JAPLA研究会資料 2019/1/19

Ｊでピタゴラス数(２つの平方数に分解)を求める
Ｊによるフェルマーの無限降下法

西川　利男

ＪによるRecreational Computingのすすめ

このところ、JAPLAに以前、発表したレポートをいろいろ見直してみた。数年前、中野嘉弘氏、山下紀幸氏といっしょにA. Beiler, Recreations in the Numbers
という本を互いに読み合って、整数論の問題をJAPLAで何回も話題にしたことがあった。この本のまえがきに、著者Beilerが学生時代にMathematical Recreations and Essaysという本をすすめられ、そこからこういう書名にした、ということが書かれている。

これにあやかって、私自身、「ＪによるRecreational Computing」という立場で、Ｊと付き合いたいと思っている。ここ数年来の私の関心テーマ、群論とか、グラフィックスなどはそのつもりである。

１．ピタゴラス数とは

さて、ピタゴラス数とは任意の整数をどうやって２乗の和で表すかという問題である。先のBeilerの本でもピタゴラス数という章があるが、当時はあまり良くわからなかった。

その後、次の書に出会った。

[1] ジョセフH. シルバーマン、鈴木治朗訳「はじめての数論」
ピアソンエデュケーション (2001).

ここでは、ピタゴラス数を計算するための無限降下法が３章にわたって解説されている。しかし、これを手計算でやるのはとうてい無理だ。

そしてごく最近、金　重明氏の次の本が出た。

[2] 金　重明「１３歳の娘に語るガウスの黄金定理」岩波書店(2013).

語り口は通常の数学書とは異なるが、必ずしもわかりやすいとはいえない。

そこで、あらためて、シルバーマンを読み直し、Ｊでプログラミングしつつ理解することにした。これによって、フェルマーの無限降下法とその基礎となる「ガウスの法による計算（余りをベースにした整数計算）」の偉大さにはじめて触れたことになった。ここまでずいぶん時間もかかったし、まさに「ぼけ予防の頭の体操＝Recreation」といったところである。

２．4n + 1 型素数と4n + 3型の素数　[1], p.162

シルバーマンによると、整数論を自分のものにするには、はじめに問題を具体的な数で少数の場合について調べその傾向を見る。一般的な文字の式誘導だけではだめだ、と言っている。また、今回は、素数に限って検討する。

最初に、準備として100までの素数について、２つの平方数の和でどう表されるか、めのこで当たってみる。

2 = 1^2 + 1^2, 3 NO, 5 = 1^2 + 2^2, 7 NO, 11 NO, 13 = 2^2 + 3^2

17 = 1^2 + 4^2, 19 NO, 23 NO, 29 = 2^2 + 5^2, 31 NO, 37 = 1^2 + 6^2

41 = 4^2 + 5^2, 43 NO, 47 NO, 53 = 5^2 + 6^2, 67 NO, 71 NO,

73 = 3^2 + 8^2, 79 NO, 83 NO, 89 = 5^2 + 8^2, 97 = 4^2 + 9^2

これらの結果から、次のような予想がたてられるだろう。

平方数かどうかによって、素数は２種類に分けられる、ということになる。

2, 5, 13, 17, 29, 37, 41, 53, 61, 73, 89, 97 ⇒ 4n+1 型の素数

3, 7, 11, 19, 23, 31, 43, 47, 59, 67, 71, 79, 83 ⇒ 4n+3 型の素数

これを、整数論では次のように言い表す。

4を法として1に合同な素数は２つの平方数の和であらわせる。

4を法として3に合同な素数は２つの平方数の和ではあらわせない。

式で書くと、

$p \equiv 1 \pmod 4$ のときにかぎり、平方数の和（ピタゴラス数）つまり

$p = x^2 + y^2$

と表せる

一般的証明はシルバーマンの本にある。

３．フェルマーの無限降下法とは［1], p.164-170

それでは、4n+1 型の素数 p 与えられたとき、具体的にどんな平方数の和であらわされるかという問題は p が大きくなるとそう簡単にはいかない。これを解く方法はフェルマーにより示され、その後オイラーらによって定式化された無限降下法である。

