficiency of alternatives to Euclid's algorithm may be determined.

One inefficient approach to finding the GCD of two natural numbers *a* and *b* is to calculate all their common

divisors; the GCD is then the largest common divisor. The common divisors can be found by dividing both numbers by successive integers from 2 to the smaller number b. The number of steps of this approach grows linearly with b, or exponentially in the number of digits. Another inefficient approach is to find the prime factors of one or both numbers. As noted above, the GCD equals the product of the prime factors shared by the two numbers a and b.[6] Present methods for prime factorization are also inefficient; many modern cryptography systems even rely on that inefficiency.[9]

The binary GCD algorithm is an efficient alternative that substitutes division with faster operations by exploiting the binary representation used by computers.[115][116] However, this alternative also scales like $O(h^2)$. It is generally faster than the Euclidean algorithm on real computers, even though it scales in the same way.[90] Additional efficiency can be gleaned by examining only the leading digits of the two numbers a and b.[117][118] The binary algorithm can be extended to other bases (k-ary algorithms),[119] with up to fivefold increases in speed.[120] Lehmer's GCD algorithm uses the same general principle as the binary algorithm to speed up GCD computations in arbitrary bases.

A recursive approach for very large integers (with more than 25,000 digits) leads to *subquadratic integer GCD algorithms*,[121] such as those of Schönhage,[122][123] and Stehlé and Zimmermann.[124] These algorithms exploit the 2×2 matrix form of the Euclidean algorithm given above. These subquadratic methods generally scale as $O(h (\log h)^2 (\log \log h))$.[90][91]

Generalizations [edit]

Although the Euclidean algorithm is used to find the greatest common divisor of two natural numbers (positive integers), it may be generalized to the real numbers, and to other mathematical objects, such as polynomials,[125] quadratic integers[126] and Hurwitz quaternions.[127] In the latter cases, the Euclidean algorithm is used to demonstrate the crucial property of unique factorization, i.e., that such numbers can be factored uniquely into irreducible elements, the counterparts of prime numbers. Unique factorization is essential to many proofs of number theory.

Rational and real numbers [edit]

Euclid's algorithm can be applied to real numbers, as described by Euclid in Book 10 of his *Elements*. The goal of the algorithm is to identify a real number g such that two given real numbers, a and b, are integer multiples of it: $a = mg$ and $b = ng$, where m and n are integers.[26] This identification is equivalent to finding an integer relation among the real numbers a and b; that is, it determines integers s and t such that $sa + tb = 0$. Euclid uses this algorithm to treat the question of incommensurable lengths.[128][129]

The real-number Euclidean algorithm differs from its integer counterpart in two respects. First, the remainders r_k are real numbers, although the quotients q_k are integers as before. Second, the algorithm is not guaranteed to end in a finite number N of steps. If it does, the fraction a/b is a rational number, i.e., the ratio of two integers

$$a/b = mg/ng = m/n$$

and can be written as a finite continued fraction $[q_0; q_1, q_2, \ldots, q_N]$. If the algorithm does not stop, the fraction a/b is an irrational number and can be described by an infinite continued fraction $[q_0; q_1, q_2, \ldots]$.[130] Examples of infinite continued fractions are the golden ratio $\varphi = [1; 1, 1, \ldots]$ and the square root of two, $\sqrt{2} = [1; 2, 2, \ldots]$.[131] The algorithm is unlikely to stop, since almost all ratios a/b of two real numbers are irrational.[132]

An infinite continued fraction may be truncated at a step k $[q_0; q_1, q_2, \ldots, q_k]$ to yield an approximation to a/b that improves as k is increased. The approximation is described by convergents m_k/n_k; the numerator and denominators are coprime and obey the recurrence relation

$$m_k = q_k m_{k-1} + m_{k-2}$$
$$n_k = q_k n_{k-1} + n_{k-2}$$

where $m_{-1} = n_{-2} = 1$ and $m_{-2} = n_{-1} = 0$ are the initial values of the recursion. The convergent m_k/n_k is the best rational number approximation to a/b with denominator n_k:[133]

$$\left|\frac{a}{b} - \frac{m_k}{n_k}\right| < \frac{1}{n_k^2}.$$

Polynomials [edit]

Main article: Polynomial greatest common divisor

Polynomials in a single variable x can be added, multiplied and factored into irreducible polynomials, which are the analogs of the prime numbers for integers. The greatest common divisor polynomial $g(x)$ of two polynomials $a(x)$ and $b(x)$ is defined as the product of their shared irreducible polynomials, which can be identified using the

Euclidean algorithm.[125] The basic procedure is similar to integers. At each step k, a quotient polynomial $q_k(x)$ and a remainder polynomial $r_k(x)$ are identified to satisfy the recursive equation

$$r_{k-2}(x) = q_k(x)\, r_{k-1}(x) + r_k(x)$$

where $r_{-2}(x) = a(x)$ and $r_{-1}(x) = b(x)$. The quotient polynomial is chosen so that the leading term of $q_k(x)\, r_{k-1}(x)$ equals the leading term of $r_{k-2}(x)$; this ensures that the degree of each remainder is smaller than the degree of its predecessor $\deg[r_k(x)] < \deg[r_{k-1}(x)]$. Since the degree is a nonnegative integer, and since it decreases with every step, the Euclidean algorithm concludes in a finite number of steps. The final nonzero remainder is the greatest common divisor of the original two polynomials, $a(x)$ and $b(x)$.[134]

