Euler-Fermat algorithm

Preprint · September 2018

DOI: 10.13140/RG.2.2.34554.75201/1

S. Nikitin
Arizona State University

Euler-Fermat algorithm [*]

Sergey Nikitin

September 28, 2018

Abstract

The paper introduces Euler-Carmichael function $s(r,b)$ and proves that for any two natural numbers $r > 1$ and b there exist nonnegative integers $s = s(r,b) > 0$, ℓ and n such that $r^\ell \cdot (r^{s(r,b)} - 1) = b \cdot n$ where $s(r,b)|m$ for any m such that $r^k \cdot (r^m - 1) = 0 \mod b$ where k is a nonnegative integer. The paper also develops a new numerical procedure for computing $s(r,b)$, ℓ and n. The procedure is named as Euler-Fermat algorithm. It has upper complexity estimate $O(\lambda(b) \cdot \log(b))$ as $b \to \infty$ where b and r are coprime and $\lambda(b)$ is Carmichael function.

1 Introduction

Fermat's "little theorem" was formulated in 17th century [1] without a proof,

$$r^{p-1} = 1 \mod p \tag{1}$$

for any prime number p and any natural number r not divisible by p. L. Euler proved this statement in the middle of 18th century [2],[3] and also eliminated primality requirement for p. Euler theorem:

$$r^{\varphi(b)} = 1 \mod b \tag{2}$$

for any two coprime integers r and b. The notation similar to $\varphi(b)$ was first used by H.F. Gauss in the middle of 19th century [4]. J.J. Sylvester introduced the term Euler totient function for $\varphi(b)$ in 19th century [7]. While there are numerous proofs of Fermat's little theorem (1) exploiting primality of p (see [3],[9], [10], [11], [12]), there are very few for Euler theorem (2) (see e.g. [5], [8]). Euler theorem can be strengthened [13], and Euler function $\varphi(b)$ in (2) can be replaced with Carmichael function $\lambda(b)$. This publication goes even further along this path and improves (2) as follows.

Let us introduce a new arithmetic function $s(r,b)$ such that for any two natural numbers $r > 1$ and b there exist nonnegative integers $s(r,b) > 0$, ℓ, n and

$$r^\ell \cdot (r^{s(r,b)} - 1) = b \cdot n, \tag{3}$$

[*] School of Mathematical & Statistical Sciences, Arizona State University, P.O. Box 871804, Tempe, AZ 85287-1804; nikitin@asu.edu

Moreover $s(r,b)|\hat{s}$ for any triple $\hat{s}$, $\hat{\ell}$, $\hat{n}$ of nonnegative integers satisfying

$$r^{\hat{\ell}} \cdot (r^{\hat{s}} - 1) = \hat{n} \cdot b.$$

In particular $s(r,b)|\varphi(b)$ and $s(r,b)|\lambda(b)$ for any natural number $r > 1$, where $\varphi(b)$, $\lambda(b)$ are Euler and Carmichael functions respectively. We call $s(r,b)$ Euler-Carmichael function. Asymptotic properties of the average values for $s(r,b)$ as $b \to \infty$ were studied in [6],[14]. This publication presents a new proof of (3) and a new numerical procedure for computing $s(r,b)$, ℓ and n. This procedure is called Euler-Fermat algorithm. If b and r are coprime then its upper complexity estimate is $O(\lambda(b) \cdot \log(b))$ as $b \to \infty$.

2 Euler-Fermat algorithm

We begin this section with Euler-Fermat statement.

$$m^{\varphi(b)} - 1 = 0 \mod b \quad \text{for} \quad (m,b) = 1$$

where (m,b) denotes the greatest common divisor for m and b. $\varphi(b)$ is Euler totient function. In other words, if $(m,b) = 1$ then one can find an integer q such that

$$b \cdot q = m^{\varphi(b)} - 1.$$

In fact the following stronger statement is true. For any two coprime positive integers m and b there exist natural numbers q and $s(m,b)$ such that

$$b \cdot q = m^{s(m,b)} - 1$$

and $s(m,b)|k$ for any k with $m^k = 1 \mod b$. The function $s(m,b)$ is called Euler-Carmichael function. This paper presents an algorithm that calculates q and $s(m,b)$ for any two natural numbers b and $m > 1$.