無限降下法の考え方は、つぎのようなものである。

最終目標は4n+1 型の素数 p を２つの平方数の和で表したいのだが、

(1) まず、易しい p の整数倍の Mp について平方和を求める。

$A^2 + B^2 = Mp$

(2) 次に、法による計算を行うことはもちろんだが、平方数の積と和を変換する次の恒等式

$$(u\text{^}2+v\text{^}2)*(A\text{^}2+B\text{^}2)=(u*A+v*B)\text{^}2+(v*A-u*B)\text{^}2$$

を利用して、

以下に述べるアルゴリズムに従い、Mより小さい数mを見つける。

$$a\text{^}2+b\text{^}2=mp,\quad m\le M-1$$

この係数mを降下させる操作をくりかえして、最後は1になるまで繰り返す。

われわれJプログラマーにとっては、数学の式誘導よりも、プログラムの実行で示したほうが、わかりよいのではなかろうか。

２つのサブルーチンとなる動詞定義ferm0と ferm1を作り、それを使うメインルーチン動詞fermを定義した。

さらに、具体的な例、素数p = 881について、示していこう。

４．無限降下法の実際

４．１　ferm0による処理

まず、動詞ferm0では、B=1とした平方和について、成り立つようなA,B,Mの組を探し出す。これには、素数pにn=1, 2, 3, と係数を順次掛けて、絨毯爆撃のようにして、平方和になるまで繰り返し、見つけるのである。

Jのプログラムは、素朴につぎのようにすればよい。

```
i =. 1
b =. 1
while. i < p
  do.
     Right =. i * p
     B2 =. *: b
     A2 =. Right - B2
     if. (<. %: A2) = (%: A2) do. goto_OK. end.
     i =. i + 1
end.
label_OK.
)
```

n、np, 右辺を開平した数と繰り返し計算の途中経過を示すと次のようになる。

```
   1 ferm0 881
1 881 29.6648
2 1762 41.9643
3 2643 51.4004
```

```
4 3524 59.3549
5 4405 66.3626
6 5286 72.698
7 6167 78.5239
8 7048 83.9464
9 7929 89.0393
10 8810 93.8563
(途中省略)
160 140960 375.445
161 141841 376.617
162 142722 377.784
163 143603 378.949
164 144484 380.109
165 145365 381.266
166 146246 382.42
167 147127 383.57
168 148008 384.717
169 148889 385.86
170 149770 387
hit !!
M = 170
387 1 170 881
```

すなわち ferm0 の処理では、

```
387^2 + 1^2 = 170 * 881
```

なる *A, B, M* の組(387, 1, 170)が見つかった。

４．２　ferm1 による処理

ここから、整数論でもっとも有用な法(mod)による合同計算が真価を発揮する

$$47 \equiv 387 \pmod{170}$$
$$1 \equiv 1 \pmod{170}$$

これから

$$47^2 + 1^2 \equiv 387^2 + 1$$
$$\equiv 149770$$
$$\equiv 0 \pmod{170}$$

となる。また

$$47^2 + 1^2 = 170 * 13$$
$$387^2 + 1^2 = 170 * 881$$

これを辺々同志掛けて

$$(47^2 + 1^2) * (387^2 + 1^2) = 170^2 * 13 * 881$$

となる。

ここで、先に示した平方数の積と和を変換する恒等式を左辺に適用して、平方数の和の形にする。

$$(47 * 387 + 1 * 1)^2 + (1 * 387 - 47 * 1)^2 = 170^2 * 13 * 881$$
$$18190^2 + 340^2 = 170^2 * 13 * 881$$

辺々を170^2で割る

$$\left(\frac{18190}{170}\right)^2 + \left(\frac{340}{170}\right)^2 = 13 * 881$$
$$170^2 + 2^2 = 13 * 881$$

