岡山理科大学紀要 第48号 A pp. 5-9(2012)

$a^{p-1} \equiv 1 \pmod{p^2}$ の新しい計算と解

石井 大輔・森 義之*・澤江 隆一*

岡山理科大学大学院理学研究科修士課程応用数学専攻
*岡山理科大学理学部応用数学科

(2012年10月1日受付、2012年11月1日受理)

1．はじめに

1－1　計算の目的

今から２０年ほど前に本論文のタイトル中にある以下の式

$$a^{p-1} \equiv 1 \pmod{p^2} \qquad (1)$$

の解は、フェルマーの最終定理にまつわる数論的な観点で研究され、Brillhart, Tonascia and Weinberger[1]は$2 \le a \le 99$ かつ $3 \le p < 10^6$ の範囲での解の一覧表を作成した。その後、 Montgomery[2]は a の範囲は同じで、$3 \le p < 2^{32}$ までの計算を行った。

私達がこの計算に興味を持つのは、後藤と大野[3]による奇数の完全数の最大素因子についての証明の中で、受容可能な円分数を決定する際にMontgomeryの計算結果を使って補題のひとつを証明しているからである(円分数の定義は [3]参照)。

奇数の完全数は存在しないであろうと予想されているが、現在まで非存在の証明はなく、その研究の方向の重要なもののひとつとして「奇数の完全数の最大素因子」がある。 最大素因子について、1998年に Hagis and Cohen は 10^6 以上であることを、2003年に Jenkins は 10^7 であることを、2008年に Goto and Ohno は 10^8 を証明した。そして、青木(島根大学)と筆者らは2012年に奇数の完全数の最大素因子は 10^9 以上であることを証明した。これに引き続いた最大素因子の証明において、(1)の解の計算は補題として寄与すると考えられる。

なお、本論文の内容とは間接的な関係ではあるが、最大素因子の更新は奇数の完全数の下限等についても更新となると考えられるので、この分野の研究に取っては重要な計算である。

1－2　プログラミングに関して

Corei7, Windows7上で動作するx86-64のgcc を使い計算を行った。ただ、計算のコアの部分はアセンブリ言語で作成し、コンパイラ・リンカーはgccを利用した。Cの関数との値渡しはそのままレジスターを使った。

Corei7は64ビットアーキテクチャーであるが、 $p<10^{10}$ としても、(1)の計算には 64x4ビット幅が必要となるので、今回の計算では$\bmod p^2$の元を$a\,p + b$ と表し、a,b をレジスターで記録して計算を行った。ここで、当然$0 \le a < p,\ 0 \le b < p$である。

冪乗の計算はバイナリ法及び下位桁から計算する方式を採用し、２つ積の剰余計算は次式が成立するので、

$$(a_1 p + b_1)(a_2 p + b_2) \bmod p^2 = \left((a_1 b_2 + a_2 b_1) p + a_1 a_2 + b_1 b_2\right) \bmod p^2 \qquad (2)$$

