## The Sum of Two Integer Cubes Restricted

Kenny Jonsson

Department of Mathematics and Computer Science
Mathematics
Degree Project 15ECTS, Bachelor Level
Supervisor: Martin Lind
Examiner: Sorina Barza
Date: September 13, 2022


#### Abstract

We study the size of sets containing sums of two integer cubes such that their representation is unique and also fit between two consecutive integer cubes. We will try to write algorithms that efficiently calculate the size of these sets and also implement these algorithms in Python ${ }^{\text {TM }}$.

Although we will fail to find a non-iterative algorithm, we will find different ways of approximating the size of these sets. We will also find that techniques used in our failed algorithms can be used to calculate the number of integer lattice points inside a circle.


Keywords - Diophantine equations, lattice points, number theory

## Acknowledgements

I would like to thank my supervisor, Martin Lind, for his valuable guidance and support.

My thanks goes out to my family for their love and support.
I am very grateful to Anders Hedin for rekindling my love for mathematics.
Last but not least, I would like to thank Ulrika Knape Lindberg, Lars Lundberg, Malin Hedlund and The Pirate Bay for inspiring me to pursue higher education.

## Contents

1 Some classical results in additive number theory ..... 2
1.1 Introduction ..... 2
1.2 Auxiliary results ..... 4
1.3 Proof of Fermat's two squares theorem ..... 5
2 Cube sums ..... 6
2.1 Counting solutions to Diophantine equations ..... 6
2.2 Initial algorithm ..... 7
2.3 Improved algorithm ..... 9
2.4 Color coded ..... 10
2.5 The $x, y$ plots ..... 12
2.6 Fast algorithm ..... 13
2.7 Heuristic asymptotic ..... 16
2.8 Application ..... 17
I Calculations ..... 19
I. 1 First pattern break ..... 19
I. 2 Limits of $x, y$ ..... 20
I. 3 Overlapping points conjecture ..... 22
I. $4 x_{\min }, x_{\max }$ ..... 23
II Code ..... 25
II. 1 For cube sums ..... 25
II. 2 For points in circle ..... 27
II. 3 Performance ..... 28

## Chapter 1

## Some classical results in additive number theory

### 1.1 Introduction

In number theory, polynomials play a fundamental role. This thesis is devoted to the study of integral solutions to polynomial equations, so-called Diophantine equations. One of the most famous result of this type is Fermat's last theorem, which states that the polynomial equation

$$
x^{n}+y^{n}=z^{n}
$$

has no positive integer solutions if $n \geqslant 3$.
A similar but different question can be formulated generally in the following way. If $P=P\left(x_{1}, x_{2}, \ldots, x_{d}\right)$ is a polynomial defined on $\mathbb{R}^{d}$, which integers belong to the image of $P(\mathbb{Z})$ ? For instance, let $P(x, y)=x^{2}+y^{2}$, for which positive integers $n$ can one find $x, y \in \mathbb{Z}$ such that $n=P(x, y)$ ? This problem was solved independently by Girard and Fermat in the 17th century. It is in fact sufficient to solve the problem for $n=$ prime number, the result is called Fermat's two-squares theorem.

Theorem 1 (Fermat's two squares theorem). Let $p$ be a prime such that $p>2$ and $x, y \in \mathbb{Z}$, then

$$
\begin{equation*}
p=x^{2}+y^{2} \Longleftrightarrow p \equiv 1 \quad(\bmod 4) . \tag{1.1.1}
\end{equation*}
$$

The standard modern proof of the previous result uses the fact that

$$
x^{2} \equiv-1 \quad(\bmod p)
$$

has a solution if and only if $p \equiv 1(\bmod 4)$, together with Dirichlet's approximation theorem. (see Hardy-Wright [1]).
D. R. Heath-Brown [2] suggested a completely different approach to derive Theorem 1 based on group actions on sets. Later in this chapter we present a short proof of Theorem 1, largely based on the approach used by S. Dolan [3].

Theorem 1 suggests several generalizations. First, we mention Lagrange's foursquares theorem which states that any positive integer $n$ can be represented

$$
n=x^{2}+y^{2}+z^{2}+w^{2} \quad(x, y, z, w \in \mathbb{Z}) .
$$

A deeper result is Legendre's three-squares theorem, which characterizes the positive integers that are sums of three squares. Finally, questions of the above type are all contained in the famous Waring's problem. For $k \geqslant 2$ and $m \in \mathbb{N}$, denote

$$
\mathcal{S}_{k}(m)=\left\{n \in \mathbb{N}: n=\sum_{j=1}^{m} x_{j}^{k} \text { for } x_{1}, x_{2}, \ldots x_{m} \in \mathbb{Z}\right\} .
$$

Define

$$
g(k)=\min \left\{m: \mathcal{S}_{k}(m) \in \mathbb{N}\right\} .
$$

Lagrange's four-squares theorem can equivalently be formulated as $g(2)=4$. Waring's problem is to calculate $g(k)$ for $k \geqslant 3$. Hilbert showed that $g(k)<\infty$ for all $k \geqslant 2$. Much work has been devoted to the computation of $g(k)$ but the problem remains wide open.

In the second part of the thesis, we consider the question of counting solutions to nonlinear Diophantine equations. Such problems are central to analytical number theory. Our counting problem is a rather subtle one: we want to compute the number of unique sums of cubes between two consecutive integer cubes. Letting $\Sigma(a)$ denote the set of these cube sums, we use

$$
\# \Sigma(a)=\#\left\{x^{3}+y^{3} \mid(a-1)^{3}<x^{3}+y^{3}<a^{3}, a, x, y \in \mathbb{N}, 0<x \leqslant y\right\},
$$

to describe its size. We devise, implement and analyze several algorithms to compute $\# \Sigma(a)$ using both algebraic and geometrical arguments. Furthermore, we consider the asymptotic behavior of $\# \Sigma(a)$ for large $a$ and observe experimentally that

$$
\lim _{a \rightarrow \infty} \frac{\# \Sigma(a)}{a} \approx \frac{\sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)}=0.8833 \ldots,
$$

where $\Gamma$ denotes the Gamma-function. We give a heuristic derivation of the above limit.