このようにしてより小さいpの係数mが得られた

つまりmとして先の170より小さい13が得られた。

すなわち、動詞 ferm1 により、

a, b, m の組(170, 2, 13)が見つかったのである。

つぎは、この(170, 2, 13)を引数にして、動詞 ferm1 を作用させるのである。

この操作を繰り返して行って、mが 1 に等しくなれば、終了となる。

無限降下法とは、ここから来たアルゴリズムである。

Jのプログラム ferm1 の主要部分は次のようになる。

```
ferm1 =: 3 : 0
'A B M p' =. y
u =. M | A
v =. M | B
m =. ( ((*: u) + (*: v)) * ((*: A) + (*: B)) ) % ((*: M) * (p))
a =. ((u*A) + (v*B)) % M
b =. ((v*A) - (u*B)) % M
a, b, m, p
)
```

整数の mod を用いる合同式計算に、Jのプリミティブ動詞 | が、如何に強力に使われているか、わかるであろう。

Jが整数論にとって、またとないすばらしいプログラミング環境であることに私はほほえんでいる。

４．３　ferm による処理

無限降下法全体の処理動詞 ferm は、最初だけ ferm0 を行い、次からは ferm1 を繰り返して、p の係数が 1 になるまで続ける。実行結果はつぎのようになる。

```
   ferm 881
387 1 170 881
107 2 13 881
25 16 1 881
881 = 25^2 + 16^2
```

途中経過の表示なども示した、Jプログラム全体のコーディングは後にあげた

５．ピタゴラス数分解のいろいろな例

５．１　シルバーマンの本[1] p.170

練習問題25.3

```
   ferm 12049
2639 1 578 12049
1493 4 185 12049
105 32 1 12049
12049 = 105^2 + 32^2
```

練習問題25.4(a)

```
   ferm 1973
259 1 34 1973
160 7 13 1973
53 84 5 1973
99 _8 5 1973
76 46 4 1973
23 38 1 1973
1973 = 23^2 + 38^2
```

練習問題25.4(b)

```
   ferm 96493
31791 1 10474 96493
1120 3 13 96493
173 258 1 96493
96493 = 173^2 + 258^2
```

５．２　金　重明の本[2] p.83

```
   ferm 3709
1609 1 678 3709
491 2 65 3709
272 14 20 3709
173 182 17 3709
159 90 9 3709
106 _60 4 3709
53 30 1 3709
3709 = 53^2 + 30^2
```