右辺をコーディングした。

アセンブリ言語のソース

```
        .text
.globl fermat
        .def    fermat;  .scl    2;       .type    32;      .endef
fermat:
       pushq   %r8
```

```
        pushq   %r9
        pushq   %r10
        pushq   %r11
        pushq   %r12
################  rcx : a     rdx p=rbx
# rax,rbx,rcx,rdx,rsi,rdi,rbp,rsp
# r8,r9,r10,r11,r12,r13
#  pw=1;goto y
# x:  a = (a*a)%m;
# y:  if( n & 1) pw= (a*pw)%m;
#     if(n>>1) goto x
#  a  %rsi:rcx   pw %r9:%r8 p=rbx
#  a^(p-1) = 1 mod p^2
        leaq      -1(%rdx), %r10
        movq      %rdx,%rbx
        movl      $0, %r9d
        movl      $1, %r8d
        movq      %r9,%rsi
        movq      %r9,%r12
         jmp      .L2
.L1: # a = (a*a)%m; %rsi:rcx
     #  (c*p + d)^2 2cd p + d^2
        movq     %rcx,%rax
        mulq     %rcx
        divq     %rbx
        movq     %rax,%rdi
        movq     %rdx,%r11
        movq     %rcx,%rax
        mulq     %rsi
        addq     %rax,%rax
        adcq     %rdx,%rdx
        addq     %rdi,%rax
        adcq     %r12,%rdx
        divq     %rbx
        movq     %r11,%rcx
        movq     %rdx,%rsi
###
.L2:
         testb    $1, %r10b
         je       .L3
  ###pw= (a*pw)%m;a  %rsi:rcx pw %r9:%r8 p=rbx
        movq    %r8,%rax
        mulq    %rcx
        divq    %rbx
        movq    %rax,%rdi  # /
        movq    %rdx,%r11  # %
        movq    %r8,%rax
        mulq    %rsi
        divq    %rbx
        addq    %rdx,%rdi
        movq    %r9,%rax
        mulq    %rcx
        divq    %rbx
        movq    %r11,%r8
        movq    %rdx,%r9
        addq    %rdi,%r9
        cmpq    %rbx,%r9
        jb      .L3
        sub     %rbx,%r9
        cmpq    %rbx,%r9
        jb      .L3
        sub     %rbx,%r9
.L3:
        shrq    %r10
        jne     .L1
####
        movq     %r12,%rax
        cmpq      $1, %r8
        je .L4
.L5:
##################
        popq      %r12
        popq      %r11
        popq      %r10
        popq      %r9
        popq      %r8
        ret
.L4:
        testq     %r9,%r9
        jne       .L5
        movq      %r8,%rax
        jmp       .L5
```

１－３　ABC予想との関連

2012年京都大学・望月新一教授は、宇宙際Teichmuller理論に基づいてABC予想を証明したとする論文を発表した。この証明が正しいかどうかについての判断には暫く時間が必要であろうが、ABC予想と奇数の完全数について少し言及する。

まず筆者の考えでは、ABC予想から、直ちに、奇数の完全数の非存在が導けることはないであろう。

次に、大きな研究方向のひとつである最大素因子について、後藤と大野[3]は 10^8 の証明の為に、計算機で26000時間以上(約3年の計算、実際にはパソコン10台程度使い４ヶ月で計算終了)の計算時間が必要であったと報告している。最大素因子 n の証明に必要な計算時間は $O(n^2)$ であるので、一桁更新するためには約100倍の計算時間が必要となる。この長大に必要となる計算時間は「受容可能な円分数」を探すための計算時間であり、受容可能な円分数は極めて限定的にしか存在しないと予想されるので、実は無駄な計算をしているだけである。その計算時間が不要となれば大幅に計算時間を短縮できる。

Murty and Wong [4] 等が議論しているように，ABC 予想を仮定すると，同じ素因子で何回も割れる円分数は非常に限られる。このことから、受容可能な円分数は極めて限定的であることが、円分数の素因数の大きさを評価することによって得られる可能性がある。

2．計算機による計算結果

筆者らの計算機による計算アルゴリズムは、Montgomery[3] の $3 \leq p \leq 2^{32}$ での計算方法とは違う方法である。

a に関しては、$2 \leq a \leq 99$ の中で、べき乗でない数は 87 個あり、その数 a について計算を行った。本研究では、[3] のリストに載っていない新しい解を 24 個付け加えた。特に、$a = 34$ と $a = 90$ については最初の解が見つかった。