### 1.2 Auxiliary results

As previously stated, Theorem 1 can be solved by different means. The approach used by Heath-Brown is to show that the equation $p=x^{2}+4 y z$ has an odd number of solutions (the odd one being when $y=z$ ). Since our proof uses a similar approach, we will find the following results useful.

Lemma 2. All perfect squares ${ }^{1}$ are congruent to 0 or 1 modulo 4.
Proof. Let $n$ be an even integer. We write $n=2 k$ for $k \in \mathbb{Z}$. Then

$$
\begin{equation*}
n^{2}=4 k^{2} \equiv 0 \quad(\bmod 4) . \tag{1.2.1}
\end{equation*}
$$

If instead $n$ is odd, we write $n=2 k+1$, and get

$$
\begin{equation*}
n^{2}=4 k^{2}+4 k+1 \equiv 1 \quad(\bmod 4) . \tag{1.2.2}
\end{equation*}
$$

By combining the results of (1.2.1) and (1.2.2), we see that all perfect squares are congruent to 0 or 1 modulo 4 .

Lemma 3. For $a, b, c \in \mathbb{N}$, any $4 k+1$ prime can be written on the form

$$
\begin{equation*}
p=(a+b+c)^{2}-4 a c . \tag{1.2.3}
\end{equation*}
$$

Proof. Let $a=c=k$ and $b=1$ in (1.2.2):

$$
\begin{align*}
(2 k+1)^{2} & =4 k^{2}+4 k+1 \Longleftrightarrow  \tag{1.2.4}\\
(a+b+c)^{2} & =4 a c+p \Longleftrightarrow  \tag{1.2.5}\\
p & =(a+b+c)^{2}-4 a c . \tag{1.2.6}
\end{align*}
$$

Lemma 4. Let

$$
\begin{align*}
& (x+\alpha)^{2}=\beta  \tag{1.2.7}\\
& (x-\alpha)^{2}=\beta \tag{1.2.8}
\end{align*}
$$

for $x, \alpha, \beta \in \mathbb{N}$ such that $\alpha^{2} \neq \beta$. For fixed $\alpha, \beta$ we see that if $x=a$ is a solution to (1.2.7), then $x=-a$ solves (1.2.8) and vice versa. This leads to exactly one of the following possibilities:
a) (1.2.7) and (1.2.8) both have no solutions.
b) Every solution to (1.2.7) gives us a solution to (1.2.8).

Both of these outcomes can be summed up as to say that (1.2.7) and (1.2.8) have an even number of solutions between them.

[^0]
### 1.3 Proof of Fermat's two squares theorem

Theorem 1 states that for a prime $p>2$ and $x, y \in \mathbb{Z}$,

$$
p=x^{2}+y^{2} \Longleftrightarrow p \equiv 1 \quad(\bmod 4) .
$$

Proof. We know that the sum of two squares is congruent to 0 , 1 or 2 modulo 4 , from Lemma 2. As all odd numbers can be written as either $4 k+1$ or $4 k+3$ (for $k \in \mathbb{Z}$ ), a prime cannot be written as the sum of two squares if it is on the form $4 k+3$, since we cannot get a residue of 3 for any sum of two squares.

$$
\begin{equation*}
\therefore p=x^{2}+y^{2} \Longrightarrow p \equiv 1 \quad(\bmod 4) . \tag{1.3.1}
\end{equation*}
$$

Given a prime $p$ on the form $4 k+1$, the solutions of

$$
\begin{equation*}
(a+b+c)^{2}=p+4 a c \tag{1.3.2}
\end{equation*}
$$

come in pairs $(a, b, c),(c, b, a)$. The only possibility for these pairs to be equal is if $c=a$. Letting $c=a$ in (1.3.2), gives us

$$
\begin{align*}
(2 a+b)^{2} & =p+4 a^{2} \Longleftrightarrow \\
p & =4 a b+b^{2} \\
& =b(4 a+b) . \tag{1.3.3}
\end{align*}
$$

Since $p$ is prime it only has two factors ( 1 and $p$ ). So the only solution to (1.3.3) is $(a, b, c)=(k, 1, k)$. Rearranging (1.3.2) gives us

$$
\begin{align*}
(a+b+c)^{2} & =p+4 a c
\end{align*} \Longleftrightarrow
$$

Observe that the left-hand side of (1.3.4) is a square and must therefore be greater than or equal to 0 , which implies $p \geqslant 4 b c$. Since $p$ is finite, the number of solutions to (1.3.4) must also be finite ${ }^{2}$. Lemma 4 shows that the number of solution to (1.3.4) is even when $b \neq c$. Since (1.3.2) has an odd number of solutions, there is at least one solution where $b=c$. Letting $b=c$ in (1.3.4), gives us

$$
\begin{align*}
a^{2} & =p-4 c^{2} \Longleftrightarrow \\
p & =a^{2}+(2 c)^{2}  \tag{1.3.5}\\
\therefore p \equiv 1 \quad(\bmod 4) & \Longrightarrow p=x^{2}+y^{2} \tag{1.3.6}
\end{align*}
$$

Combining (1.3.1) and (1.3.5), gives us

$$
p=x^{2}+y^{2} \Longleftrightarrow p \equiv 1 \quad(\bmod 4) .
$$

[^1]
## Chapter 2

## Cube sums

As many equations, proofs and code that belongs to this chapter are quite lengthy and not that interesting, unless you are already fascinated by this subject, they are being presented in appendices I and II. See I for calculations and II for source code (written in Python" ${ }^{\text {TM }}$ ).

