愛知工業大学研究報告
第36号B　平成13年

楕円曲線法による素因数分解に関する実験的考察

An experimental consideration on the elliptic curve factoring method(ECM)

小池慎一†　山住富也††
Shin-ichi KOIKE　Tomiya YAMAZUMI

abstract We show some characters of the elliptic factoring method(ECM). In some trial factoring the Mersenne Number M137, 7's are success and 5's failed. This is its probablistic character. Also we show that for factoring, an order of curve must be decomposited to some small prime factors. At last, using random number instead of prime, a number is factorized.

1　はじめに

計算機が高速化し、より大きな数値の計算が可能になってきた。またネットワークの普及とともに、ＲＳＡ公開鍵方式の暗号や楕円曲線暗号が考案され大きな数を素因数分解する問題が注目されるようになった。

そこで、本報告では楕円曲線法 [1, 2, 3, 4] として知られる素因数分解の方法を採り上げその性質について考察した。はじめに、この方法のヒントとなった$p-1$法 [3] について数値例を示し、ついで楕円曲線法について例を示す。$p-1$法では因数が見つからない場合、テストする数を大きくする以外になく、やがてそれは時間的に不可能になる。しかし、後者では同じ基準で新しいパラメータを与えて繰り返し試行できる。チャンスが多い、という意味で因数分解できる可能性が大きい。例えば、メルセンヌ数M137は10進数42桁で20桁と22桁の2個の因数を持つが、12回試行して成功が7回、失敗が5回である。このような確率的な側面を実験データとして示す。

ついで、楕円曲線上の点の位数がどのように分布するかを小さい数の例で示す。これにより、最大位数が小さい素数の積に分解されることが必須条件であることがわかる。

また、曲線上の点のp倍を計算する時に用いられる平方乗法のアルゴリズムの構造を検討して、乱数を用いても因数発見が可能であること示す。この例では、たまたま少ない計算回数で因数が見つかった。

†愛知工業大学情報科学部（豊田市）
††名古屋文理大学情報文化学部（稲沢市）

2　楕円曲線法よる素因数分解

はじめに、Lenstraが楕円曲線法を考案する際にヒントにしたPollardの$p-1$法について説明し、そのあとで楕円曲線法による素因数分解の手法について述べる。

2.1　Pollardの$p-1$法について

まず、Pollardの$p-1$法のアルゴリズムについて述べる。その後、素因数が求められる原理を説明する。

$p-1$法のアルゴリズムは以下の通りである。
素因数分解する合成数をn、nの素因数の1つをpとする。

- step.1
  整数kをある境界Mより小さなすべての数の積、またはLCMとする。（ただし、後述する数値例ではLCMを用いる。）
- step.2
  2からnの間で任意の整数aを選ぶ。
- step.3
  $a^k \bmod n$を求める。
- step.4
  $d = GCD(a^k \bmod n - 1, n)$なる$d$を求める。

- step.5
  d が n の自明な因数であるならば終了（n の因数 p は d）。そうでない場合、a、k を選びなおしで step.2 からやり直す。

上のアルゴリズムによって n の因数 p が求まる理由は以下の通りである。

k が境界 M 以下のすべての数の LCM、すなわち、k は M 以下のすべての数で割り切れるとする。また、$p-1$ の因数はすべて M より小さな値を持つとする。言い換えると k の因数を並べ替えると $p-1$ を含む。そこで、$k = k' \times (p-1)$ とおけば、フェルマーの小定理により、$a^k = a^{k' \times (p-1)} \equiv (a^{p-1} \bmod p)^{k'} \equiv 1 \bmod p$ となる。すなわち、$p|(a^k - 1)$ となるので、$d = GCD(a^k - 1, n)$ を求めることにより、p が求まる。

$p-1$ 法は $p-1$ に大きな因数を含まれる場合、$a^k = 1 \bmod p$ を満たす k にその大きな因数を含まなくてはならない。その値は未知であるので、求められない場合の選択枝は k を大きくして再試行する以外になく計算時間の点で断念させざるを得ないことが多い。

2.2 楕円曲線法について

2.2.1 楕円曲線とは

楕円曲線 E は次式で表される曲線である。

$$E : y^2 = x^3 + ax + b \quad (1)$$

整数 a, b に対し、素数 p を法とする楕円曲線は次式となる。

$$E : y^2 \equiv x^3 + ax + b \bmod p \quad (2)$$

ここで、$4a^3 + 27b^2 \neq 0 \bmod p$ なる条件を満たすとき、上式の楕円曲線上に存在するすべての有理点 (x, y) に仮想的な点 O（無限遠点）を加えた点の集合は、次の加算について群をなす。