Jのプログラム

```
NB. revised 2018/10/29
NB. ferm0 881
ferm0 =: 3 : 0
0 ferm0 y.
:
i =. 1
p =. y.
b =. 1
while. i < p
  do.
     Right =. i * p
NB.      B2 =. 1
     B2 =. *: b
     A2 =. Right - B2
if. 1 = x. do.  wr i, Right, (%: A2) end.          NB.  途中経過表示
     if. (<. %: A2) = (%: A2) do. goto_OK. end.
goto_AUTO.
NB. read YN                                        NB.  キー入力による途
中経過表示
     YN =. rd 1                                    NB.  そのまま空CRなら
次へ続行
     if. 0 = #YN do. goto_CR. end.                 NB.  'y', 'yes'なら次
へ続行
     if. 'n' = YN                                  NB.  'n' なら実行取り
やめ
        do. return. end.
label_CR.
label_AUTO.
     i =. i + 1
end.
label_OK.
if. 1 = x. do.  wr 'hit !!' end.
     M =. i
if. 1 = x. do.  wr 'M = ', ": M end.
     A =. %: A2
```

```
     B =. %: B2
A, B, M, p
)

NB. ferm1 387 1 170 881
NB. 1 ferm1 387 1 170 881 => 途中説明
NB. ferm1 107 2 13 881
NB. 1 ferm1 107 2 13 881 => 途中説明
ferm1 =: 3 : 0
0 ferm1 y.
:
NB. p =. y.
'A B M p' =. y.

if. 0 = x. do. goto_skip1. end.
wr 'A^2  +  B^2 =  M  * p'
wr (": A) ,'^2 + ',(": B) , '^2 = ', (": M) , ' * ' , (": p)
wr '----------------------------'
label_skip1.

     u =. M | A
     v =. M | B
     m =. ( ((*: u) + (*: v)) * ((*: A) + (*: B)) ) % ((*: M) * (p))

if. 0 = x. do. goto_skip2. end.
     wr 'u=', (": u), ', v=', (": v)
     wr 'm=', ": m
wr 'u^2 + v^2 =  M * m'
NB. wr (":u), '^2', (":v), '^2 = ', (":M) , ' * ', (":m)
wr (":u), '^2 + ', (":v), '^2 = ', (":M) , ' * ', (":m)
wr '----------------------------'

wr '(u*A + v*B)^2    +  (v*A - u*B)^2  = (M^2) * m * p'
label_skip2.

   PP  =. '(', (": u), '*', (":A), ' + ', (":v), '*', (":B), ')^2  + '
```

```
   QQ  =. '(', (": v), '*', (":A), ' - ', (":u), '*', (":B), ')^2  = '
   RR =.  (":M),'^2 * ', (":m), ' * ', (":p)

if. 0 = x. do. goto_skip3. end.
wr PP, QQ, RR
wr (": (u*A) + (v*B)), '^2 + ', (": (v*A) - (u*B)), '^2 = ', (":M),
'^2 * ', (": m*p)
wr '------------------------------'
wr (": ((u*A) + (v*B)) % (M)) ,'^2 + ', (": ((v*A) - (u*B)) % (M)),
'^2 = ', (": m), '*', (":p)
label_skip3.

a =. ((u*A) + (v*B)) % M
b =. ((v*A) - (u*B)) % M

a, b, m, p
)

NB. Usage:
NB.     ferm 881            シルバーマン  p.166
NB. 107 2 13 881
NB. 25 16 1 881
NB. 881 = 25^2 + 16^2

NB.     ferm 3709           金  重明「13歳の娘に語るガウスの黄金定理」p.85
NB. 491 2 65 3709
NB. 272 14 20 3709
NB. 173 182 17 3709
NB. 159 90 9 3709
NB. 106 _60 4 3709
NB. 53 30 1 3709
NB. 3709 = 53^2 + 30^2

ferm =: 3 : 0
10 ferm y.
:
```

```
jend =. x.
p =. y.
'a b m p' =. ferm0 p
j =. 0
while. j < jend
  do.
    wr 'a b m p' =. ferm1 a, b, m, p
    if. 1 = m do. goto_OK. end.
    j =. j + 1
  end.
label_OK.
NB. a, b, m, p
(": p), ' = ', (":a), '^2 + ', (":b), '^2'
)
```

シルバーマン　p.166

```
   ferm 881
107 2 13 881
25 16 1 881
881 = 25^2 + 16^2
```

シルバーマン　p.170, 練習問題 25.3

```
   ferm 12049
1493 4 185 12049
105 32 1 12049
12049 = 105^2 + 32^2
```

シルバーマン　p.170, 練習問題 25.4(a)

```
   ferm 1973
160 7 13 1973
53 84 5 1973
99 _8 5 1973
76 46 4 1973
23 38 1 1973
1973 = 23^2 + 38^2
```

シルバーマン　p.170, 練習問題 25.4(b)

```
   ferm 96493
1120 3 13 96493
173 258 1 96493
96493 = 173^2 + 258^2
```