For example, consider the following two quartic polynomials, which each factor into two quadratic polynomials

$$a(x) = x^4 - 4x^3 + 4x^2 - 3x + 14 = (x^2 - 5x + 7)(x^2 + x + 2)$$

and

$$b(x) = x^4 + 8x^3 + 12x^2 + 17x + 6 = (x^2 + 7x + 3)(x^2 + x + 2).$$

Dividing $a(x)$ by $b(x)$ yields a remainder $r_0(x) = x^3 + (2/3)\, x^2 + (5/3)\, x - (2/3)$. In the next step, $b(x)$ is divided by $r_0(x)$ yielding a remainder $r_1(x) = x^2 + x + 2$. Finally, dividing $r_0(x)$ by $r_1(x)$ yields a zero remainder, indicating that $r_1(x)$ is the greatest common divisor polynomial of $a(x)$ and $b(x)$, consistent with their factorization.

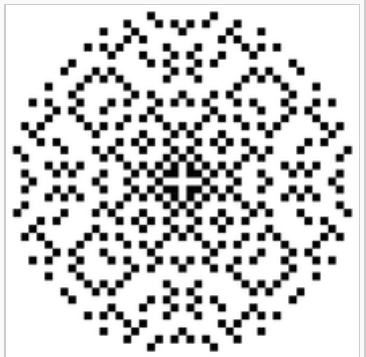
Many of the applications described above for integers carry over to polynomials.[135] The Euclidean algorithm can be used to solve linear Diophantine equations and Chinese remainder problems for polynomials; continued fractions of polynomials can also be defined.

The polynomial Euclidean algorithm has other applications, such as Sturm chains, a method for counting the zeros of a polynomial that lie inside a given real interval.[136] This in turn has applications in several areas, such as the Routh–Hurwitz stability criterion in control theory.[137]

Finally, the coefficients of the polynomials need not be drawn from integers, real numbers or even the complex numbers. For example, the coefficients may be drawn from a general field, such as the finite fields GF(p) described above. The corresponding conclusions about the Euclidean algorithm and its applications hold even for such polynomials.[125]

Gaussian integers [edit]

The Gaussian integers are complex numbers of the form $\alpha = u + vi$, where u and v are ordinary integers and i is the square root of negative one.[138] By defining an analog of the Euclidean algorithm, Gaussian integers can be shown to be uniquely factorizable, by the argument above.[39] This unique factorization is helpful in many applications, such as deriving all Pythagorean triples or proving Fermat's theorem on sums of two squares.[138] In general, the Euclidean algorithm is convenient in such applications, but not essential; for example, the theorems can often be proven by other arguments.



Distribution of Gaussian primes $u + vi$ in the complex plane, with norms $u^2 + v^2$ less than 500

The Euclidean algorithm developed for two Gaussian integers α and β is nearly the same as that for normal integers,[139] but differs in two respects. As before, the task at each step k is to identify a quotient q_k and a remainder r_k such that

$$r_k = r_{k-2} - q_k\, r_{k-1}$$

where $r_{k-2} = \alpha$, $r_{k-1} = \beta$, and every remainder is strictly smaller than its predecessor, $|r_k| < |r_{k-1}|$. The first difference is that the quotients and remainders are themselves Gaussian integers, and thus are complex numbers. The quotients q_k are generally found by rounding the real and complex parts of the exact ratio (such as the complex number α/β) to the nearest integers.[139] The second difference lies in the necessity of defining how one complex remainder can be "smaller" than another. To do this, a norm function $f(u + vi) = u^2 + v^2$ is defined, which converts every Gaussian integer $u + vi$ into a normal integer. After each step k of the Euclidean algorithm, the norm of the remainder $f(r_k)$ is smaller than the norm of the preceding remainder, $f(r_{k-1})$. Since the norm is a nonnegative integer and decreases with every step, the Euclidean algorithm for Gaussian integers ends in a finite number of steps.[140] The final nonzero remainder is the gcd(α,β), the Gaussian integer of largest norm that divides both α and β; it is unique up to multiplication by a unit, ±1 or ±i.[141]

Many of the other applications of the Euclidean algorithm carry over to Gaussian integers. For example, it can be used to solve linear Diophantine equations and Chinese remainder problems for Gaussian integers;[142] continued fractions of Gaussian integers can also be defined.[139]

Euclidean domains [edit]

A set of elements under two binary operations, + and −, is called a Euclidean domain if it forms a commutative ring *R* and, roughly speaking, if a generalized Euclidean algorithm can be performed on them.[143][144] The two operations of such a ring need not be the addition and multiplication of ordinary arithmetic; rather, they can be more general, such as the operations of a mathematical group or monoid. Nevertheless, these general operations should respect many of the laws governing ordinary arithmetic, such as commutativity, associativity and distributivity.

