Python で楽しむ初等整数論

Hayao Suzuki

PyCon mini Hiroshima 2019

October 12, 2019

Contents

1. 自己紹介
2. 初等整数論とは
3. ピタゴラス数
4. 2 つの平方数の和で表せる素数
5. まとめ

自己紹介

お前誰よ

名前	Hayao Suzuki（鈴木　駿）
Twitter	@CardinalXaro
ブログ	https://xaro.hatenablog.jp/
専門	数学 (組合せ論・グラフ理論)
学位	修士（工学)、電気通信大学
仕事	株式会社アイリッジで Python プログラマをしている

自己紹介

技術書の査読

- 『Effective Python』（オライリージャパン）
- 『NumPy によるデータ分析入門』（オライリージャパン）など
- https://xaro.hatenablog.jp/ に一覧あります。

いろんな発表

- 「SymPy による数式処理」（PyCon JP 2018）など
- https://xaro.hatenablog.jp/ に一覧あります。

初等整数論とは

整数論とは

数の性質に関する数学の一分野

初等整数論とは

代数的な手法や解析的な手法を使わない数論

初等だから簡単ですね？ よかった！

初等とは予備知識がいらないという意味であり決して簡単ではありません。

今日の発表

なぜ Python を使うのか

- 比較的素直にプログラミングできる
- 標準ライブラリも外部ライブラリも豊富にある
- Guido van Rossum 氏はアムステルダム大学で数学と計算機科学を専攻していた

今回使うもの

- Python 3.7.x
- SymPy（記号計算ライブラリ、数論ライブラリも豊富）
- Silverman『はじめての数論 原著第 3 版』（丸善出版）（ネタ元）

今日の発表

Python で楽しむ初等整数論

- 定理の結果を Python でプログラミングしてみよう
- 定理の証明から Python でプログラミングしてみよう

でも... 難しいんでしょう？

割り算ができれば大丈夫！（個人差があります）

定理の結果を Python でプログラミングしてみよう

ピタゴラスの定理

定理 (Pythagoras)

直角三角形の斜辺の長さを c とし、それ以外の辺の長さを a, b とする。そのとき、

$$a^2 + b^2 = c^2$$

が成り立つ。

ピタゴラス数

定義（ピタゴラス数）

$a^2 + b^2 = c^2$ が成り立つ自然数の組 (a, b, c) をピタゴラス数と呼ぶ。

ピタゴラス数を計算しよう：根性編

根性で計算だ！

```
from itertools import product

for a, b, c in product(range(1, 20), repeat=3):
    if a ** 2 + b ** 2 == c ** 2:
        print(a, b, c)
```

ピタゴラス数を計算しよう：根性編

根性で計算したぞ！

3 4 5
4 3 5
5 12 13
6 8 10
8 6 10
8 15 17
9 12 15
12 5 13
12 9 15
15 8 17

既約ピタゴラス数

命題（ピタゴラス数からピタゴラス数を作る）

自然数の組 (a, b, c) がピタゴラス数ならば、任意の自然数 n に対して (na, nb, nc) もまたピタゴラス数である。

証明.

$$(na)^2 + (nb)^2 = n^2(a^2 + b^2) = (nc)^2.$$

□

定義（既約ピタゴラス数）

互いに素であるピタゴラス数を既約ピタゴラス数と呼ぶ。

既約ピタゴラス数を計算しよう

ちょっと工夫した

```
from itertools import combinations
from math import gcd

for a, b, c in combinations(range(1, 50), 3):
    if a ** 2 + b ** 2 == c ** 2 and gcd(a, b) == 1:
        print(a, b, c)
```

ちょっと工夫した

- a, b, c はそれぞれ違う数である
- a, b は互いに素である

既約ピタゴラス数を計算しよう

既約ピタゴラス数の計算

3 4 5
5 12 13
7 24 25
8 15 17
9 40 41
12 35 37
20 21 29

既約ピタゴラス数をもっと上手く計算しよう

補題

既約ピタゴラス数 (a, b, c) の a, b の一方は偶数でもう一方は奇数である。また、c は常に奇数である。

証明.

自然数の組 (a, b, c) を既約ピタゴラス数とする。
まず、a, b が共に偶数ならば、ピタゴラス数の定義から c も偶数となり、(a, b, c) は共通因数 2 を持つことになり仮定に反する。次に、a, b が共に奇数ならば、ピタゴラス数の定義から c は偶数となる。このとき、$a = 2x + 1, b = 2y + 1, c = 2z$ とすると

$$a^2 + b^2 = c^2 \Rightarrow 2(x^2 + x + y^2 + y) + 1 = 2z^2$$

となり矛盾。
よって、a, b の内、一方は偶数でもう一方は奇数である。また、ピタゴラス数の定義から c は奇数となる。 □

既約ピタゴラス数をもっと上手く計算しよう

補題

既約ピタゴラス数 (a, b, c) において、a を奇数、b を偶数とする。このとき、$(c-b)(c+b)$ は互いに素である。

証明.