In order to illustrate the algorithm let us take $m = 2$ and

$$b = 2^3 + 2^2 + 0 \cdot 2 + 1$$

in the binary form

$$1101$$

The algorithm that delivers the value of Euler-Carmichael function $s(2,13)$ and q in

$$13 \cdot q = 2^{s(2,13)} - 1$$

is illustrated in (4).

Example of Euler-Fermat algorithm in numeral system base 2

$$\begin{array}{cccccccccccccc}
 & & & & & & & & 1 & 1 & 0 & 1 & \quad & 1 \\
 & & & & & & & 1 & 1 & 0 & 1 & & & x \\
 & & & & & & 1 & 0 & 0 & 1 & 1 & 1 & & \\
 & & & & & 1 & 1 & 0 & 1 & & & & & x^3 \\
 & & & & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & & \\
 & & & & 1 & 1 & 0 & 1 & & & & & & x^4 \\
 & & & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & & \\
 & & & 1 & 1 & 0 & 1 & & & & & & & x^5 \\
 & & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & \\
1 & 1 & 0 & 1 & & & & & & & & & & x^8 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & &
\end{array} \tag{4}$$

If $x = 2$ then

$$(x^3 + x^2 + 1) \cdot (x^8 + x^5 + x^4 + x^3 + x + 1) = 4095$$
$$2^{12} - 1 = 4095$$

and

$$\begin{aligned} s(2,13) &= 12 \\ n &= x^3 + x^2 + 1 = 13 \\ q &= x^8 + x^5 + x^4 + x^3 + x + 1 = 315 \end{aligned}$$

$$13 \cdot 315 = 4095.$$

Calculations (5) show an application of Euler-Fermat algorithm in base 10 numeral system. It delivers the value $s(10,17)$ for Euler-Carmichael function.

Example of Euler-Fermat algorithm in numeral system base 10

We need to find m and $s(10,17)$ such that

$$17 \cdot m = 10^{s(10,17)} - 1$$

Since

$$3 \cdot 7 = 1 \quad \mod 10$$

we take $m = 3 \cdot n$. Then we need to find n such that

$$51 \cdot n = 10^{s(10,17)} - 1.$$

$$\begin{array}{ccccc}
9\cdot 51 & & & 9\cdot 1 & \\
9\cdot 51 & & & = & 459 \\
4\cdot 51 & & & 4\cdot 10 & \\
459 & + & 4\cdot 10\cdot 51 & = & 2499 \\
5\cdot 51 & & & 5\cdot 10^2 & \\
2499 & + & 5\cdot 10^2\cdot 51 & = & 27999 \\
2\cdot 51 & & & 2\cdot 10^3 & \\
27999 & + & 2\cdot 10^3\cdot 51 & = & 129999 \\
7\cdot 51 & & & 7\cdot 10^4 & \\
129999 & + & 7\cdot 10^4\cdot 51 & = & 3699999 \\
3\cdot 51 & & & 3\cdot 10^5 & \\
3699999 & + & 3\cdot 10^5\cdot 51 & = & 18999999 \\
1\cdot 51 & & & 1\cdot 10^6 & \\
18999999 & + & 10^6\cdot 51 & = & 69999999 \\
3\cdot 51 & & & 3\cdot 10^7 & \\
69999999 & + & 3\cdot 10^7\cdot 51 & = & 1599999999 \\
4\cdot 51 & & & 4\cdot 10^8 & \\
1599999999 & + & 4\cdot 10^8\cdot 51 & = & 21999999999 \\
8\cdot 51 & & & 8\cdot 10^9 & \\
21999999999 & + & 8\cdot 10^9\cdot 51 & = & 429999999999 \\
7\cdot 51 & & & 7\cdot 10^{10} & \\
429999999999 & + & 7\cdot 10^{10}\cdot 51 & = & 3999999999999 \\
6\cdot 51 & & & 6\cdot 10^{12} & \\
3999999999999 & + & 6\cdot 10^{12}\cdot 51 & = & 309999999999999 \\
9\cdot 51 & & & 9\cdot 10^{13} & \\
309999999999999 & + & 9\cdot 10^{13}\cdot 51 & = & 4899999999999999 \\
1\cdot 51 & & & 1\cdot 10^{14} & \\
4899999999999999 & + & 1\cdot 10^{14}\cdot 51 & = & 9999999999999999
\end{array} \qquad (5)$$