The generalized Euclidean algorithm requires a *Euclidean function*, i.e., a mapping *f* from *R* into the set of nonnegative integers such that, for any two nonzero elements *a* and *b* in *R*, there exist *q* and *r* in *R* such that $a = qb + r$ and $f(r) < f(b)$.[145] An example of this mapping is the norm function used to order the Gaussian integers above.[146] The function *f* can be the magnitude of the number, or the degree of a polynomial.[147] The basic principle is that each step of the algorithm reduces *f* inexorably; hence, if *f* can be reduced only a finite number of times, the algorithm must stop in a finite number of steps. This principle relies heavily on the natural well-ordering of the non-negative integers;[148] roughly speaking, this requires that every non-empty set of non-negative integers has a smallest member.

The fundamental theorem of arithmetic applies to any Euclidean domain: Any number from a Euclidean domain can be factored uniquely into irreducible elements. Any Euclidean domain is a unique factorization domain (UFD), although the converse is not true.[148] The Euclidean domains and the UFD's are subclasses of the GCD domains, domains in which a greatest common divisor of two numbers always exists.[149] In other words, a greatest common divisor may exist (for all pairs of elements in a domain), although it may not be possible to find it using a Euclidean algorithm. A Euclidean domain is always a principal ideal domain (PID), an integral domain in which every ideal is a principal ideal.[150] Again, the converse is not true: not every PID is a Euclidean domain.

The unique factorization of Euclidean domains is useful in many applications. For example, the unique factorization of the Gaussian integers is convenient in deriving formulae for all Pythagorean triples and in proving Fermat's theorem on sums of two squares.[138] Unique factorization was also a key element in an attempted proof of Fermat's Last Theorem published in 1847 by Gabriel Lamé, the same mathematician who analyzed the efficiency of Euclid's algorithm, based on a suggestion of Joseph Liouville.[151] Lamé's approach required the unique factorization of numbers of the form $x + \omega y$, where *x* and *y* are integers, and $\omega = e^{2i\pi/n}$ is an *n*th root of 1, that is, $\omega^n = 1$. Although this approach succeeds for some values of *n* (such as *n*=3, the Eisenstein integers), in general such numbers do *not* factor uniquely. This failure of unique factorization in some cyclotomic fields led Ernst Kummer to the concept of ideal numbers and, later, Richard Dedekind to ideals.[152]

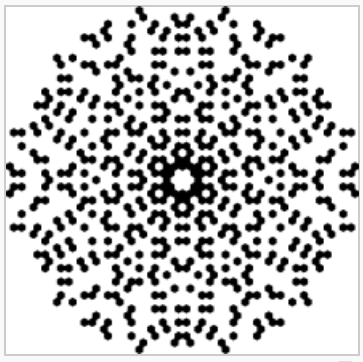
Unique factorization of quadratic integers [edit]

The quadratic integer rings are helpful to illustrate Euclidean domains. Quadratic integers are generalizations of the Gaussian integers in which the imaginary unit *i* is replaced by a number ω. Thus, they have the form $u + v\omega$, where *u* and *v* are integers and ω has one of two forms, depending on a parameter *D*. If *D* does not equal a multiple of four plus one, then

$$\omega = \sqrt{D}.$$

If, however, *D* does equal a multiple of four plus one, then

$$\omega = \frac{1+\sqrt{D}}{2}.$$



Distribution of Eisenstein primes $u + v\omega$ in the complex plane, with norms less than 500. The number ω equals the cube root of 1.

If the function *f* corresponds to a norm function, such as that used to order the Gaussian integers above, then the domain is known as *norm-Euclidean*. The norm-Euclidean rings of quadratic integers are exactly those where *D* = −11, −7, −3, −2, −1, 2, 3, 5, 6, 7, 11, 13, 17, 19, 21, 29, 33, 37, 41, 57 or 73.[18][153] The quadratic integers with *D* = −1 and −3 are known as the Gaussian integers and Eisenstein integers, respectively.

If *f* is allowed to be any Euclidean function, then the list of possible *D* values for which the domain is Euclidean is not yet known.[154] The first example of a Euclidean domain that was not norm-Euclidean (with *D* = 69) was published in 1994.[154] In 1973, Weinberger proved that a quadratic integer ring with *D* > 0 is Euclidean if, and only if, it is a principal ideal domain, provided that the generalized Riemann hypothesis holds.[126]

Noncommutative rings [edit]

The Euclidean algorithm may be applied to noncommutative rings such as the set of Hurwitz quaternions.[127] Let α and β represent two elements from such a ring. They have a common right divisor δ if α = ξδ and β = ηδ for some choice of ξ and η in the ring. Similarly, they have a common left divisor if α = δξ and β = δη for some choice of ξ and η in the ring. Since multiplication is not commutative, there are two versions of the Euclidean algorithm, one for right divisors and one for left divisors.[127] Choosing the right divisors, the first step in finding the gcd(α, β) by the Euclidean algorithm can be written

$$\rho_0 = \alpha - \psi_0\beta = (\xi - \psi_0\eta)\delta$$

where ψ_0 represents the quotient and ρ_0 the remainder. This equation shows that any common right divisor of α and β is likewise a common divisor of the remainder ρ_0. The analogous equation for the left divisors would be

$$\rho_0 = \alpha - \beta\psi_0 = \delta(\xi - \eta\psi_0) \ .$$

With either choice, the process is repeated as above until the greatest common right or left divisor is identified. As in the Euclidean domain, the "size" of the remainder ρ_0 must be strictly smaller than β, and there must be only a finite number of possible sizes for ρ_0, so that the algorithm is guaranteed to terminate.[155]