自然数の組 (a, b, c) を既約ピタゴラス数とし、a を奇数、b を偶数とする。仮定から、$a^2 = c^2 - b^2 = (c-b)(c+b)$ とできる。
$c-b, c+b$ の共通因数を d とすると、d は
$(c+b)+(c-b) = 2c, (c+b)-(c-b) = 2b$ も割り切る。仮定から b, c は既約なので d は 1 または 2 である。しかし、
$(c-b)(c+b) = a^2$ であり、a は奇数なので d は 1 に他ならない。

□

既約ピタゴラス数をもっと上手く計算しよう

補題

既約ピタゴラス数 (a, b, c) において、a を奇数、b を偶数とする。このとき、$c - b, c + b$ はそれぞれ平方数となる。

証明.

自然数の組 (a, b, c) を既約ピタゴラス数とし、a を奇数、b を偶数とする。仮定から、$a^2 = c^2 - b^2 = (c - b)(c + b)$ とできる。$c - b, c + b$ は互いに素であり、かつ $(c - b)(c + b) = a^2$ なので、$(c - b)(c + b)$ は平方数となる。よって、$c - b, c + b$ がそれぞれ平方数となる。 □

既約ピタゴラス数をもっと上手く計算しよう

命題

互いに素な奇数 $s, t (s > t)$ 対して、
$a = st, b = \dfrac{s^2 - t^2}{2}, c = \dfrac{s^2 + t^2}{2}$ は既約ピタゴラス数となる。

証明.

自然数の組 (a, b, c) を既約ピタゴラス数とし、a を奇数、b を偶数とする。仮定から、$a^2 = c^2 - b^2 = (c - b)(c + b)$ とできる。$c - b, c + b$ はそれぞれ平方数なので、$c + b = s^2, c - b = t^2$ とできる。よって、

$$c = \frac{s^2 + t^2}{2}, b = \frac{s^2 - t^2}{2}$$

となる。また、

$$a = \sqrt{(c - b)(c + b)} = st$$

となる。 □

既約ピタゴラス数をもっと上手く計算しよう

上手く工夫した

```
from itertools import combinations
from math import gcd

for t, s in filter(
    lambda x: gcd(*x) == 1,
    combinations(range(1, 10, 2), r=2),
):
    print(
        s * t,
        int((s ** 2 - t ** 2) / 2),
        int((s ** 2 + t ** 2) / 2),
    )
```

既約ピタゴラス数をもっと上手く計算しよう

既約ピタゴラス数をもっと上手く計算した

3 4 5
5 12 13
7 24 25
9 40 41
15 8 17
21 20 29
35 12 37
45 28 53
63 16 65

Python で楽しむ初等整数論

定理の結果を Python でプログラミングしてみよう

- 結果をそのまま実装して具体例を観察する
- 意外な法則が見つかる（かも）

定理の証明から Python でプログラミングしてみよう

2 つの平方数の和で表せる素数

定理 (Fermat の 2 平方和定理)

奇素数 p が 2 つの平方数の和で表せることの必要十分条件は $p \equiv 1 \pmod 4$ である。

証明 ($\Rightarrow$).

奇素数 p が 2 つの平方数の和で表せるならば、$p = x^2 + y^2$ となる。p は奇数なので、x を偶数、y を奇数としてよい。このとき、
$x = 2m, y = 2n + 1$ とすると

$$p = x^2 + y^2 = (2m)^2 + (2n + 1)^2 = 4(m^2 + n^2 + n) + 1$$

より、p は 4 を法として 1 と合同である。

□

証明のスケッチ

証明 ($\Leftarrow$) のスケッチ

- p を 4 を法として 1 と合同な素数とする。
- いい感じに $A^2 + B^2 = Mp$ なる (A, B, M) を探す。
- (A, B, M) からいい感じに $a^2 + b^2 = Mr, r \leq M - 1$ なる (a, b, r) を探す。
- M が 1 になるまで繰り返す。

これを無限降下法と呼ぶ。

証明 ($\Leftarrow$) のスケッチ

補題

$$(u^2+v^2)(A^2+B^2) = (uA+vB)^2 + (vA-uB)^2$$

証明.

計算するだけ。

$$\begin{aligned}
&(uA+vB)^2 + (vA-uB)^2 \\
&= (u^2A^2 + 2uAvB + v^2B^2) + (v^2A^2 - 2vAuB + u^2B^2) \\
&= u^2A^2 + v^2B^2 + v^2A^2 + u^2B^2 \\
&= (u^2+v^2)(A^2+B^2).
\end{aligned}$$

□

証明 ($\Leftarrow$) のスケッチ

補題

合同方程式 $x^2 \equiv -1 \pmod{p}$ は解を持つ。

証明（天下り）.