表 1．$a^{p-1} \equiv 1 \mod p^2$ の解　ただし、$2 \leq a \leq 99$、$3 \leq p < 1.3 \times 10^{12}$

a	p の値	a	p の値
2	1093 3511	54	19 1949
3	11 1006003	55	3 30109 7278001 **27207490529 902060958301**
5	20771 40487 53471161 1645333507 **6692367337 188748146801**	56	647 7079771 **115755260963**
6	66161 534851 3152573	57	5 47699 86197
7	5 491531	58	131 42250279
10	3 487 56598313	59	2777
11	71	60	29 9566295763
12	2693 123653	61	
13	863 1747591	62	3 19 127 1291
14	29 353 **7596952219**	63	23 29 36713 401771
15	29131 **119327070011**	65	17 163
17	3 46021 48947 **478225523351**	66	89351671 **588024812497**
18	5 7 37 331 33923 1284043	67	7 47 268573
19	3 7 13 43 137 63061489	68	5 7 19 113 2741
20	281 46457 9377747 122959073	69	19 223 631 2503037
21		70	13 142963
22	13 673 1595813 492366587	71	3 47 331
23	13 2481757 13703077 **15546404183**	72	
24	5 25633	73	3
26	3 5 71 486999673 **6695256707**	74	5 **1251922253819**
28	3 19 23	75	17 43 347 31247
29		76	5 37 1109 9241 661049 **20724663983**
30	7 160541 **94727075783**	77	32687
31	7 79 6451 2806861	78	43 151 181 1163 56149 4229335793
33	233 47441 **9639595369**	79	7 263 3037 1012573 60312841
34	**46145917691**	80	3 7 13 6343
35	3 1613 3571	82	3 5 46145917691
37	3 77867 **76407520781**	83	4871 13691 315746063
38	17 127	84	163 653 20101
39	8039	85	11779
40	11 17 307 66431	86	68239 **6232426549**
41	29 1025273 138200401	87	1999 48121 **604807523183**
42	23 **719867822369**	88	2535619637
43	5 103	89	3 13
44	3 229 5851	90	**6590291053**
45	1283 131759 157635607	91	3 293
46	3 829	92	727 383951 12026117 18768727 1485161969
47		93	5 509 9221 81551
48	7 257	94	11 241 32143 463033
50	7	95	2137 15061 **96185643031**
51	5 41	96	109 5437 8329 12925267 **103336004179**
52	461 1228488439	97	7 2914393 **76704103313**
53	3 47 59 97	98	3 28627 61001527
		99	5 7 13 19 83

太文字は新しく付け加わった解である

3．考察

筆者らが計算に使ったパソコンは、CPUが Core i7 3770K(Ivy Bridge)、Clock 3.5GHz, メモリを16GB搭載したものであった。Core i7ではx86-64のコードが実行可能であり、コア数４でスレッド総数は８つであり、１台のパソコンで８つの並列計算まではそれほどの処理速度の低下がなく実行可能であった。

Montgomery[2] の計算はDECstation 3100 (MIPS architecture)で行われているが、32ビットアーキテクチャの制限もあり、$3 \le p < 2^{32}$ の計算範囲で終わっている。

本論文では a の範囲は同じであるが、$3 \le p < 1.3\times10^{12}$ までの計算を行い新しい解24個を付け加えている。実際には計算機による計算は10兆までをほぼ終えているが、ここで１兆と少しまでの解のリストのみを記載しているのには以下の理由がある。

式(1)の解を p^2 フェルマーの解と呼ぶとすれば、筆者らのこの解の利用は受容可能な円分数の決定へのものとなるだろうからである。つまり、受容可能な円分数を計算アルゴリズムの改良とABC予想を使って計算時間の著しい短縮が出来たとしても、現状のCPUの計算速度とメモリの制限により、奇数の完全数の最大素因子の計算はせいぜい 10^{12} 程度までと予想されるからである。

本論文での p^2 フェルマーの解の計算は[2]とは違った方法で計算をしていると述べている。勿論多倍長計算で行うことも可能であるが、64ビットレジスター２個で済むところを多倍長計算では64ビット領域が４つ必要となり、CPUのレジスターのみを使っての計算も複雑になり利点がないと考えられる。

