千葉県立船橋高等学校
理数科課題研究 2019 年

x^2+ny^2 の形で表される素数の条件

On the prime numbers of the form x^2+ny^2

千葉県立船橋高等学校理数科 3 年
森田大輝

導入と目的

フェルマー(1607~1665)は、自然数 x,y を用いて x^2+y^2 の形で表される奇素数(2 以外の素数)と 4 を法として 1 と合同な奇素数が同じものであることを発見した。これはフェルマーの二平方和定理と呼ばれ、下のように表すことができる。

p を奇素数、x,y を自然数とする。

$$p = x^2 + y^2 \quad \Leftrightarrow \quad p \equiv 1 \pmod 4$$

さらにフェルマーは, x^2+2y^2 や x^2+3y^2 の形で表される素数についての研究も行い、「$p=x^2+2y^2 \Leftrightarrow p\equiv 1,3 \pmod 8$」「$p=x^2+3y^2 \Leftrightarrow p\equiv 1 \pmod 3 \Leftrightarrow p\equiv 1,7 \pmod{12}$」のような結果を得ている。

今回の研究では、3 より大きな自然数 n について、x^2+ny^2 の形で表される素数について、4n を法とした同様の判定方法がないか、プログラミングによって探してみた。

方法

今回は C 言語を利用した。

⑴ 約 1550 万以下の素数(100 万個)の配列を作成する。
⑵ ⑴の素数の中で x^2+ny^2 の形で表せるもののリストを作成する。
⑶ ⑵で作成したリストを 4n で割った余りのリストを作成する。
⑷ ⑶のリストの数と 4n を法として合同な素数(リスト⑴の)が全て x^2+ny^2 の形か調べる。
⑸ ⑷の条件をクリアした n のリストを作成する。

これを実行するプログラムが次である。

```
/*素数配列 p[i]を作成*/
p[1] = 2;  p[2] = 3;  l = 5;
while (i <1000000) {
        j=1;  isprime = 1;
        while (p[j] <= sqrt(l)){
                if (l-(l/p[j])*p[j] == 0) { isprime = 0; break;};
                j++;
        };
        if (isprime == 1) { i++; p[i] = l; };
        l = l+2;
};
/*素数配列 p[i]作成　おわり*/
```

```
/*各々の n について p=x²+ny² ⇔ p≡a,b,… (mod 4n) のような素数の判定法がないか調べる
    u は p を 4n で割った余り
    z はある素数 p について p=x²+ny² を満たす(x,y)の組数
    q[k],r[k]は p が p=x²+ny² の形で２通りに表されるときの p,z の値
    f[u]=1 のとき u と合同な p のうち少なくとも１つは x²+ny² の形で表せる
    g[u]=1 のとき u と合同な p のうち少なくとも１つの p が x²+ny² の形で表せない
    全ての u に対し f[u]*g[u]=0 ならば fg=0
    ① n の値を定め、各々の p について x²+ny² の形で表せるか確かめる
    ② q[k],r[k],f[u],g[u],fg,を求める
  n を変えこれらの動作を再び行う*/
for (n=1; n <= NMAX; n++) {
        fg = 0; k = 1;
        for (i=1; i<= (4*n); i++) { f[i] =0;  g[i] =0; };
        for (i = 1; i <= I; i++) {
                y = 1; z = 0; t = p[i]-n*y*y; u = p[i]-(p[i]/(4*n))*(4*n);
                while (t >0) {
                        x = sqrt(t); if (t == x*x) { z++;  f[u] = 1; };
                        y++;    t = p[i]-n*y*y;
                };
                if (z == 0) {g[u] = 1;};
                if (z >= 2) {q[k]  = p[i]; r[k] = z; k++; };
        };
        for (j=1;j <= 4*n;j++)  { if (f[j]*g[j] ==1) { fg = 1; }; };
        if (fg == 0) {
                printf( "  n n= %d : ",n);
                for (j=1;j <= 4*n;j++) { if (f[j]==1) printf(" %d",j);};
        };
};
```

結果

p をリスト(1)の素数とするとき、「p=x²+ny² ⇔ p≡a,b,… (mod 4n)」の形の判定法が得られるような n の値 65 個

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16, 18, 21, 22, 24, 25, 28, 30, 33, 37, 40, 42, 45, 48, 57, 58, 60, 70, 72, 78, 85, 88, 93, 102, 105, 112, 120, 130, 133, 165, 168, 177, 190, 210, 232, 240, 253, 273, 280, 312, 330, 345, 357, 385, 408, 462, 520, 760, 840, 1320, 1365, 1848