### 2.1 Counting solutions to Diophantine equations

While working on problems involving different ways of summing powers of natural numbers ${ }^{1}$ the author, out of curiosity, plotted some of the results. A few tangents later the question that spawned this paper arose - how many unique numbers on the form $x^{3}+y^{3}, x, y \in \mathbb{N}$ fit between two consecutive integer cubes? Letting $\Sigma(a)$ denote the set of these $x, y$ sums, we use

$$
\begin{equation*}
\# \Sigma(a)=\#\left\{x^{3}+y^{3} \mid(a-1)^{3}<x^{3}+y^{3}<a^{3}, a, x, y \in \mathbb{N}, 0<x \leqslant y\right\} \tag{2.1.1}
\end{equation*}
$$

to describe the size of $\Sigma(a)$ and

$$
\begin{equation*}
\mathcal{K}=\frac{\# \Sigma(a)}{a} \tag{2.1.2}
\end{equation*}
$$

to denote the approximate relation between $a$ and $\# \Sigma(a)$. The values of $\Sigma(a)$ were calculated manually for $1 \leqslant a \leqslant 10$, mainly to determine if finding $\mathcal{K}$ would prove to be a trivial task. These first ten values can be viewed in Table 2.1.7 but let us first find the elements and size of $\Sigma(a)$ for some small $a$-values, to get a feel for what we are looking for.

[^2]\[

$$
\begin{align*}
\# \Sigma(1) & =\#\left\{x^{3}+y^{3} \mid 0^{3}<x^{3}+y^{3}<1^{3}\right\} \\
& =\#\{ \}=0,  \tag{2.1.3}\\
\# \Sigma(2) & =\#\left\{x^{3}+y^{3} \mid 1^{3}<x^{3}+y^{3}<2^{3}\right\} \\
& =\#\{1+1\}=\#\{2\}=1,  \tag{2.1.4}\\
\# \Sigma(3) & =\#\left\{x^{3}+y^{3} \mid 8<x^{3}+y^{3}<27\right\} \\
& =\#\{1+8,8+8\}=2,  \tag{2.1.5}\\
\# \Sigma(4) & =\#\left\{x^{3}+y^{3} \mid 27<x^{3}+y^{3}<64\right\} \\
& =\#\{1+27,8+27,27+27\}=3 . \tag{2.1.6}
\end{align*}
$$
\]

Thus far, $\mathcal{K}$ seems rather ordinary. For $a \leqslant 4, \Sigma(a)$ contains elements of the form $y^{3}=(a-1)^{3}$ and $x^{3} \leqslant(a-1)^{3}$. Looking at Table 2.1.7, this pattern is broken at $a=5$ and Equation (I.1.5) explains why. Even though this pattern ${ }^{2}$ is broken, $\# \Sigma(a)$ does not seem that interesting until $a=10$. This was the authors first indication that these type of cube sums might be worth exploring. The second one being that the integer sequence generated by calculating $\# \Sigma(a)$ for $a \in[1,10]$ could not be found in The On-Line Encyclopedia of Integer Sequences ${ }^{\circledR}\left(\text { OEIS }{ }^{\circledR}\right)^{3}$, at the time of writing this paper.

Table 2.1.7: The first ten values of $\# \Sigma(a)$.

| a | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\# \Sigma(a)$ | 0 | 1 | 2 | 3 | 3 | 5 | 6 | 6 | 8 | 7 |

### 2.2 Initial algorithm

Wanting to visualize the relation between $a$ and $\# \Sigma(a)$ the author dipped their toes into programming and learned some of the basics for generating sets/lists and plotting the results. While there are dynamic mathematical software out there, the ones known to the author seemed ill-fitted for this task. The choice of programming language fell on Python ${ }^{\text {TM }}$, based on the number of tutorials, guides and videos that are available to anyone with an internet connection. Algorithm 2.2.1 contains the pseudocode for one of the first functioning algorithms.

[^3]```
Algorithm 2.2.1 Pseudocode of II.1.1
    SumList = []
    For \(x\) in \([1, a-1]\)
        For \(y\) in \([1, a-1]\)
            Sum \(=x^{3}+y^{3}\)
            If \((a-1)^{3}<\) Sum \(<a^{3}\) and
                If Sum is not in SumList \({ }^{4}\)
                Add Sum to SumList
            Else do nothing
            Else do nothing
    Return the length of Sumlist
```

Any reader that knows a bit of programming will see that using Algorithm 2.2.1 is not a very efficient way of calculating $\# \Sigma(a)$. This algorithm will perform $(a-1)^{2}$ loops in order to find the value of $\# \Sigma(a)$. A value that, based on the information at hand, seems to be less than $a$. Despite its lack of efficiency, Algorithm 2.2.1 is at least fast enough to quickly generate the first one hundred values of $\# \Sigma(a)$, the results of which are visualized in Figure 2.2.2.


Figure 2.2.2: Plot of $\# \Sigma(a)$ and $\mathcal{K} a$.
The results seen in Figure 2.2.2, even though appearing chaotic, hints at the possibility that the relation between $\# \Sigma(a)$ and $a$ is asymptotically linear. An

[^4]initial hand wave approximation was that
\[

$$
\begin{equation*}
\frac{\# \Sigma(a)}{a}=\mathcal{K} \approx 0.88 \tag{2.2.3}
\end{equation*}
$$

\]

### 2.3 Improved algorithm

Wanting to calculate $\# \Sigma(a)$ for much larger $a$-values, given the poor performance of Algorithm 2.2.1, we might be tempted to start over entirely but the general idea of it is valid. We want to count the number of times that $x^{3}+y^{3}$ lays between $(a-1)^{3}$ and $a^{3}$, for different $x-, y$-values. The simplest way of doing that is to loop over each possible $x$ - and $y$-value, but Algorithm 2.2.1 loops over much more than just the possible values. We can reduce the number of loops significantly by limiting the ranges of $x$ and $y$. Equations (I.2.10) and (I.2.19) show that

$$
\begin{align*}
\left|\frac{a-1}{\sqrt[3]{2}}\right| & \leqslant y \leqslant a-1 \text { and }  \tag{2.3.1}\\
\max \left(1,\left\lceil\sqrt[3]{(a-1)^{3}-y^{3}} \mid\right)\right. & \leqslant x \leqslant\left\lfloor\sqrt[3]{a^{3}-y^{3}}\right\rfloor \tag{2.3.2}
\end{align*}
$$

are the suitable ranges. With these, we can improve Algorithm 2.2.1.