金　重明 p.83

```
   ferm 3709
491 2 65 3709
272 14 20 3709
173 182 17 3709
159 90 9 3709
106 _60 4 3709
53 30 1 3709
3709 = 53^2 + 30^2
```

```
NB. Usage:
NB.     ferm 881            シルバーマン  p.166
NB. 107 2 13 881
NB. 25 16 1 881
NB. 881 = 25^2 + 16^2

NB.     ferm 3709           金  重明「13歳の娘に語るガウスの黄金定理」p.85
NB. 491 2 65 3709
NB. 272 14 20 3709
NB. 173 182 17 3709
NB. 159 90 9 3709
NB. 106 _60 4 3709
NB. 53 30 1 3709
NB. 3709 = 53^2 + 30^2

ferm =: 3 : 0
10 ferm y.
:
jend =. x.
p =. y.
'a b m p' =. ferm0 p
j =. 0
while. j < jend
  do.
    wr 'a b m p' =. ferm1 a, b, m, p
    if. 1 = m do. goto_OK. end.
    j =. j + 1
  end.
label_OK.
NB. a, b, m, p
(":p), ' = ', (":a), '^2 + ', (":b), '^2'
)
```

```
NB. 素数かどうか
qprime =: (1&=)@(#@q:)
NB.  10までの素数:  (qprime # ]) >: i.10 => 2 3 5 7
NB.  100までの素数:  P100 =: (qprime # ]) >: i.100 => 2 3 5 7 11 13 , , 79 83 89
97

NB.  100までの素数(P100) をmod 4 で表すと
NB.   4 | (qprime # ]) >: i.100
NB. 2 3 1 3 3 1 1 3 3 1 3 1 1 3 3 1 3 1 3 3 1 3 3 1 1

NB. 2つの平方数の和に表せる素数、 シルバーマン p.162  ================
NB.  4n + 1 素数: (1 = 4 | P100) # P100 => 5 13 17 29 37 41 53 61 73 89 97
NB.  4n + 3 素数: (3 = 4 | P100) # P100 => 3 7 11 19 23 31 43 47 59 67 71 79 83

NB. シルバーマン  p.166 降下法の手続き ================================
NB. Usage: search 881
NB.        1 search 881 => 途中経過表示
search =: 3 : 0
0 search y.
:
i =. 1
'p b'=. y.
while. i < p
  do.
     Right =. i * p
NB.      B2 =. 1
     B2 =. *: b
     A2 =. Right - B2
if. 1 = x. do.  wr i, Right, (%: A2) end.          NB.  途中経過表示
     if. (<. %: A2) = (%: A2) do. goto_OK. end.
goto_AUTO.
NB. read YN                                        NB.  キー入力による途中経過表示
     YN =. rd 1                                    NB.  そのまま空CRなら次へ続行
     if. 0 = #YN do. goto_CR. end.                 NB.  'y', 'yes'なら次へ続行
     if. 'n' = YN                                  NB.  'n' なら実行取りやめ
        do. return. end.
```

```
label_CR.
label_AUTO.
     i =. i + 1
end.
label_OK.
     wr 'hit !!'
     wr 'M = ', ": M =. i
     A =. %: A2
     B =. %: B2
     wr 'A=', (": A), ', B=', (": B)
wr 'A^2  +  B^2 =  M  * p'
wr (": A) ,'^2 + ',(": B) , '^2 = ', (": M) , ' * ' , (": p)
wr '----------------------------'
     u =. M | A
     v =. M | B
     wr 'u=', (": u), ', v=', (": v)
     wr 'm=', ": m =. ( ((*: u) + (*: v)) * ((*: A) + (*: B)) ) % ((*: M) * (p))
wr 'u^2 + v^2 =  M * m'
NB. wr (":u), '^2', (":v), '^2 = ', (":M) , ' * ', (":m)
wr (":u), '^2 + ', (":v), '^2 = ', (":M) , ' * ', (":m)
wr '----------------------------'
wr '(u*A + v*B)^2    +  (v*A - u*B)^2  = (M^2) * m * p'
   PP  =. '(', (": u), '*', (":A), ' + ', (":v), '*', (":B), ')^2  + '
   QQ  =. '(', (": v), '*', (":A), ' - ', (":u), '*', (":B), ')^2  = '
   RR =.  (":M),'^2 * ', (":m), ' * ', (":p)
wr PP, QQ, RR
wr (": (u*A) + (v*B)), '^2 + ', (": (v*A) - (u*B)), '^2 = ', (":M), '^2 * ', (":