Most of the results for the GCD carry over to noncommutative numbers. For example, Bézout's identity states that the right gcd(α, β) can be expressed as a linear combination of α and β.[156] In other words, there are numbers σ and τ such that

$$\Gamma_{\text{right}} = \sigma\alpha + \tau\beta$$

The analogous identity for the left GCD is nearly the same:

$$\Gamma_{\text{left}} = \alpha\sigma + \beta\tau \ .$$

Bézout's identity can be used to solve Diophantine equations. For instance, one of the standard proofs of Lagrange's four-square theorem, that every positive integer can be represented as a sum of four squares, is based on quaternion GCDs in this way.[155]

See also [edit]

- Euclidean rhythm, a method for using the Euclidean algorithm to generate musical rhythms

Notes [edit]

- **a.** ^ Some widely used textbooks, such as I. N. Herstein's *Topics in Algebra* and Serge Lang's *Algebra*, use the term "Euclidean algorithm" to refer to Euclidean division.

References [edit]

1. ^ Stark 1978, p. 16
2. ^ Stark 1978, p. 21
3. ^ LeVeque 1996, p. 32
4. ^ LeVeque 1996, p. 31
5. ^ Grossman, J. W. (1990). *Discrete Mathematics*. New York: Macmillan. p. 213. ISBN 0-02-348331-8.
6. ^ [a] [b] Schroeder 2005, pp. 21–22
7. ^ Schroeder 2005, p. 19
8. ^ Ogilvy, C. S.; Anderson, J. T. (1966). *Excursions in number theory*. New York: Oxford University Press. pp. 27–29.
9. ^ [a] [b] Schroeder 2005, pp. 216–219
10. ^ [a] [b] LeVeque 1996, p. 33
11. ^ Stark 1978, p. 25
12. ^ Ore 1948, pp. 47–48
13. ^ Stark 1978, p. 18
14. ^ Stark 1978, pp. 16–20
15. ^ Knuth 1997, p. 320
16. ^ Lovász, L.; Pelikán, J.; Vesztergombi, K. (2003). *Discrete Mathematics: Elementary and Beyond*. New York: Springer-Verlag. pp. 100–101. ISBN 0-387-95584-4.
17. ^ Kimberling, C. (1983). "A Visual Euclidean Algorithm". *Mathematics Teacher* **76**: 108–109.
18. ^ [a] [b] Cohn 1962, pp. 104–110
19. ^ Knuth 1997, pp. 319–320
20. ^ Knuth 1997, pp. 318–319
21. ^ Stillwell 1997, p. 14
22. ^ [a] [b] Ore 1948, p. 43