が見つかった。

また、n を固定したとき、p=x²+ny²となる(x,y)の組はあっても１組しかないことが確かめられた。

上記の n について「$p=x^2+ny^2 \Leftrightarrow p \equiv a, b, \cdots \pmod{4n}$」の a, b, … の値にある性質が見られた。(性質Ⅰ Ⅱ Ⅲ)

性質Ⅰ

65個の n のうち、

n = 1, 5, 8, 9, 13, 16, 21, 24, 25, 33, 37, 40, 45, 48, 57, 72, 85, 88, 93, 105, 112, 120, 133, 165, 168, 177, 232, 240, 253, 273, 280, 312, 345, 357, 385, 408, 520, 760, 840, 1320, 1365, 1848

の42個については、

$$p = x^2 + ny^2 \quad \Leftrightarrow \quad p \equiv m^2 \pmod{4n}$$

(m は n と互いに素な、ある奇数)

例 1)

n = 13 のとき

13 と互いに素な奇数 m は 4n を法として

1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37,
41, 43, 45, 47, 49, 51

これらの平方は 4n を法として、

1, 9, 17, 25, 29, 49

よって、 $p = x^2 + 13y^2 \Leftrightarrow p \equiv 1, 9, 17, 25, 29, 49 \pmod{52}$

性質Ⅱ

65個の n のうち、

n = 2, 4, 6, 10, 12, 18, 22, 28, 30, 42, 58, 60, 70, 78, 102, 130, 190, 210, 330, 462

の20個については、

$$p = x^2 + ny^2 \quad \Leftrightarrow \quad p \equiv m^2 \text{ or } m^2 + n \pmod{4n}$$

(m は n と互いに素な、ある奇数)

例 2)

n = 10 のとき

10 と互いに素な奇数 m は 4n を法として

1, 3, 7, 9, 11, 13, 17, 19, 21, 23, 27, 29, 31, 33, 37, 39

これらの平方は 4n を法として、

1, 9

また、これらに n を加えたものは

11, 19

よって、 $p = x^2 + 10y^2 \Leftrightarrow p \equiv 1, 9, 11, 19 \pmod{40}$

性質Ⅲ

65個の n のうち,

n = 3, 7, 15

の3個については、

$$p = x^2 + ny^2 \Leftrightarrow p \equiv m^2 \text{ or } m^2 + 2n \pmod{4n}$$

(m は n と互いに素な、ある奇数)

※上の結果は全て15485863までの素数についての結果である。

参考文献

ウィキペディア：二個の平方数の和（https://ja.wikipedia.org/wiki/二個の平方数の和）

ジョセフ・H・シルヴァーマン 著、鈴木治郎 訳：はじめての数論（丸善出版）

研究の経過・反省・感想等

始めはテーマがなかなか決まらず、そのころ興味があった整数や複素数などに関する本を読んだりしてテーマを探した。最終的には、8 月中旬に、Wikipedia でこの研究のきっかけであるフェルマーの二平方和定理を見つけ、この定理は n が 3 より大きいときでも 4n を法とした同様の定理があるのではないかと素朴な疑問を抱き、この研究を始めた。しかし、高校程度の数学しかまだ学んでいない私にとって推測した定理を証明することは不可能に近く、さらに手計算でこれが正しいものであるか実験することも時間的な面でも困難であった。そこで、プログラムを作成し推測した定理を大量に検証させることにした。

私は今までにプログラミングをしたことがなかったのでプログラミングの基礎から学び、簡単なものを書く練習から始め、徐々に自分の研究に使うプログラムを書いた。しかし、初めての経験でもあったため、なかなか効率の良い方法のプログラムが書けず、効率の良い方法を見つけるのに非常に苦労した。また手計算で行うときの方法とは全く異なる方法を用いたプログラムを書いたので、自分でも方針がわからなくなることもあり理解をする事がとても難しかった。さらに、プログラム中のミスを見つけることもかなり大変だった。たった一つのミスを見つけることに一日近く費やしたこともあり、考えたり書いたり作業とはまた別の苦労があった。しかしこれらのことは、プログラミングをしたことがなかった私にとってとても良い経験になった。

また、結果から法則性を見つける段階においては、結果の数字が意味するところを考え、見つけた法則に当てはまらない数字についても何か一貫性が見られないか等、自らの手を用いて考察した。このことは、実際の研究においても重要な作業だと思うので、今後の為になる研究ができたと思う。

今回の研究では、新しいことにも挑戦する事ができたし、また数学的な思考力を養う事もできたのではないかと思う。いつか自分の手でこの推測した定理を証明できるよう、これからも数学の学習に積極的に取り組んでいきたい。