If $x = 10$ then

$$\begin{array}{rcl}
17\cdot 3\cdot (9+4\cdot x+5\cdot x^2+2\cdot x^3+7\cdot x^4+3\cdot x^5+x^6+ & & \\
3\cdot x^7+4\cdot x^8+8\cdot x^9+7\cdot x^{10}+6\cdot x^{12}+9\cdot x^{13}+x^{14}) & = & 10^{16}-1
\end{array}$$

For 17 we calculated $m = 588235294117647$ and $s(10,17) = 16$ such that $17\cdot m = 10^{s(10,17)} - 1$.

For a natural number $r > 1$ we introduce r-polynomials of the form

$$p(x) = \sum_{j=0}^{m} p_j \cdot x^j$$

with p_j taking values $0,\ 1,\ \ldots\ r-1$ for $j = 0,\ 1,\ \ldots m$. The sequence

$$p_m\ p_{m-1}\ \cdots p_1\ p_0 \qquad (6)$$

is the representation of the number $p(r)$ in base r numeral system. Z_r denotes the ring of integers modulo r. An integer z is a zero divisor in Z_r if there exists a non-zero

$u \in \mathrm{Z}_r$ and $z \cdot u = 0 \mod r$. Given natural numbers b and $r > 1$ Euler-Fermat algorithm calculates natural numbers k, n, and s such that

$$b \cdot n = r^k \cdot (r^s - 1). \tag{7}$$

Moreover, Euler-Fermat algorithm computes the smallest s for which (7) takes place. s is set to be the value of Euler-Carmichael function $s(r,b)$.

Euler-Fermat Algorithm

1. Set $k = 0$, $n = 1$, $s = 1$. While $(r,b) > 1$ and $b > 1$ do the following.

   Set b equal to $\frac{b}{(r,b)}$.

   Set n to $n \cdot \frac{r}{(r,b)}$.

   Increment k by one.

   If $b = 1$ then terminate the algorithm. If $(r,b) = 1$ then taking $b_0 = b \mod r$ yields $(r,b_0) = 1$ and there exists $a \in \mathrm{Z}_r$ such that $a \cdot b_0 = 1 \mod r$. Let $[a \cdot (r-1)]$ denote $a \cdot (r-1) \mod r$. Introduce an integer q and an r-polynomial $h(x)$. Initialize them as $q = [a \cdot (r-1)]$ and $h(x)$ is the r-polynomial for $[a \cdot (r-1)] \cdot b$.

2. Let

   $$h_{\bar{m}}\, h_{\bar{m}-1}\, \ldots\, h_1\, h_0$$

   be the digital representation of $h(r)$ in base r numeral system. If $h_j = r-1$ for all j then set $s = \bar{m}+1$, $n = n \cdot q$ and terminate the algorithm. Otherwise, let j be the first integer such that $h_j < r-1$. Set $h(x)$ to be the r-polynomial for

   $$h(r) + [(r-1-h_j) \cdot a] \cdot b \cdot r^j \tag{8}$$

   and set

   $$q = q + [(r-1-h_j) \cdot a] \cdot r^j$$

   where

   $$[(r-1-h_j) \cdot a] = (r-1-h_j) \cdot a \mod r.$$

   Repeat step 2.

Discussion of Euler-Fermat algorithm culminates with the following statement.