```
Algorithm 2.3.3 Pseudocode of improved algorithm II.1.4
    SumList = []
    \(y m i n=\lceil(a-1) / \sqrt[3]{2}\rceil\)
    For \(y\) in [ymin, a-1]
        \(\operatorname{xmin}=\max \left(1,\left\lceil\sqrt[3]{(a-1)^{3}-y^{3}}\right\rceil\right)\)
        \(\operatorname{xmax}=\left\lfloor\sqrt[3]{a^{3}-y^{3}}\right\rfloor\)
        For \(x\) in [xmin, \(x m a x]\)
            Sum \(=x^{3}+y^{3}\)
            If \((a-1)^{3}<\) Sum \(<a^{3}\) and
                If Sum is not in SumList
                    Add Sum to SumList
            Else do nothing
            Else do nothing
    return the length of Sumlist
```

Algorithm 2.3.3 is significantly faster ${ }^{5}$ at generating $\# \Sigma(a)$ than Algorithm 2.2.1 and can now be used to calculate $\# \Sigma(a)$ up to $a=10000$ reasonably fast.

[^5]

Figure 2.3.4: Plot of $\# \Sigma(a)$ and $\mathcal{K} a$, for $a \in[1,10000]$.
In Figure 2.3 .4 we se that $\# \Sigma(a)$ still seems to be asymptotically linear up to $a=10000$. Still eyeballing our relation $\mathcal{K}$, it now seems that $\mathcal{K} \approx 0.883$ might be a slightly better fit but we have no explanation for why this could be. Even when zooming in and looking closer at $\# \Sigma(a)$ a distinct pattern is not obvious.

### 2.4 Color coded

After many failed attempts, at finding the pattern of $\# \Sigma(a)$, the strange idea of printing $\left\{1^{3}, 2^{3}, \ldots, a^{3}\right\}$ as columns and highlighting the $x^{3}, y^{3}$ pairs that satisfies our conditions (2.1.1), popped up. Instead of explaining the code, let us just look at the results in Figure 2.4.1.


Figure 2.4.1: $\# \Sigma(80)$ highlighted.
This idea turned out to be a segue to the next idea - plotting the actual $x, y$ pairs in the Euclidean plane and look for patterns there rather than looking for patterns in $\# \Sigma(a)$.

### 2.5 The $x, y$ plots

If we rewrite (2.1.1) as functions, we get

$$
\begin{align*}
& y<\sqrt[3]{a^{3}-x^{3}}  \tag{2.5.1}\\
& y>\sqrt[3]{(a-1)^{3}-x^{3}}  \tag{2.5.2}\\
& x \leqslant y \text { and }  \tag{2.5.3}\\
& x>0 \tag{2.5.4}
\end{align*}
$$

These are all functions that can be represented in the Euclidean plane, provided that we choose a specific $a$-value.

Letting $a=17, f=f(a, x)=\sqrt[3]{a^{3}-x^{3}}, g=g(a, x)=\sqrt[3]{(a-1)^{3}-x^{3}}$ and $h=h(x)=x$ we can plot $f, g$ and $h$ along with the $x, y$ pairs in $\# \Sigma(17)$.


Figure 2.5.5: \# $\Sigma(17)$.
In Figure 2.5.5, the black points corresponds to $\# \Sigma(17)$. The red point $(2,16)$ has also been added. This is a point that is not included in $\# \Sigma(17)$ using Algorithm 2.3.3, since $2^{3}+16^{3}=9^{3}+15^{3}=4104$, and the $x, y$ pair 9,15 gets added first. Points like this one will however play a crippling roll in the upcoming algorithm. Before moving on to this algorithm let us define the set $D(a)$ which will contain these duplicates. In the case of $\# \Sigma(17), D(17)=\left\{2^{3}+16^{3}\right\}$, so $\# D(17)=1$.

### 2.6 Fast algorithm

Looking at Figure 2.5 .5 it seems that $\# \Sigma(17)$ can be calculated by setting a counter to $x_{\max }=13$, followed by increasing the counter by one each time a $x$-value corresponds to more than one $y$-value and finally reducing the counter by one for each duplicate value (in this case the red point). There are multiple ways of tackling this problem and one of them is Algorithm 2.6.1.