23. ^ [a] [b] Stewart, B. M. (1964). *Theory of Numbers* (2nd ed.). New York: Macmillan. pp. 43–44. LCCN 64010964.
24. ^ Lazard, D. (1977). "Le meilleur algorithme d'Euclide pour *K*[*X*] et **Z**". *Comptes Rendus Acad. Sci. Paris* **284**: 1–4.
25. ^ [a] [b] Knuth 1997, p. 318
26. ^ [a] [b] Weil, A. (1983). *Number Theory*. Boston: Birkhäuser. pp. 4–6. ISBN 0-8176-3141-0.
27. ^ Jones, A. (1994). "Greek mathematics to AD 300". *Companion encyclopedia of the history and philosophy of the mathematical sciences*. New York: Routledge. pp. 46–48. ISBN 0-415-09238-8.
28. ^ van der Waerden, B. L. (1954). *Science Awakening*. translated by Arnold Dresden. Groningen: P. Noordhoff Ltd. pp. 114–115.
29. ^ von Fritz, K. (1945). "The Discovery of Incommensurability by Hippasus of Metapontum". *Annals of Mathematics* **46** (2): 242–264. doi:10.2307/1969021. JSTOR 1969021.
30. ^ Heath, T. L. (1949). *Mathematics in Aristotle*. Oxford Press. pp. 80–83.
31. ^ Fowler, D. H. (1987). *The Mathematics of Plato's Academy: A New Reconstruction*. Oxford: Oxford University Press. pp. 31–66. ISBN 0-19-853912-6.
32. ^ Becker, O. (1933). "Eudoxus-Studien I. Eine voreuklidische Proportionslehre und ihre Spuren bei Aristoteles und Euklid". *Quellen und Studien zur Geschichte der Mathematik B* **2**: 311–333.
33. ^ [a] [b] Stillwell 1997, p. 31
34. ^ [a] [b] Tattersall 2005, p. 70
35. ^ Rosen 2000, pp. 86–87
36. ^ Ore 1948, pp. 247–248
37. ^ Tattersall 2005, pp. 72, 184–185
38. ^ Tattersall 2005, pp. 72–76
39. ^ [a] [b] Gauss, C. F. (1832). "Theoria residuorum biquadraticorum". *Comm. Soc. Reg. Sci. Gött. Rec.* **4**. Reprinted in Gauss, C. F. (2011). *Werke* **2**. Cambridge Univ. Press. pp. 65–92. doi:10.1017/CBO9781139058230.004 and doi:10.1017/CBO9781139058230.005.
40. ^ Stillwell 1997, pp. 31–32
41. ^ Lejeune Dirichlet 1894, pp. 29–31
42. ^ Richard Dedekind in Lejeune Dirichlet 1894, Supplement XI
43. ^ Stillwell 2003, pp. 41–42
44. ^ Sturm, C. (1829). "Mémoire sur la résolution des équations numériques". *Bull. des sciences de Férussac* **11**: 419–422.
45. ^ Weisstein, Eric W., "Integer Relation", *MathWorld*.
46. ^ Peterson, I. (12 August 2002). "Jazzing Up Euclid's Algorithm". *ScienceNews*.
47. ^ Cipra, B. A. (16 May 2000). "The Best of the 20th Century: Editors Name Top 10 Algorithms" (PDF). *SIAM News* (Society for Industrial and Applied Mathematics) **33** (4).
48. ^ Cole, A. J.; Davie, A. J. T. (1969). "A game based on the Euclidean algorithm and a winning strategy for it". *Math. Gaz.* **53** (386): 354–357. doi:10.2307/3612461. JSTOR 3612461.
49. ^ Spitznagel, E. L. (1973). "Properties of a game based on Euclid's algorithm". *Math. Mag.* **46** (2): 87–92. doi:10.2307/2689037. JSTOR 2689037.
50. ^ Rosen 2000, p. 95
51. ^ Roberts, J. (1977). *Elementary Number Theory: A Problem Oriented Approach*. Cambridge, MA: MIT Press. pp. 1–8. ISBN 0-262-68028-9.
52. ^ Jones, G. A.; Jones, J. M. (1998). "Bezout's Identity". *Elementary Number Theory*. New York: Springer-Verlag. pp. 7–11.
53. ^ Rosen 2000, p. 81
54. ^ Cohn 1962, p. 104
55. ^ Rosen 2000, p. 91
56. ^ Schroeder 2005, p. 23
57. ^ Rosen 2000, pp. 90–93
58. ^ [a] [b] Koshy, T. (2002). *Elementary Number Theory with Applications*. Burlington, MA: Harcourt/Academic Press. pp. 167–169. ISBN 0-12-421171-2.
59. ^ Bach, E.; Shallit, J. (1996). *Algorithmic number theory*. Cambridge, MA: MIT Press. pp. 70–73. ISBN 0-262-02405-5.
60. ^ Stark 1978, pp. 26–36
61. ^ [a] [b] Ore 1948, p. 44
62. ^ Stark 1978, pp. 281–292
63. ^ Rosen 2000, pp. 119–125
64. ^ Schroeder 2005, pp. 106–107
65. ^ Schroeder 2005, pp. 108–109
66. ^ Rosen 2000, pp. 120–121
67. ^ Stark 1978, p. 47
68. ^ Schroeder 2005, pp. 107–109
69. ^ Stillwell 1997, pp. 186–187
70. ^ Schroeder 2005, p. 134