Theorem 1 *For any two natural numbers b and $r > 1$ Euler-Fermat algorithm calculates in a finite number of steps the nonnegative integers n, k and $s(r,b)$ such that*

$$b \cdot n = r^k \cdot (r^{s(r,b)} - 1)$$

and $s(r,b)|w$ if $r^\ell \cdot (r^w - 1) = 0 \mod b$ for some integer ℓ. Moreover, $k = 0$ if b is neither zero nor a divisor of zero in Z_r.

Proof.

We start with b such that $(b,r)=1$. Let $p(x)$ be the r-polynomial that corresponds to b, $b=p(r)$. Consider an iteration from step 2 of Euler-Fermat algorithm (see (4), (5)). Omitting the largest segment with all digits equal to $r-1$ on the right we observe that the algorithm changes only the segment on the left. The length of this segment is less than or equal to $deg(p)+1$, the number of digital places in the representation of b in base r numeral system. Algorithm does not create strings of zeroes. Therefore there are $r^{deg(p)+1}-1$ such segments and among them one that terminates our algorithm. If the algorithm does not loop then after a finite number of steps it reaches the segment with all digital places occupied by $r-1$.

Suppose $d(x)$ is an r-polynomial that corresponds to the segment that triggers the loop. Then there exist r-polynomials $q_0(x)$, $q_1(x)$ and

$$p(x)\cdot q_0(x) \quad = \quad d(x)\cdot x^k + x^k - 1 \tag{9}$$

$$p(x)\cdot q_1(x)+d(x) \quad = \quad d(x)\cdot x^s + x^s - 1 \tag{10}$$

where k, s are natural numbers and $x=r$. Hence, (9) yields that $p(r)$ and $d(r)+1$ are coprime,

$$(p(r),d(r)+1)=1.$$

It follows from (10) that

$$p(x)\cdot q_1(x)=(d(x)+1)\cdot(x^s-1)$$

and $d(r)+1|q_1(r)$. There exists an r-polynomial $h(x)$ such that

$$q_1(r)=h(r)\cdot(d(r)+1).$$

Thus the algorithm terminates after a finite number of steps,

$$p(r)\cdot h(r)=r^s-1.$$

By construction, the natural number s is the smallest positive integer such that

$$r^s=1 \mod p(r) \tag{11}$$

s is the value of Euler-Carmichael function $s(r,p(r))$. If

$$r^w=1 \mod p(r)$$

and $w>s$ then $w=q\cdot s+u$ where $0\le u<s$. It follows from (11) that

$$r^w=r^{q\cdot s+u}=r^u=1 \mod p(r)$$

Therefore the only possible value for u is 0 and $s|w$.

If b is a zero or a divisor of zero in Z_r then there exist natural numbers m, ℓ and j such that j is neither zero nor a divisor of zero in Z_r and

$$b \cdot m = r^{\ell} \cdot j \tag{12}$$

Since j is not a divisor of zero in Z_r then we already established that there exist s and n such that

$$j \cdot n = r^s - 1$$

Multiplying (12) with n yields

$$b \cdot m \cdot n = r^{\ell} \cdot j \cdot n = r^{\ell} \cdot (r^s - 1).$$

Q.E.D.

Complexity of Euler-Fermat algorithm is related to the value $s(r,n)$ and the complexity of calculating the greatest common divisor (gcd).

Theorem 2 *The complexity of Euler-Fermat algorithm for a fixed value of $r > 1$ is*

$$O((s(r,b) + M(\log(b)) \cdot \log(\log(b))) \log(b)) \textit{ as } b \to \infty$$

where $M(\log(b))$ is the complexity of the chosen multiplication algorithm. Moreover, if $(b,r) = 1$ then the complexity is

$$O(s(r,b)\log(b)) \textit{ as } b \to \infty.$$

Proof.