```
Algorithm 2.6.1 Pseudocode of II.1.5
    ymin = \lceil(a-1)/\sqrt{3}{2}\rceil
    Counter = xmax
    For y in [ymin + 1, a - 1]
        Calculate the largest x-value for given y
            If multiple y-values correspond to this x-value
                Increment the counter by 1
            Else do nothing
    return counter
```

Line five and six of Algorithm 2.6.1 might need some explaining. In Figure 2.5 .5 we see that multiple $y$-values can correspond to a single $x$-value. The proof of Conjecture 6 (found in Appendix I) shows why multiple stacked points only occur at the $x$-extreme for any given $y$-value, which explains why we only need to calculate this once for every $y$-value, except for $y_{\text {min }}{ }^{6}$. There is however still one piece missing in this algorithm. We need to reduce the counter for every duplicate sum $(\# \mathcal{D}(a))$. All attempts at writing algorithms to solve this last piece of our puzzle have performed worse at finding $\# \mathcal{D}(a)$ than Algorithm 2.2.1 does at finding $\# \Sigma(a)$, so for now this will have to remain an approximation. Letting $\# S(a)=$ $\# \Sigma(a)+\# \mathcal{D}(a)^{7}$, we can plot $\# S(a)$ along with $\# \Sigma(a)$.

[^6]

Figure 2.6.2: $\# \Sigma(a)$ and $S(a)$, for $a \in[1,100]$.
Looking at Figure 2.6.2, it appears that $\# S(a)$ is not that far off. Letting

$$
\varepsilon(a)=\frac{\# S(a)}{\# \Sigma(a)}, \text { if } \frac{\# S(a)}{\# \Sigma(a)} \neq 1,
$$

we can plot this function for a few $a$-values.


Figure 2.6.3: $\varepsilon(a)$, for $a \in[1,10000]$.

Note that Figure 2.6.3 contains less than 10000 values. This stems from the fact that $\# S(a)=\# \Sigma(a)$ for some values. The figure indicates that the percentual error of $\# S(a)$ is negligible and any competent number theorist can prove why this is, but let us move on.

Using mathematical notation, Algorithm 2.6.1 can be written as

$$
\begin{align*}
& \# S(a)=\left\lfloor\frac{a}{\sqrt[3]{2}}\right\rfloor+\sum_{y=y_{\min }+1}^{a-1}\left\lfloor y-g\left(a, x_{\max }(y)\right)\right\rfloor, \text { where }  \tag{2.6.4}\\
& g(a, x)=\sqrt[3]{(a-1)^{3}-x^{3}} \tag{2.6.5}
\end{align*}
$$

and $x_{\max }(y)$ is the largest $x$-value for a given $y$. The first term of (2.6.4) is the expression for $x_{\max }$ (which is the global maximum value of $x$ ), from Equation (I.4.3). The second term simply calculates the distance between a given point and the function $g(a, x)$ and returns 1 if multiple $y$-values is found for a given $x$-value, and 0 otherwise.

This is, sadly, where the story about finding a closed expression of $\# \Sigma(a)$ ends, for now. We can still try to get a expression for the approximation of $\mathcal{K}$, at least. While looking over some early notes that involved integration, the scribblings

$$
\begin{equation*}
\frac{\# \Sigma(a)}{a} \stackrel{?}{\approx} \frac{\int_{0}^{a} \sqrt[3]{a^{3}-x^{3}} \mathrm{~d} x}{a^{2}} \tag{2.6.6}
\end{equation*}
$$

where found. At first glance this might seem a bit odd, but letting $N(r)$ denote the number of integer lattice points inside a circle with radius $r$, it turns out Gauss showed[4] that

$$
\begin{equation*}
N(r)=r^{2} \pi+E(r) \tag{2.6.7}
\end{equation*}
$$

where

$$
\begin{equation*}
|E(r)| \leqslant 2 \sqrt{2} \pi r . \tag{2.6.8}
\end{equation*}
$$

Even though the shape of our area is not circular, it is close enough that we can apply the same logic ${ }^{8}$. There is, however, still something strange about (2.6.6). $\int_{0}^{a} \sqrt[3]{a^{3}-x^{3}} \mathrm{~d} x$ is not the area in which our points reside, so why would this make sense? This question will be addressed in the upcoming section.

[^7]
### 2.7 Heuristic asymptotic

Let $f(x)=\sqrt[3]{a^{3}-x^{3}}$ and $g(x)=\sqrt[3]{(a-1)^{3}-x^{3}}$. Also let $d, s$ and $F$ denote the areas of the highlighted sections in Figures 2.7.1, 2.7.2 and 2.7.3 respectively.


Figure 2.7.1


Figure 2.7.2


Figure 2.7.3

We see that our points belong to section $d$ of Figure 2.7.1. Given that the area approximates the number of integer lattice points, we get $\# \Sigma(a) \approx d$. As we have defined $F$ as $\int_{0}^{a} \sqrt[3]{a^{3}-x^{3}} \mathrm{~d} x$, (2.6.6) can be written as

$$
\begin{equation*}
\frac{d}{a} \stackrel{?}{\approx} \frac{F}{a^{2}} \tag{2.7.4}
\end{equation*}
$$

Let us look at $F$ from the right-hand side of (2.7.4).

$$
\begin{align*}
F & =\int_{0}^{a} f(x) \mathrm{d} x \\
& =\int_{0}^{a} \sqrt[3]{a^{3}-x^{3}} \mathrm{~d} x \\
& =\frac{a^{2} \sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)} \Longrightarrow  \tag{2.7.5}\\
\frac{F}{a^{2}} & =\frac{\sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)} \tag{2.7.6}
\end{align*}
$$

We can rewrite the left-hand side of (2.7.4) to avoid unnecessary calculations.

$$
\begin{align*}
\frac{d}{a} & =\frac{2 d}{2 a}=\frac{s}{2 a} .  \tag{bysymmetry}\\
s & =\int_{0}^{a} f(x) \mathrm{d} x-\int_{0}^{a-1} g(x) \mathrm{d} x \\
& =\frac{a^{2} \sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)}-\frac{(a-1)^{2} \sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)} \Longrightarrow \\