平方剰余の相互法則から導ける。 □

いい感じに $A^2 + B^2 = Mp$ なる (A, B, M) を探す

合同方程式 $x^2 \equiv -1 \pmod{p}$ の解を A とし、$B = 1$ とすると、

$A^2 + B^2$ は p の倍数である。また、$M = \dfrac{A^2 + B^2}{p}$ となる。

いい感じに $A^2 + B^2 = Mp$ なる (A, B, M) を探す

合同方程式 $x^2 \equiv -1 \pmod{p}$ の解を算出する（天下り）

```
from random import randint
from sympy.ntheory import isprime, legendre_symbol


def solve_quadratic(p):
    if not (isprime(p) and p % 4 == 1):
        raise ValueError
    while True:
        a = randint(1, p - 1)
        b = pow(a, (p - 1) // 4, p)
        if legendre_symbol(a, p) == -1 and 1 <= b < p:
            return b
```

いい感じに $A^2 + B^2 = Mp$ なる (A, B, M) を探す

いい感じに (A, B, M) を探す

```
def sums_of_two_squares(p):
    """4 を法として 1 と合同な素数を 2 つの平方数の和で表す"""

    if not (isprime(p) and p % 4 == 1):
        raise ValueError

    A, B = solve_quadratic(p), 1
    M = divmod(A ** 2 + B ** 2, p)[0]
```

いい感じに $a^2 + b^2 = mp$ を探す

いい感じに $a^2 + b^2 = Mr$ を探す

以下の条件を満たすように a, b を選ぶ。

$$a \equiv A \pmod{M}$$
$$b \equiv B \pmod{M}$$
$$-\frac{1}{2}M \leq a, b \leq \frac{1}{2}M$$

r と M の関係の秘密

$$r = \frac{a^2 + b^2}{M} \leq \frac{\left(\frac{M}{2}\right)^2 + \left(\frac{M}{2}\right)^2}{M} = \frac{M}{2} < M.$$

いい感じに $a^2 + b^2 = Mr$ を探す

いい感じに $a^2 + b^2 = Mr$ を探す

```
from sympy.ntheory.modular import solve_congruence



def sums_of_two_squares(p):
    """4 を法として 1 と合同な素数を 2 つの平方数の和で表す"""

    if not (isprime(p) and p % 4 == 1):
        raise ValueError

    A, B = solve_quadratic(p), 1
    M = divmod(A ** 2 + B ** 2, p)[0]

    a = solve_congruence((A, M), symmetric=True)[0]
    b = solve_congruence((B, M), symmetric=True)[0]
    r = divmod(a ** 2 + b ** 2, M)[0]
```

総仕上げ

2 つの方程式

$$A^2 + B^2 = Mp$$
$$a^2 + b^2 = Mr$$

2 つの方程式の関係

$$(a^2 + b^2)(A^2 + B^2) = M^2rp$$
$$\Rightarrow (aA + bB)^2 + (bA - aB)^2 = M^2rp$$
$$\Rightarrow \left(\frac{aA + bB}{M}\right)^2 + \left(\frac{bA - aB}{M}\right)^2 = rp$$

$r \leq \frac{M}{2}$ なので、効率よく計算できる。

2 つの平方数の和で表せる素数

2 つの平方数の和で表せる素数

```
def sums_of_two_squares(p):
    A, B = solve_quadratic(p), 1
    M = divmod(A ** 2 + B ** 2, p)[0]
    while True:
        a = solve_congruence((A, M), symmetric=True)[0]
        b = solve_congruence((B, M), symmetric=True)[0]
        r = divmod(u ** 2 + v ** 2, M)[0]

        s = abs((a * A + b * B) // M)
        t = abs((b * A - a * B) // M)
        if r == 1:
            print(f"${s}^2 + {t}^2={p}$")
            return
        else:
            A, B, M = s, t, r
```

2 つの平方数の和で表せる素数

2 つの平方数の和で表せる素数

```
from sympy import primerange

primes_1_mod_4 = (
    p for p in primerange(1, 100) if p % 4 == 1
)

for p in primes_1_mod_4:
    sums_of_two_squares(p)
```

2 つの平方数の和で表せる素数

2 つの平方数の和で表せる素数

$2^2 + 1^2 = 5$
$3^2 + 2^2 = 13$
$4^2 + 1^2 = 17$
$5^2 + 2^2 = 29$
$6^2 + 1^2 = 37$
$5^2 + 4^2 = 41$
$7^2 + 2^2 = 53$
$6^2 + 5^2 = 61$
$8^2 + 3^2 = 73$
$8^2 + 5^2 = 89$
$9^2 + 4^2 = 97$

Python で楽しむ初等整数論

定理の証明から Python でプログラミングしてみよう

- 構成的な証明はプログラミングできる（こともある）
- プログラミングできれば証明を理解できる（かもしれない）

まとめ

なぜ Python を使うのか

- 比較的素直にプログラミングできる
- 標準ライブラリも外部ライブラリも豊富にある
- 困ったら SymPy を当たれば解決することが多い

定理の結果を Python でプログラミングしてみよう

- 結果をそのまま実装して具体例を観察する
- 意外な法則が見つかる（かも）

定理の証明から Python でプログラミングしてみよう

- 構成的な証明はプログラミングできる（こともある）
- プログラミングできれば証明を理解できる（かもしれない）