The complexity of addition is $O(\log(b))$ as $b \to \infty$ (see [15], [16], [17]). Euler-Fermat algorithm terminates when we reach a number with $s(r,b)$ numeral places occupied by $r-1$. At each iteration of step 2 in Euler-Fermat algorithm one performs (8) that has complexity $O(\log(b))$ as $b \to \infty$. One needs less than $s(r,b)$ iterations of (8) in order to complete the algorithm and the complexity estimate

$$O(s(r,b)\log(b)) \text{ as } b \to \infty.$$

follows for $(b,r) = 1$.

One needs not more than $O(\log(b))$ applications of gcd at step 1. It is well-known that the complexity of Knut-Schönhge fast gcd is

$$O(M(\log(b))\log(\log(b))).$$

Q.E.D.

By theorem 1 $s(r,b)|\lambda(b)$ where $\lambda(b)$ is the value of Carmichael function at b. The more rough upper estimate of the complexity is given by

$$O((\lambda(b) + M(\log(b))\log(\log(b)))\log(b)) \text{ as } b \to \infty$$

and for $(b,r) = 1$ we have

$$O(\lambda(b)\log(b)) \text{ as } b \to \infty$$

3 Conclusion

The algorithm presented in this publication was implemented and tested as a part of
http://github.com/mathhobbit/EditCalculateAndChart/releases
see functions EF and FE in EditCalculateAndChart application.

References

[1] P. Fermat, Oeuvres de Fermat, supp. T. I-IV par M. C. de Waard, éd. P. Tannery et C. Henry, Paris, Gauthier-Villars, 1922.

[2] L.Euler, Commentationes arithmeticae collectae, T. 1, 2. St. Petersburg, 1849.

[3] L.Euler, Theoremata circa residua ex divisione potestatum relicta, Novi Commentarii academiae scientiarum Petropolitanae,1761.

[4] C. F. Gauss, tr. Arthur A. Clarke: Disquisitiones Arithmeticae, Yale University Press, 1965

[5] V.I. Arnold, Ergodic and Arithmetical Properties of Geometrical Progression's Dynamics and of its Orbits, Moscow Mathematical Journal, Volume 5, Number 1, January-March, 2005, pages 5-22.

[6] V. Arnold. Number-theoretical turbulence in Fermat-Euler arithmetics and large Young diagrams geometry statistics. J. Math. Fluid Mech., 7(suppl. 1):S4 S50, 2005

[7] J. J. Sylvester, On certain ternary cubic-form equations, American Journal of Mathematics, 2 , 1879, pages 357393

[8] B. Fine, G. Rosenberger, Number Theory, An Introduction via the Distribution of Primes, Birkhäuser,Boston, Basel, Berlin, 2007.

[9] S.W. Golomb, Combinatorial Proof of Fermat's "Little" Theorem, The American Mathematical Monthly, Vol. 63, No. 10, (Dec., 1956), pp. 718

[10] K. Iga, A Dynamical Systems Proof of Fermat's Little Theorem, Mathematics Magazine, 76 (1),(Feb.,2003) pp. 4851

[11] G. Vacca, Intorno alla prima dimostrazione di un teorema di Fermat, Bibliotheca Mathematica, 2nd series,1894, 8(2), pp.46-48

[12] G. Alkauskas, A Curious Proof of Fermat's Little Theorem, American Mathematical Monthly,2009, 116 (4), pp.362364

[13] R.D. Carmichael, The Theory of Numbers, Mathematical Monogrphs ed. M. Merriman and R.S.Woodward, 13, New York, John Wiley and Sons, 1914.

[14] P.Kurlsberg, C. Pomerance, On a Problem of Arnold: the Average Multiplicative Order of a Given Integer,arXiv:1108.5209 [math.NT], 2011, pp.18

[15] D. Knuth, The Art of Computer Programming, Volume 2,Volume 3, Third Edition, Addison-Wesley 1997.

[16] G. Gathen, J.Gerhard, Modern computer algebra (2nd ed.). Cambridge, UK ; New York, NY, USA: Cambridge University Press,2003.

[17] C. Yap, Fundamental problems of algorithmic algebra. New York ; Oxford: Oxford University Press, 2000.