\frac{s}{2 a} & =\left(1-\frac{1}{2 a}\right) \frac{\sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)} . \tag{2.7.8}
\end{align*}
$$

The right-hand side of (2.7.8) does not equal the right-hand side of (2.7.6), but

$$
\begin{align*}
\lim _{a \rightarrow \infty}\left(1-\frac{1}{2 a}\right) \frac{\sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)} & =\frac{\sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)} \Longrightarrow  \tag{2.7.9}\\
\lim _{a \rightarrow \infty} \frac{F}{a^{2}}-\frac{s}{2 a} & =0 \text { and } \\
\frac{s}{2 a} & =\frac{d}{a}, \text { so } \\
\frac{d}{a} & \approx \frac{F}{a^{2}}, \text { for sufficiently large } a .
\end{align*}
$$

In fact $1-\frac{1}{2 a}=0.95$, for $a=10$, so the approximation is somewhat accurate even for small $a$-values.

Summing up our approximation of $\mathcal{K}$, we get

$$
\begin{aligned}
\mathcal{K} & =\frac{\# \Sigma(a)}{a} \\
& \approx \frac{d}{a} \\
& \approx \frac{F}{a^{2}} \\
& =\frac{\sqrt{\pi} \Gamma(4 / 3)}{2^{2 / 3} \Gamma(5 / 6)} \\
& =0.8833 \ldots,
\end{aligned}
$$

which is very close to our initial eye ball approximation.

### 2.8 Application

Since this paper stemmed from curiosity, possible applications where not considered. There might be some use of the failings of Algorithm 2.6.1. Since it counts all points in a given area, the type of reasoning behind it can be used to write an algorithm that counts the number of integer lattice points inside a circle while looping less than $\frac{r}{3}$ times, where $r$ is the radius. As I was not able to find out if this is efficient or not, compared to existing algorithms, we will not go into details about its inner workings here. Implementation of such an algorithm, along with a brief explanation, is found in Appendix II.

## Bibliography

[1] G. H. Hardy and E. M. Wright, An Introduction to the Theory of Numbers. Oxford Uni. Press, 1979.
[2] R. Heath-Brown, "Fermat's two squares theorem," Invariant, 1984.
[3] S. Dolan, "105.38 a very simple proof of the two-squares theorem," The Mathematical Gazette, vol. 105, no. 564, 2021.
[4] G. H. Hardy, Ramanujan: Twelve Lectures on Subjects Suggested by His Life and Work. Cambridge Uni. Press, 1940.

## Appendix I

## Calculations

## I. 1 First pattern break

Let $a, x, y \in \mathbb{N}$. For which values $a$ is the inequality

$$
\begin{align*}
& x^{3}+y^{3}<a^{3} \text { true, when }  \tag{I.1.1}\\
& x^{3}=y^{3}=(a-1)^{3} ? \tag{I.1.2}
\end{align*}
$$

Using (I.1.2) in (I.1.1) and solving for equality, gives us

$$
\begin{align*}
2(a-1)^{3}=a^{3} & \Longleftrightarrow \\
2 a^{3}-6 a^{2}+6 a-2=a^{3} & \Longleftrightarrow \\
a^{3}-6 a^{2}+6 a-2=0 & \Longleftrightarrow \\
a=2+\sqrt[3]{2}+2^{2 / 3}, \text { or } a & =2-2^{-2 / 3}-2^{-1 / 3} \pm i\left(\frac{\sqrt{3} \sqrt[3]{2}(1-\sqrt[3]{2})}{2}\right) . \tag{I.1.3}
\end{align*}
$$

The only real solution is

$$
\begin{equation*}
a=2+\sqrt[3]{2}+2^{2 / 3} \approx 4.85 \tag{I.1.4}
\end{equation*}
$$

Recalling that we only are looking for positive integer solutions, we conclude that (I.1.1) holds true when

$$
\begin{equation*}
a \in[1,4] . \tag{I.1.5}
\end{equation*}
$$

## I. 2 Limits of $x, y$

This section will be devoted to finding different limits of $x$ and $y$ that satisfies

$$
\left\{\begin{array}{l}
(a-1)^{3}<x^{3}+y^{3}  \tag{I.2.1}\\
x^{3}+y^{3}<a^{3} \\
0<x \leqslant y \\
a, x, y \in \mathbb{N}
\end{array}\right.
$$

Wanting to find the smallest $y$-value ( $y_{\text {min }}$ ), we can use the equality from (I.2.1) to get

$$
\begin{align*}
& 2 y_{\text {min }}^{3}>(a-1)^{3} \Longleftrightarrow  \tag{I.2.2}\\
& y_{\text {min }}^{3}>\frac{(a-1)^{3}}{2} \Longleftrightarrow  \tag{I.2.3}\\
& y_{\text {min }}>\frac{a-1}{\sqrt[3]{2}} . \tag{I.2.4}
\end{align*}
$$

Since there is no $a \in \mathbb{N}$ such that $\frac{a-1}{\sqrt[3]{2}} \in \mathbb{N}$, we know that

$$
\begin{align*}
\left\lceil\frac{a-1}{\sqrt[3]{2}}\right\rceil & >\frac{a-1}{\sqrt[3]{2}}  \tag{I.2.5}\\
\therefore y_{\min } & =\left\lceil\frac{a-1}{\sqrt[3]{2}}\right\rceil . \tag{I.2.6}
\end{align*}
$$

Wanting to find the largest $y$-value ( $y_{\max }$ ), the first solution that comes to minds is $y=a-1$. Solving the second inequality of (I.2.1) for $x$, using $y=a-1$, we get

$$
\begin{align*}
& x^{3}<a^{3}-(a-1)^{3} \Longleftrightarrow  \tag{I.2.7}\\
& x^{3}<3 a^{2}-3 a+1 \tag{I.2.8}
\end{align*}
$$

We see that $x=1$ is a solution, as long as $a>1$.

$$
\begin{equation*}
\therefore y_{\max }=a-1, \text { for } a>1 . \tag{I.2.9}
\end{equation*}
$$

Combining (I.2.6) and (I.2.9), we get

$$
\begin{equation*}
\left\lceil\frac{a-1}{\sqrt[3]{2}}\right\rceil \leqslant y \leqslant a-1, \text { for } a>1 \tag{I.2.10}
\end{equation*}
$$

When it comes to the limits on $x$, we have to define two types of extremes. We are going to calculate $x$ as a function of $y$, and since $y$ is found in a range, $x$ will assume a minimum and maximum value for each $y$-value. Let us denote these
by $x_{\text {min }}(y)$ and $x_{\max }(y)$, respectively. Since one out of these min and max values will also be the global min and max for $x$, we will need notation for these too. Let $x_{\text {min }}, x_{\text {max }}$ denote the global minimum and maximum. We will start by finding $x_{\min }(y)$ and $x_{\max }(y)$.