71. ^ Moon, T. K. (2005). *Error Correction Coding: Mathematical Methods and Algorithms*. John Wiley and Sons. p. 266. ISBN 0-471-64800-0.
72. ^ Rosen 2000, pp. 143–170
73. ^ Schroeder 2005, pp. 194–195
74. ^ Graham, R.; Knuth, D. E.; Patashnik, O. (1989). *Concrete mathematics*. Addison-Wesley. p. 123.
75. ^ Vinogradov, I. M. (1954). *Elements of Number Theory*. New York: Dover. pp. 3–13.
76. ^ Crandall & Pomerance 2001, pp. 225–349
77. ^ Knuth 1997, pp. 369–371
78. ^ Shor, P. W. (1997). "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". *SIAM Journal on Scientific and Statistical Computing* **26**: 1484. doi:10.1137/s0097539795293172.
79. ^ Dixon, J. D. (1981). "Asymptotically fast factorization of integers". *Math. Comput.* **36** (153): 255–260. doi:10.2307/2007743. JSTOR 2007743.
80. ^ Lenstra, H. W., Jr. (1987). "Factoring integers with elliptic curves". *Annals of Mathematics* **126** (3): 649–673. doi:10.2307/1971363. JSTOR 1971363.
81. ^ Knuth 1997, pp. 380–384
82. ^ Knuth 1997, pp. 339–364
83. ^ Reynaud, A.-A.-L. (1811). *Traité d'arithmétique à l'usage des élèves qui se destinent à l'École Polytechnique* (6th ed.). Paris: Courcier. Note 60, p. 34. As cited by Shallit (1994).
84. ^ Finck, P.-J.-E. (1841). *Traité élémentaire d'arithmétique à l'usage des candidats aux écoles spéciales*. Derivaux.
85. ^ Shallit, J. (1994). "Origins of the analysis of the Euclidean algorithm". *Historia Math.* **21**: 401–419. doi:10.1006/hmat.1994.1031.
86. ^ Lamé, G. (1844). "Note sur la limite du nombre des divisions dans la recherche du plus grand commun diviseur entre deux nombres entiers". *Comptes Rendus Acad. Sci.* **19**: 867–870.
87. ^ Grossman, H. (1924). "On the Number of Divisions in Finding a G.C.D". *The American Mathematical Monthly* **31** (9): 443. doi:10.2307/2298146. JSTOR 2298146.
88. ^ Honsberger, R. (1976). *Mathematical Gems II*. The Mathematical Association of America. pp. 54–57. ISBN 0-88385-302-7.
89. ^ [a] [b] Knuth 1997, pp. 257–261
90. ^ [a] [b] [c] Crandall & Pomerance 2001, pp. 77–79, 81–85, 425–431
91. ^ [a] [b] Möller, N. (2008). "On Schönhage's algorithm and subquadratic integer gcd computation" (PDF). *Mathematics of Computation* **77** (261): 589–607. Bibcode:2008MaCom..77..589M. doi:10.1090/S0025-5718-07-02017-0.
92. ^ [a] [b] [c] Knuth 1997, p. 344
93. ^ Ore 1948, p. 45
94. ^ [a] [b] Knuth 1997, p. 343
95. ^ Mollin 2008, p. 21
96. ^ LeVeque 1996, p. 35
97. ^ Mollin 2008, pp. 21–22
98. ^ Knuth 1997, p. 353
99. ^ Knuth 1997, p. 357
100. ^ Tonkov, T. (1974). "On the average length of finite continued fractions". *Acta arithmetica* **26**: 47–57.
101. ^ Weisstein, Eric W., "Porter's Constant", *MathWorld*.
102. ^ Porter, J. W. (1975). "On a Theorem of Heilbronn". *Mathematika* **22**: 20–28. doi:10.1112/S0025579300004459.
103. ^ Knuth, D. E. (1976). "Evaluation of Porter's Constant". *Computers and Mathematics with Applications* **2** (2): 137–139. doi:10.1016/0898-1221(76)90025-0.
104. ^ Dixon, J. D. (1970). "The Number of Steps in the Euclidean Algorithm". *J. Number Theory* **2** (4): 414–422. doi:10.1016/0022-314X(70)90044-2.
105. ^ Heilbronn, H. A. (1969). "On the Average Length of a Class of Finite Continued Fractions". In Paul Turán. *Number Theory and Analysis*. New York: Plenum. pp. 87–96. LCCN 76016027.
106. ^ Knuth 1997, p. 354
107. ^ [a] [b] Norton, G. H. (1990). "On the Asymptotic Analysis of the Euclidean Algorithm". *Journal of Symbolic Computation* **10**: 53–58. doi:10.1016/S0747-7171(08)80036-3.
108. ^ Knuth 1997, p. 355
109. ^ Knuth 1997, p. 356
110. ^ Knuth 1997, p. 352
111. ^ Wagon, S. (1999). *Mathematica in Action*. New York: Springer-Verlag. pp. 335–336. ISBN 0-387-98252-3.
112. ^ Cohen 1993, p. 14
113. ^ Cohen 1993, pp. 14–15, 17–18
114. ^ Sorenson, Jonathan P. (2004). "An analysis of the generalized binary GCD algorithm". *High primes and misdemeanours: lectures in honour of the 60th birthday of Hugh Cowie Williams*. Fields Institute Communications **41**. Providence, RI: American Mathematical Society. pp. 327–340. MR 2076257. "The algorithms that are used the most in practice today [for computing greatest common divisors] are probably the binary algorithm and Euclid's algorithm for smaller numbers, and either Lehmer's algorithm or Lebealean's version