この楕円曲線上の 2 点を $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ とすると、2 点の和 $P_3 = P_1 + P_2$ の座標 (x_3, y_3) は次式となる。

$$x_3 \equiv \lambda^2 - x_1 - x_2 \quad (3)$$

$$y_3 \equiv \lambda(x_1 - x_3) - y_1 \quad (4)$$

ただし、λ は、

$$P_1 \neq P_2 \ : \ \lambda = (y_2 - y_1)(x_2 - x_1) \quad (5)$$

$$P_1 = P_2 \ : \ \lambda = (x_1^2 + a)/2y_1 \quad (6)$$

となる。λ は、$P_1 \neq P_2$ ののとき、P_1, P_2 を通る直線の傾き、$P_1 = P_2$ のとき、点 P_1 における接線の傾きである。また、$x_1 = x_2, y_1 = -y_2$ の場合、点 P_3 は無限遠点 O と定義する。

次に、楕円曲線 E の性質について重要なものを 2 つ挙げる。

- 点の個数（Hasse の定理）
  楕円曲線 E の点の個数を N とすると、その値は以下の不等式で与えられる。

  $$p + 1 - 2\sqrt{p} < N < p + 1 + 2\sqrt{p} \quad (7)$$

  N の値は式 (1) のパラメータ a, b に依存する。

- 位数
  楕円曲線 E 上の点 P を m 個加えるとき、すなわち mP を計算すると、$mP = O$（無限遠点）となるような最小の m の値をその点の位数と言う。mP の計算においては、λ の分母が p と互いに素とならない場合は分母の逆元は存在しない。

  パラメータ a, b を与えられた楕円曲線 $E(a, b)$ 上の点は様々な位数を持つ点が存在する。大きな位数が合成数である場合、位数の定義からして、その位数を構成する点の中にその位数の素因数を位数とする点が存在する。位数 m が素数であるとき mP を計算するとそれは無限遠点になる。

  パラメータ a, b の値により、因数として含まれる素数の大きさに較べて小さな (10 進数で 5-6 桁程度) 素数の位数を持つ点が含まれている場合にはその因数が検出可能になる。この性質を利用して素数の検出を行うのが楕円曲線法である。大きな素数の位数の場合には時間的な制約もあって検出を試みるよりも、新しいパラメータ a, b で再度テストする。

2.2.2 楕円曲線法による素因数分解のアルゴリズム

Lenstraによる楕円曲線法のアルゴリズムは以下の通りである。

素因数分解する合成数をn、nの素因数の1つをpとする。

- step.1
  Mを定める。
  本実験では、Lを目標とする素因数pについて選択し、Mを次式で定める。

$$M = \prod_{p:\text{素数}, p^e \leq L < p^{e+1}} p^e$$

- step.2
  楕円曲線$E: y^2 \equiv x^3 + ax + b \bmod n$のパラメータ$a, b$を定める。
- step.3
  E上の点$P(x, y)$を選ぶ。
- step.4
  $MP \bmod n$を計算する。
- step.5
  $MP \bmod n$の計算途中に、λの分母が0となり割算ができなくなれば、その分母からnの約数pが求まる。$MP \bmod n$を順次計算しても素因数分解できないとき、step.2へ戻り、パラメータを変更して試行を繰り返す。

$p-1$法は素因数分解できないときはMを大きくしていくしかないので、計算時間もそれによって長くなる。それに対して楕円曲線法では、Eのパラメータa, bを変更し試行を繰り返すことができるため、$p-1$法に対して計算時間が短くすむ。

2.3 数値例：$p-1$法と楕円曲線法との比較

2.3.1 $p-1$法の数値例

$p-1$法により、以下に示す3個の数について因数分解を試みた。例1と例2は2個の素数p, qの積で、例3はメルセンヌ数M103である。

因数pおよびqと$p-1, q-1$の値を以下に示す。

- 例1

$$\begin{aligned} n &= 39998253218964413 \\ &= 199995959 \times 199995307 = pq \end{aligned}$$

$$\begin{aligned} p-1 &= 199995959 - 1 \\ &= 2 \times 59 \times 97 \times 101 \times 173 \\ q-1 &= 199995307 - 1 \\ &= 2 \times 3 \times 7 \times 83 \times 103 \times 557 \end{aligned}$$

- 例2

$$\begin{aligned} n &= 39999875600088977 \\ &= 199999777 \times 199999601 = pq \end{aligned}$$

$$\begin{aligned} p-1 &= 199999777 - 1 \\ &= 2^5 \times 3 \times 19^2 \times 29 \times 199 \\ q-1 &= 199999601 - 1 \\ &= 2^4 \times 5^2 \times 31^2 \times 127^2 \end{aligned}$$