Solving the first inequality of (I.2.1) for $x$, gives us

$$
\begin{align*}
& x_{\min }(y)^{3}>(a-1)^{3}-y^{3} \text { and }  \tag{I.2.11}\\
& x_{\max }(y)^{3}<a^{3}-y^{3} . \tag{I.2.12}
\end{align*}
$$

Starting with the minimum value:

$$
\begin{gather*}
x_{\min }(y)^{3}>(a-1)^{3}-y^{3} \Longleftrightarrow  \tag{I.2.13}\\
x_{\min }(y)>\sqrt[3]{(a-1)^{3}-y^{3}} . \tag{I.2.14}
\end{gather*}
$$

Since there is no $a, y \in \mathbb{N}$ such that $\sqrt[3]{(a-1)^{3}-y^{3}} \in \mathbb{N}$, we know that

$$
\begin{align*}
\left|\sqrt[3]{(a-1)^{3}-y^{3}}\right| & >\sqrt[3]{(a-1)^{3}-y^{3}}, \text { for } y<a-1  \tag{I.2.15}\\
\therefore x_{\min }(y) & =\left\lceil\sqrt[3]{(a-1)^{3}-y^{3}} \mid, \text { for } y<a-1\right. \text { and }  \tag{I.2.16}\\
x_{\min }(y) & =1, \text { for } y=a-1, a>1 \tag{I.2.17}
\end{align*}
$$

When $y=y_{\text {min }}$ we might run into a symmetry problem, which we will tackle later on. For now, let us use the same reasoning for $x_{\max }(y)$ as for $x_{\min }(y)$ and make the weaker statement:

$$
\begin{equation*}
x_{\max }(y) \leqslant\left\lfloor\sqrt[3]{a^{3}-y^{3}}\right\rfloor . \tag{I.2.18}
\end{equation*}
$$

Combining (I.2.16), (I.2.17), (I.2.18) and letting $x(y)=\left[x_{\min }(y), x_{\max }(y)\right]$, we get

$$
\begin{equation*}
\max \left(1,\left\lceil\sqrt[3]{(a-1)^{3}-y^{3}}\right\rceil\right) \leqslant x(y) \leqslant\left\lfloor\sqrt[3]{a^{3}-y^{3}}\right\rfloor \tag{I.2.19}
\end{equation*}
$$

Before moving on to $x_{\text {min }}$ and $x_{\text {max }}$, let us make a statement. For every $y$-value, there might be multiple $x$-values that satisfies (I.2.1).

## I. 3 Overlapping points conjecture

Lemma 5. $\alpha-\beta<1 \Longrightarrow\lfloor\alpha\rfloor-\lceil\beta\rceil \leqslant 0$, for $\alpha, \beta \in \mathbb{R}^{+}$.
The proof of Lemma 5 is elementary calculations.
Conjecture 6. $x_{\max }\left(y_{i+1}\right) \leqslant x_{\min }\left(y_{i}\right)$, for $a>1$.
Proof. For some arbitrary $y \in\left[\frac{a-1}{\sqrt[3]{2}}, a-1\right)$, we know that

$$
\begin{align*}
x_{\max }\left(y_{i+1}\right) & =\left\lfloor\sqrt[3]{a^{3}-y^{3}}\right\rfloor  \tag{I.3.1}\\
x_{\min }\left(y_{i}\right) & =\left\lceil\sqrt[3]{(a-1)^{3}-(y-1)^{3}}\right\rceil \tag{I.3.2}
\end{align*}
$$

Letting $f_{a}(y)=\sqrt[3]{a^{3}-y^{3}}$, we can plot $f_{a}(y)$ and $f_{a-1}(y-1)$ to help visualize our upcoming argument.


Figure I.3.3: Plot of $\sqrt[3]{a^{3}-y^{3}}$ and $\sqrt[3]{(a-1)^{3}-(y-1)^{3}}$.
We know that $y \nless 1$, when $a>0$, so let us look at what happens when $y=1$.

$$
\begin{align*}
f_{a}(1) & =\sqrt[3]{a^{3}-1}<a  \tag{I.3.4}\\
f_{a-1}(1-1) & =\sqrt[3]{(a-1)^{3}}=a-1 \tag{I.3.5}
\end{align*}
$$

Since $f_{a}(1)-f_{a-1}(1-1)<1$, we know that $x_{\max }(1)-x_{\min }(1) \leqslant 0$, by Lemma 5 . In Figure I.3.3 we can see that $f_{a}(y)-f_{a-1}(y-1)$ is a decreasing function, which implies that $x_{\max }\left(y_{i+1}\right) \ngtr x_{\text {min }}\left(y_{i}\right)$ for any $y$-value, in its domain.

$$
\therefore x_{\max }\left(y_{i+1}\right) \leqslant x_{\min }\left(y_{i}\right), \text { for } a>1 .
$$

## I. $4 x_{\min }, x_{\max }$

For $a>1, x_{\text {min }}$ is simply 1 , which can be proven by elementary calculation.
While we have already calculated the ranges for $x$ in our previous algorithm (2.3.3), these are of no use if we want the actual value of $x_{\max }$. This stems from the fact that we are calculating the range for $x$, not a value. Some of the values in this range does not satisfy our initial condition

$$
\begin{equation*}
0<x \leqslant y, \tag{I.4.1}
\end{equation*}
$$

and gets filtered out in line 9 of Algorithm 2.3.3-"If x is less than or equal to y". Recall our calculation of $y_{\text {min }}$,

$$
y_{\min }=\left\lceil\frac{a-1}{\sqrt[3]{2}}\right\rceil .
$$

It might be tempting to think that we can remove the ceiling function from the calculation of $y_{\min }$ when calculating $x_{\max }$, but this will lead to errors like when $a=5$ :

$$
\begin{aligned}
y_{\text {min }} & =\left\lceil\frac{5-1}{\sqrt[3]{2}}\right\rceil \\
& =4 . \\
y & =\frac{5-1}{\sqrt[3]{2}} \\
& \approx 3.174802 \\
x_{\text {max }} \stackrel{?}{=} \sqrt[3]{a^{3}-y^{3}} & \approx 4.530655 \\
\lfloor 4.530655\rfloor & =4, \text { but } \\
4^{3}+4^{3} & =128 \nless 5^{3} .
\end{aligned}
$$

The reason that this problem arises stems from the fact there will not always be a point on the line $y=x$, as seen in Figure I.4.2.


Figure I.4.2: $\# \Sigma(15)$, where $y_{\min }=12$ and $x_{\max }=11$.

There is, however, an easy way to solve this problem. Following the same line of reasoning as when calculating $y_{\text {min }}$, we see that $2\left\lfloor\frac{a}{\sqrt[3]{2}}\right\rfloor^{3}<a^{3}$, which in turn leads to

$$
\begin{equation*}
x_{\max }=\left\lfloor\frac{a}{\sqrt[3]{2}}\right\rfloor . \tag{I.4.3}
\end{equation*}
$$

## Appendix II

## Code

## II. 1 For cube sums

Note that the following algorithms are optimized for readability rather than speed.