of the *k*-ary GCD algorithm for larger numbers."
115. ^ Knuth 1997, pp. 321–323
116. ^ Stein, J. (1967). "Computational problems associated with Racah algebra". *Journal of Computational Physics* **1** (3): 397–405. Bibcode:1967JCoPh...1..397S. doi:10.1016/0021-9991(67)90047-2.
117. ^ Knuth 1997, p. 328
118. ^ Lehmer, D. H. (1938). "Euclid's Algorithm for Large Numbers". *The American Mathematical Monthly* **45** (4): 227–233. doi:10.2307/2302607. JSTOR 2302607.
119. ^ Sorenson, J. (1994). "Two fast GCD algorithms". *J. Algorithms* **16**: 110–144. doi:10.1006/jagm.1994.1006.
120. ^ Weber, K. (1995). "The accelerated GCD algorithm". *ACM Trans. Math. Soft.* **21**: 111–122. doi:10.1145/200979.201042.
121. ^ Aho, A.; Hopcroft, J.; Ullman, J. (1974). *The Design and Analysis of Computer Algorithms*. New York: Addison–Wesley. pp. 300–310. ISBN 0-201-00029-6.
122. ^ Schönhage, A. (1971). "Schnelle Berechnung von Kettenbruchentwicklungen". *Acta Informatica* **1** (2): 139–144. doi:10.1007/BF00289520.
123. ^ Cesari, G. (1998). "Parallel implementation of Schönhage's integer GCD algorithm". In G. Buhler. *Algorithmic Number Theory: Proc. ANTS-III, Portland, OR*. Lecture notes in Computer Science **1423**. New York: Springer-Verlag. pp. 64–76.
124. ^ Stehlé, D.; Zimmermann, P. (2005). "Gal's accurate tables method revisited". *Proceedings of the 17th IEEE Symposium on Computer Arithmetic (ARITH-17)*. Los Alamitos, CA: IEEE Computer Society Press.
125. ^ [a] [b] [c] Lang, S. (1984). *Algebra* (2nd ed.). Menlo Park, CA: Addison–Wesley. pp. 190–194. ISBN 0-201-05487-6.
126. ^ [a] [b] Weinberger, P. "On Euclidean rings of algebraic integers". *Proc. Sympos. Pure Math.* **24**: 321–332.
127. ^ [a] [b] [c] Stillwell 2003, pp. 151–152
128. ^ Boyer, C. B.; Merzbach, U. C. (1991). *A History of Mathematics* (2nd ed.). New York: Wiley. pp. 116–117. ISBN 0-471-54397-7.
129. ^ Cajori, F, (1894). *A History of Mathematics*. New York: Macmillan. p. 70. ISBN 0-486-43874-0.
130. ^ Joux, Antoine (2009). *Algorithmic Cryptanalysis*. CRC Press. p. 33. ISBN 9781420070033.
131. ^ Fuks, D. B.; Tabachnikov, Serge (2007). *Mathematical Omnibus: Thirty Lectures on Classic Mathematics*. American Mathematical Society. p. 13. ISBN 9780821843161.
132. ^ Darling, David (2004). "Khintchine's constant". *The Universal Book of Mathematics: From Abracadabra to Zeno's Paradoxes*. John Wiley & Sons. p. 175. ISBN 9780471667001.
133. ^ Williams, Colin P. (2010). *Explorations in Quantum Computing*. Springer. pp. 277–278. ISBN 9781846288876.
134. ^ Cox, Little & O'Shea 1997, pp. 37–46
135. ^ Schroeder 2005, pp. 254–259
136. ^ Grattan-Guinness, Ivor (1990). *Convolutions in French Mathematics, 1800-1840: From the Calculus and Mechanics to Mathematical Analysis and Mathematical Physics. Volume II: The Turns*. Science Networks: Historical Studies **3**. Basel, Boston, Berlin: Birkhäuser. p. 1148. ISBN 9783764322380. "Our subject here is the 'Sturm sequence' of functions defined from a function and its derivative by means of Euclid's algorithm, in order to calculate the number of real roots of a polynomial within a given interval".
137. ^ Hairer, Ernst; Nørsett, Syvert P.; Wanner, Gerhard (1993). "The Routh–Hurwitz Criterion". *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics **8** (2nd ed.). Springer. pp. 81ff. ISBN 9783540566700.
138. ^ [a] [b] [c] Stillwell 2003, pp. 101–116
139. ^ [a] [b] [c] Hensley, Doug (2006). *Continued Fractions*. World Scientific. p. 26. ISBN 9789812564771.
140. ^ Dedekind, Richard (1996). *Theory of Algebraic Integers*. Cambridge Mathematical Library. Cambridge University Press. pp. 22–24. ISBN 9780521565189.
141. ^ Johnston, Bernard L.; Richman, Fred (1997). *Numbers and Symmetry: An Introduction to Algebra*. CRC Press. p. 44. ISBN 9780849303012.
142. ^ Adams, William W.; Goldstein, Larry Joel (1976). *Introduction to Number Theory*. Prentice-Hall. Exercise 24, p. 205. ISBN 9780134912820. "State and prove an analogue of the Chinese remainder theorem for the Gaussian integers."
143. ^ Stark 1978, p. 290
144. ^ Cohn 1962, pp. 104–105
145. ^ Lauritzen, Niels (2003). *Concrete Abstract Algebra: From Numbers to Gröbner Bases*. Cambridge University Press. p. 130. ISBN 9780521534109.
146. ^ Lauritzen (2003), p. 132
147. ^ Lauritzen (2003), p. 161
148. ^ [a] [b] Sharpe, David (1987). *Rings and Factorization*. Cambridge University Press. p. 55. ISBN 9780521337182.
149. ^ Sharpe (1987), p. 52
150. ^ Lauritzen (2003), p. 131
151. ^ Lamé, G. (1847). "Mémoire sur la résolution, en nombres complexes, de l'équation $A^n + B^n + C^n = 0$". *J. Math. Pures Appl.* **12**: 172–184.
152. ^ Edwards, H. (2000). *Fermat's last theorem: a genetic introduction to algebraic number theory*. Springer. p. 76.
153. ^ LeVeque, W. J. (2002) [1956]. *Topics in Number Theory, Volumes I and II*. New York: Dover Publications.

pp. II:57,81. ISBN 978-0-486-42539-9. Zbl 1009.11001.
154. ^ [a] [b] Clark, D. A. (1994). "A quadratic field which is Euclidean but not norm-Euclidean". *Manuscripta Mathematica* **83**: 327–330. doi:10.1007/BF02567617. Zbl 0817.11047.
155. ^ [a] [b] Davidoff, Giuliana; Sarnak, Peter; Valette, Alain (2003). "2.6 The Arithmetic of Integer Quaternions". *Elementary Number Theory, Group Theory and Ramanujan Graphs*. London Mathematical Society Student Texts **55**. Cambridge University Press. pp. 59–70. ISBN 9780521531436.
156. ^ Ribenboim, Paulo (2001). *Classical Theory of Algebraic Numbers*. Universitext. Springer-Verlag. p. 104. ISBN 9780387950709.