- 例3 (M103)

$$\begin{aligned} 2^{103} - 1 &= 10141204801825835211973625643007 \\ &= 2550183799 \times 3976656429941438590393 = pq \\ p-1 &= 2 \times 3 \times 83^2 \times 103 \times 599 \\ q-1 &= 2^3 \times 3 \times 103 \times 149 \times 4657 \times 71429 \times 32456563 \end{aligned}$$

これらの数について、$p-1$法で因数分解を試みた結果を表1に示す。

表1: $p-1$による因数分解の結果

n	$p-1$と$q-1$の最大因数	
39998253218964113	173,557	失敗
39999875600088977	199,127	成功
M103	599,32456563	成功

例1は、$p-1$, $q-1$の最大の因数が分かっているので、Mを577としてテストしてみた。しかし、aを2から10000まで変化させて試行したが、因数が見つからなかった。その理由については後述する。

例2はMを199として試行したところ、$a=29$で因数p=199999777が得られた。

例 3 では、$M = 599$ の時に $a = 155$ で因数 2550183799 が得られた。

一見素因数分解が簡単に見える例 1 が失敗したのは以下の理由による。一般に M の値は大きい方がよいとされているが、この例では、2 個の因数 p と q の $p-1$ および $q-1$ の因数の積を k が同時に含んでしまっている。したがって、$a^k \bmod n \equiv 1$ となり、$GCD(a^k \bmod n - 1, n) = GCD(0, n) = n$ となってしまい因数は検出されない。M の値として例 1 については、173 と 577 の間の値が選択されたときにのみ、成功する。

例 3 は例 1 、 2 より桁数が大きいが、素因数 p の $p-1$ の因数が小さいので、素因数分解された。

このように $p-1$ 法では、M の値の選択についての自由度は少ない。当たればきれいに分解できるが、より大きな M を要求される場面では現実的に素因数分解不可能となる。

2.3.2 楕円曲線法の数値例

以下の 4 個のメルセンヌ数

$$M101(= 2^{101} - 1),\ M103(= 2^{103} - 1),$$

$$M137(= 2^{137} - 1),\ M139(= 2^{139} - 1)$$

について実験を行った。

これらの数の全桁数と因数の桁数をまとめると表 2 のようになる。M101 と M103 の数字の桁数はほぼ同じであるが、小さい因数の桁数が M101 の方が大きく因数分解は困難である。同様に、M137 と M139 では M137 の方が因数分解は困難である。

表 2: テストに用いた数の全桁数と因数の桁数

数	全桁数	因数の桁数
M101	31	13 ×18
M103	32	10 ×22
M137	42	20 ×22
M139	42	13 ×30

結果を表 3 ～ 6 に示す。M101,M103,M139 はすべて成功したが、M137 はわれわれのプログラムでは 200 ～ 300 時間かかることが多く、コンピュータの障害などにより途中で中断したことが何度もあった。それも試行回数に含めた。予想通り、M137 は因数分解が困難であることが実証された。

表 3: M101 の素因数分解の試行回数と正否

No.	試行回数	正否
1	6	成功
2	11	成功
3	5	成功
4	13	成功
5	18	成功
6	24	成功
7	41	成功
8	75	成功
9	66	成功
10	158	成功
平均試行回数	417/10(= 4.17)	

表 4: M103 の素因数分解の試行回数と正否

No.	試行回数	正否
1	1	成功
2	5	成功
3	18	成功
4	20	成功
5	2	成功
6	17	成功
7	9	成功
8	3	成功
9	1	成功
平均試行回数	76/9 = (8.44)	

注　失敗とは、端末の予期せぬ切断、コンピュータの停止などで試行の途中で強制的にうち切られたもの。

表 5: M137 の素因数分解の試行回数と正否

No.	試行回数	正否
1	7	成功
2	8	成功
3	387	成功
4	164	成功
5	123	失敗
6	205	失敗
7	286	失敗
8	163	失敗
9	163	失敗
10	1023	成功
11	583	成功
12	721	成功
平均試行回数	3833/7(= 547.6)	

表 6: M139 の素因数分解の試行回数と正否

No.	試行回数	正否
1	1	成功
2	13	成功
3	22	成功
4	5	成功
5	1	失敗
6	3	失敗
7	81	失敗
8	6	失敗
9	12	失敗
10	2	成功
11	20	成功
平均試行回数	166/11(= 15.1)	

3 考察

3.1 楕円曲線法に素数検出の仕組みについて

楕円曲線法では、与えられた曲線の位数 (order) の点が無限遠点になることによって因数が発見される。そこで曲線に対する位数がどのような分布であるかを小さい素数について調べてみた。

例として素数 $q = 347$ を取り上げた。この値は、整数型のサイズが 32 ビットの C 言語で容易にプログラミング可能であること、また、楕円曲線のすべての点を数え上げることが物理的に可能であることから選んだ。