Montgomery自身はこのフェルマーの解でよりもモンゴメリ乗除算で有名である。これを組み込んで計算を速くしようと試みたが、Core i7 3770K(Ivy Bridge)では、筆者らの現在の計算は本質的に多倍長計算ではないと言う事もありモンゴメリ乗除算による計算速度アップは10%程度であった。これはモンゴメリ乗除算を組み込む事によるCPU内レジスターの不足(MMXレジスターは利用しないとする)、メモリー間アクセスの発生等で割の合うことではないと判断される。とは言え、Core i7 の古い型番ではモンゴメリ乗除算は有効に機能し、効率的になるようである。

フェルマーの解の計算に限定して計算速度をアップするアルゴリズムの構築は可能であると考える。ひとつの方法は確率論的なアルゴリズムの導入(一般に言われる確率論的なアルゴリズムと言うより、量子計算的なものに近い) が考えられる。他の高速化の方法としてはベキ乗計算の高速化することである。

参考文献

[1] J. Brillhart, J. Tonascia, and P. Weinberger: On the Fermat quotient. Computers in Number Theory (A. O. L. Atkin and B. 1. Birch, eds.). Academic Press, London and New York, 1971, pp. 213-222.
[2] P. L. Montgomery: New solutions of $a^{p-1} \equiv 1 \pmod{p^2}$, Math. Comp. 61 (1993), 361-363.
[3] T. Goto and Y. Ohno: Odd perfect numbers have a prime factor exceeding 10^8, Math. Comp. 77 (2008), 1859-1868.
[4] M. R. Murty and S. Wong: The ABC conjecture and prime divisors of the Lucas and Lehmer sequences, Number Theory for the Millenium, III, (Urbana, IL, 2000), A. K. Peters, Natick, MA, 2002, 43-54.

A new calculation and solution of $a^{p-1} \equiv 1 \pmod{p^2}$

Daisuke Ishii, Yoshiyuki Mori* and Ryuichi Sawae*

Graduate School of Science,
**Department of Applied Mathmatics, Faculty of Science,*
Okayama University of Science,
1-1 Ridai-cho, Kita-ku, Okayama 700-0005, Japan

(Received October 1, 2012; accepted November 1, 2012)

In Montgomery[2], it is stated that some number-theoretic questions such as Fermat's conjecture require primes p satisfying

$$a^{p-1} \equiv 1 \pmod{p^2} \qquad (1)$$

for given a not a power and its solutions for $2 \le a \le 99$ and $3 \le p < 2^{32}$ are listed.

Since we need the solutions for a proof of a prime factor in odd perfect numbers, its further calculation is needed. So, we have calculated the solutions of (1) for $2 \le a \le 99$ and $3 \le p < 1.3 \times 10^{12}$.

We have a news that Prof. Mochizuki of Kyoto University has released a 500-page proof of the ABC conjecture. If we employ the ABC conjecture, there is a possibility that the calculations related to odd perfect numbers will be extremely decreased. Then, the list of the solutions in this paper will be usefull data to complete the proof.

Keywords: number theory; odd perfect number; ABC conjecture.

References

[1] J. Brillhart, J. Tonascia, and P. Weinberger: On the Fermat quotient. Computers in Number Theory (A. O. L. Atkin and B. 1. Birch, eds.). Academic Press, London and New York, 1971, pp. 213-222.
[2] P. L. Montgomery: New solutions of $a^{p-1} \equiv 1 \pmod{p^2}$, Math. Comp. 61 (1993), 361-363.
[3] T. Goto and Y. Ohno: Odd perfect numbers have a prime factor exceeding 10^8, Math. Comp. 77 (2008), 1859-1868.
[4] M. R. Murty and S. Wong: The ABC conjecture and prime divisors of the Lucas and Lehmer sequences, Number Theory for the Millenium, III, (Urbana, IL, 2000), A. K. Peters, Natick, MA, 2002, 43-54.