Bibliography [edit]

- Cohen, H. (1993). *A Course in Computational Algebraic Number Theory*. New York: Springer-Verlag. ISBN 0-387-55640-0.
- Cohn, H. (1962). *Advanced Number Theory*. New York: Dover. ISBN 0-486-64023-X.
- Cox, D.; Little, J.; O'Shea, D. (1997). *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra* (2nd ed.). Springer-Verlag. ISBN 0-387-94680-2.
- Crandall, R.; Pomerance, C. (2001). *Prime Numbers: A Computational Perspective* (1st ed.). New York: Springer-Verlag. ISBN 0-387-94777-9.
- Lejeune Dirichlet, P. G. (1894). Dedekind, Richard, ed. *Vorlesungen über Zahlentheorie (Lectures on Number Theory)* (in German). Braunschweig: Vieweg. LCCN 03005859. OCLC 490186017.. See also Vorlesungen über Zahlentheorie
- Knuth, D. E. (1997). *The Art of Computer Programming, Volume 2: Seminumerical Algorithms* (3rd ed.). Addison–Wesley. ISBN 0-201-89684-2.
- LeVeque, W. J. (1996) [1977]. *Fundamentals of Number Theory*. New York: Dover. ISBN 0-486-68906-9.
- Mollin, R. A. (2008). *Fundamental Number Theory with Applications* (2nd ed.). Boca Raton: Chapman & Hall/CRC. ISBN 978-1-4200-6659-3.
- Ore, O. (1948). *Number Theory and Its History*. New York: McGraw–Hill.
- Rosen, K. H. (2000). *Elementary Number Theory and its Applications* (4th ed.). Reading, MA: Addison–Wesley. ISBN 0-201-87073-8.
- Schroeder, M. (2005). *Number Theory in Science and Communication* (4th ed.). Springer-Verlag. ISBN 0-387-15800-6.
- Stark, H. (1978). *An Introduction to Number Theory*. MIT Press. ISBN 0-262-69060-8.
- Stillwell, J. (1997). *Numbers and Geometry*. New York: Springer-Verlag. ISBN 0-387-98289-2.
- Stillwell, J. (2003). *Elements of Number Theory*. New York: Springer-Verlag. ISBN 0-387-95587-9.
- Tattersall, J. J. (2005). *Elementary Number Theory in Nine Chapters*. Cambridge: Cambridge University Press. ISBN 978-0-521-85014-8.

External links [edit]

- Demonstrations of Euclid's algorithm
- Weisstein, Eric W., "Euclidean Algorithm", *MathWorld*.
- Euclid's Algorithm at cut-the-knot
- Euclid's algorithm at PlanetMath.org.
- The Euclidean Algorithm at MathPages
- Euclid's Game at cut-the-knot
- Music and Euclid's algorithm

Wikimedia Commons has media related to ***Euclidean algorithm***.

v · t · e	**Number-theoretic algorithms**
Primality tests	AKS test · APR test · Baillie–PSW · ECPP test · Elliptic curve · Pocklington · Fermat · Lucas · *Lucas–Lehmer* · *Lucas–Lehmer–Riesel* · *Proth's theorem* · *Pépin's* · Quadratic Frobenius test · Solovay–Strassen · Miller–Rabin
Prime-generating	Sieve of Atkin · Sieve of Eratosthenes · Sieve of Sundaram · Wheel factorization
Integer factorization	Continued fraction (CFRAC) · Dixon's · Lenstra elliptic curve (ECM) · Euler's · Pollard's rho · $p-1$ · $p+1$ · Quadratic sieve (QS) · General number field sieve (GNFS) · *Special number field sieve (SNFS)* · Rational sieve · Fermat's · Shanks' square forms · Trial division · Shor's
Multiplication	Ancient Egyptian · Long · Karatsuba · Toom–Cook · Schönhage–Strassen · Fürer's
Discrete logarithm	Baby-step giant-step · Pollard rho · Pollard kangaroo · Pohlig–Hellman · Index calculus · Function field sieve
Greatest common divisor	Binary · **Euclidean** · Extended Euclidean · Lehmer's
Modular square root	Cipolla · Pocklington's · Tonelli–Shanks
Other algorithms	Chakravala · Cornacchia · Integer relation · Integer square root · Modular exponentiation · Schoof's
Italics indicate that algorithm is for numbers of special forms · Smallcaps indicate a deterministic algorithm	

Categories: Number theoretic algorithms | Euclid