例 1 $q = 347, a = 310, b = 198, N = 325$

このパラメータの場合には、位数 2 の点が 1 個あるが、その他は位数が 163 あるいは 326 である（表 7）。したがって、位数が 163 の点で $p = 163$ でテストすれば $q = 347$ が検出される。ところで、326=2×163 であり、アルゴリズムのはじめのステップで $p = 2$ でテストされるので 2 倍された点に移動する。その時点で位数 163 の点になりこれは巡回群になっているのであとはすべて位数 163 の点をめぐる。したがって、

表 7: $a = 310, b = 198, N = 325$

位数	個数
2	1 †
163	162†
326	162
合計	325

† は巡回群

$p = 163$ で検出される。

例 2 $q = 347,\ a = 94,\ b = 225,\ N = 347$

この場合には、最大の位数が 174 である（表 8）。この数は $174 = 2 \times 3 \times 29$ と素因数分解される。したがって、位数 2, 3, 29 の巡回群を持つ。例えば点 $P = (345, 305)$ は位数 87 の点であるが、$2^7 \times 3$ 倍すると位数 29 の巡回群上の点に飛び込み、以降その郡内で移動する。そして、$p = 29$ のテストの時に因数が検出される。

上の 2 つの例で確かめられたように、最大位数の素因数分解された位数を持つ巡回群があって、その群の位数とテストする p の値が一致したときに因数が検出される。もし、最大位数が大きな素数にしか素因数分解されないと楕円曲線法では検出されない。しかし、パラメータ a, b を変化させることにより、最大位数は変化する

表 8: $a = 94, b = 225, N = 347$

位数	個数
2	3 †
3	2 †
6	6
29	28 †
58	84
87	56
174	168
合計	347

† は巡回群

ので、$p-1$ 法とは異なり、試行の機会はずっと多い。

3.2 p の代わりに乱数を用いる試み

楕円曲線上の点を m 倍する計算では平方乗法と呼ばれる高速算法が用いられている。われわれが用いたのは、下位ビットから判定する方法である。それによれば、例えば乱数の実現値 18935 は

$18935 = 1+18934 = 3+18932 = 7+18928 = 23 + 18912 = 55 + 18880 = 119 + 18816 = 247 + 18688 = 503 + 18432 = 2551 + 16384$

のような順で計算される。2 のべき乗はすべて計算されるが、それ以外に 1, 3, 7, 23, 55, 119, 247, 503, 2551 の点を算出する。ここで、3, 7, 23, 503, 2551 は素数、$55 = 11 \times 5, 119 = 7 \times 11, 247 = 13 \times 19$ であり、計算開始の点が位数 3, 5, 7, 23, 11, 13, 19, 503, 2551 の巡回群のいずれかに属していれば検出される。

このように、途中の計算で 2 のべき乗の点および、その他の奇数あるいは奇素数を算出するので、計算開始点がそれらの位数を持つ巡回群に属するか否かの判定がなされる。そこで、同じアルゴリズムで素数 $p = 2, 3, 5, \ldots$ の代わりに 16 ビットの奇数乱数に置き換えてテストしてみた。その結果、M101, M103 では素数を発見できた。試行回数は少ないが、平均試行回数を示すと

M103 の場合: 平均試行回数=17/4=4.25

M101 の場合: 平均試行回数=8/3=2.7

となった。

ここで、試行内での繰り返しの上限を素数 p を用いた場合の上限 L とした。素数の場合には、$L = 20000$ 以下の素数密度がほぼ 1/10 であるのに対して、乱数を用いる場合には L そのものが繰り返し回数になる。したがって、この値の 10 倍と上の結果を較べるのが妥当である。

すると、この実験にかんする限り、M101 では乱数による方法の方が試行回数は少なくて済んだ。

反面、桁数の多い M137 では 65 万回の試行でも因数を発見できず、試行を打ち切った。これは、テストする楕円曲線の位数と使用している乱数との相互作用もあるので、これだけのテストでは乱数による方法の是非をにわかには決めがたい。

以上の考察から、テストする p の系列について、よりよい方法が見付かるかも知れないと予想される。

参考文献

[1] 小山謙二:高速だ円曲線法による素因数分解, 電子情報通信学会論文誌 Vol.J70-D, No.12, pp.2730-2738 (1787.12).

[2] 小山謙二, 静谷啓樹:素因数分解と離散対数アルゴリズム, 情報処理,Vol.34,No.2, pp.157-169(1993.2).

[3] J.H. シルヴァーマン、J. テイト (足立恒雄, 木田雅成, 小松啓一, 田谷久雄訳):楕円曲線入門, シュプリンガー・ファアラーク東京 (1995)

[4] 木田祐司, 牧野潔夫, コンピュータ整数論, 日本評論社 (1994).

(受理　平成13年 3 月19日)