```
Algorithm II.1.1 Working version of the first algorithm (2.2.1)
def first_algorithm(a):
    " " "
    :param a: Integer
    :return: Number of unique cube sums between (a-1)**3 and a**3:
    """
    if type(a) == int and a > 0:
        sumlist = []
        for x in range(1, a):
            for y in range(1, a):
            calc = x**3 + y**3
            if (a-1)**3<calc<a**3:
                if calc not in sumlist:
                        sumlist.append(calc)
        return len(sumlist)
    else:
        print('This function only handles integers greater than 0.')
    return
```

The first algorithm have no dependencies but all of the following requires implementations of both the ceiling and floor functions, so let us define them before moving on. To save space we will skip the docstrings and input checks in the following code.

## Algorithm II.1.2 Ceiling function

```
    def cf(x):
        return int(-1*x//1*-1)
```

Algorithm II.1.3 Floor function

```
    def fl(x):
        return int(x//1)
```

```
Algorithm II.1.4 Working version of improved algorithm (2.3.3)
def second_algorithm(a):
    mylist = []
    ymin \(=c f((a-1) /(2 * *(1 / 3)))\)
    for \(y\) in range(ymin, a):
        \(x m i n=\max (1, \quad c f(((a-1) * * 3-y * * 3) * *(1 / 3)))\)
        xmax \(=f l((a * * 3-y * * 3) * *(1 / 3))+1\)
        for \(x\) in range(xmin, xmax):
            if \(0<x\) < \(y\) :
            temp \(=\mathrm{y} * * 3+\mathrm{x} * * 3\)
            if temp not in mylist:
                    mylist. append (temp)
    return len(mylist)
```

```
Algorithm II.1.5 Working version of fast algorithm 2.6.1
def fast_calc(a):
    counter \(=f l(a / 2 * *(1 / 3))\)
    for \(y\) in range (a-1, counter, -1):
        \(\operatorname{xmax}=\mathrm{fl}((\mathrm{a} * * 3-\mathrm{y} * * 3) * *(1 / 3))\)
        deltay \(=\mathrm{y}-((\mathrm{a}-1) * * 3-\mathrm{xmax} * * 3) * *(1 / 3)\)
        counter += fl(deltay)
    return counter
```


## II. 2 For points in circle

```
Algorithm II.2.1 Circle point algorithm
    def circle_points (r):
    if \(r==0\) :
            number_of_points \(=1\)
            return number_of_points
    elif \(r==1\) :
            number_of_points \(=5\)
            return number_of_points
    sq_points \(=(2 * r+1) * * 2\)
    base \(=c f(r *(2-2 * *(1 / 2)))\)
    tri_points \(=\) int (base* (base +1 )/2)
    x_init \(=c f(r / 2 * *(1 / 2))\)
    it_end \(=r+1\)
    iter_points \(=0\)
    for i in range((it_end - x_init)):
        x_it \(=x_{\text {_init }}+i\)
        ymax \(=c f\left(2 * *(1 / 2) * r-x \_i t\right)\)
        ymin \(=\left(r * * 2-x_{-} i t * * 2\right) * *(1 / 2)\)
        diff \(=\) ymax - ymin
        iter_points \(+=\) fl(diff)
        if diff - fl(diff) == 0 :
            iter_points -= 1
        number_of_points = sq_points - \(4 *\left(t r i \_p o i n t s+2 * i t e r \_p o i n t s\right)\)
    return number_of_points
```

Instead of commenting the code of II.2.1, let us draw a figure to help us understand it. By symmetry we only need to look at a quarter of a circle.


Figure II.2.2: Calculated points in circle with $r=8$.
Thinking inside the box, we see that there are far fewer points outside the circle but inside the square with side length $2 r$. So instead of counting all the points inside the circle, we calculate the points in the square, subtract four times the number of red points and eight times the number of green points. The number of points in the square is simply $(2 r+1)^{2}$, while the number of red points is $\frac{b(b+1)}{2}$, where $b=$ the number of points in the base of the red triangle. The only points left are the green ones, which are calculated by using the same approach as in the fast algorithm (2.6.1).

## II. 3 Performance

In table II.3.1 we see some performance comparisons of our algorithms. Note that these where carried out in an virtual environment on a old laptop, so the actual calculation times are less relevant then how they scale.

Table II.3.1: Calculation times in seconds.

|  | $a$ | 100 | 1000 | 10000 |
| :---: | :---: | :---: | :---: | :---: |
| Algorithm |  |  |  |  |
| First | 0.01459 | 1.76689 | 155.59696 | - |
| Improved | 0.00019 | 0.01149 | 0.94286 | 89.13900 |
| Fast | 0.00007 | 0.00070 | 0.00710 | 0.07515 |


[^0]:    ${ }^{1} \mathrm{~A}$ perfect square is of the form $n^{2}$, where $n \in \mathbb{N}$.

[^1]:    ${ }^{2}$ At least non-negative ones.

[^2]:    ${ }^{1}$ Initially the idea was that this paper would explore natural density, where Waring's problem was one of the topics to research. Since this paper no longer has that much in common with natural density, that part has been omitted.

[^3]:    ${ }^{2}$ That we can add $1 \leqslant x \leqslant a-1$ to $y=a-1$.
    ${ }^{3}$ While one could probably spend a life time coming up with sequences that is not in OEIS ${ }^{\circledR}$, this still seemed quite exciting, at the time.

[^4]:    ${ }^{4}$ We need to exclude duplicates. One example being $1^{3}+12^{3}=9^{3}+10^{3}=1729$.

[^5]:    ${ }^{5}$ Performance comparisons can be found in Appendix II.

[^6]:    ${ }^{6}$ We calculate if there can be a point under a given $y$-value, so there is no need to check under $y_{\text {min }}$.
    ${ }^{7}$ What is calculated by Algorithm 2.6.1

[^7]:    ${ }^{8}$ Note that this does not apply to all areas. Consider a rectangle made up of the points $\left(x_{0}, \eta\right),\left(x_{0}, 1-\eta\right),\left(x_{1}, \eta\right)$ and $\left(x_{1}, 1-\eta\right)$ for some infinitesimal $\eta \in \mathbb{R}$. This region will not contain any integer lattice points, no matter the values of $x_{0}, x_{1}$.