m*p)
'___'
(": ((u*A) + (v*B)) % (M)) ,'^2 + ', (": ((v*A) - (u*B)) % (M)), '^2 = ', (":
m*p)
)

NB. revised 2018/10/29
NB. ferm0 881
ferm0 =: 3 : 0
```

```
0 ferm0 y.
:
i =. 1
p =. y.
b =. 1
while. i < p
  do.
     Right =. i * p
NB.      B2 =. 1
     B2 =. *: b
     A2 =. Right - B2
if. 1 = x. do.  wr i, Right, (%: A2) end.          NB.  途中経過表示
     if. (<. %: A2) = (%: A2) do. goto_OK. end.
goto_AUTO.
NB. read YN                                        NB.  キー入力による途中経過表示
     YN =. rd 1                                    NB.  そのまま空CRなら次へ続行
     if. 0 = #YN do. goto_CR. end.                 NB.  'y', 'yes'なら次へ続行
     if. 'n' = YN                                  NB.  'n' なら実行取りやめ
        do. return. end.
label_CR.
label_AUTO.
     i =. i + 1
end.
label_OK.
if. 1 = x. do.  wr 'hit !!' end.
     M =. i
if. 1 = x. do.  wr 'M = ', ": M end.
     A =. %: A2
     B =. %: B2
A, B, M, p
)

NB. ferm1 387 1 170 881
NB. 1 ferm1 387 1 170 881 => 途中説明
NB. ferm1 107 2 13 881
NB. 1 ferm1 107 2 13 881 => 途中説明
```

```
ferm1 =: 3 : 0
0 ferm1 y.
:
NB. p =. y.
'A B M p' =. y.

if. 0 = x. do. goto_skip1. end.
wr 'A^2  +  B^2 =  M  * p'
wr (": A) ,'^2 + ',(": B) , '^2 = ', (": M) , ' * ' , (": p)
wr '----------------------------'
label_skip1.

     u =. M | A
     v =. M | B
     m =. ( ((*: u) + (*: v)) * ((*: A) + (*: B)) ) % ((*: M) * (p))

if. 0 = x. do. goto_skip2. end.
     wr 'u=', (": u), ', v=', (": v)
     wr 'm=', ": m
wr 'u^2 + v^2 =  M * m'
NB. wr (":u), '^2', (":v), '^2 = ', (":M) , ' * ', (":m)
wr (":u), '^2 + ', (":v), '^2 = ', (":M) , ' * ', (":m)
wr '----------------------------'

wr '(u*A + v*B)^2    +  (v*A - u*B)^2  = (M^2) * m * p'
label_skip2.

   PP  =. '(', (": u), '*', (":A), ' + ', (":v), '*', (":B), ')^2  + '
   QQ  =. '(', (": v), '*', (":A), ' - ', (":u), '*', (":B), ')^2  = '
   RR =.  (":M),'^2 * ', (":m), ' * ', (":p)

if. 0 = x. do. goto_skip3. end.
wr PP, QQ, RR
wr (": (u*A) + (v*B)), '^2 + ', (": (v*A) - (u*B)), '^2 = ', (":M), '^2 * ', (":
m*p)
wr '----------------------------'
```

```
wr (": ((u*A) + (v*B)) % (M)) ,'^2 + ', (": ((v*A) - (u*B)) % (M)), '^2 = ', (":
m), '*', (":p)
label_skip3.
a =. ((u*A) + (v*B)) % M
b =. ((v*A) - (u*B)) % M
a, b, m, p
)

NB. Usage:
NB.     ferm 881            シルバーマン  p.166
NB. 107 2 13 881
NB. 25 16 1 881
NB. 881 = 25^2 + 16^2

NB.     ferm 3709           金  重明「13歳の娘に語るガウスの黄金定理」p.85
NB. 491 2 65 3709
NB. 272 14 20 3709
NB. 173 182 17 3709
NB. 159 90 9 3709
NB. 106 _60 4 3709
NB. 53 30 1 3709
NB. 3709 = 53^2 + 30^2

ferm =: 3 : 0
10 ferm y.
:
jend =. x.
p =. y.
'a b m p' =. ferm0 p
j =. 0
while. j < jend
  do.
    wr 'a b m p' =. ferm1 a, b, m, p
    if. 1 = m do. goto_OK. end.
    j =. j + 1
  end.
```

```
label_OK.
NB. a, b, m, p
(':' p), ' = ', (':a), '^2 + ', (':b), '^2'
)
```