Report for Project 1

October 27, 2022

1 Introduction

This project consists of three parts:

1. The approximation of the Catalan constant
2. Calculation of cubic taxicab number in two ways
3. Calculation of taxicab number in n ways

2 Task A

This task aims to find the best approximation of the Catalan constant among all possible positive integers p and q such that $p+q$ is smaller than or equal to the given positive integer N.

2.1 What is the Catalan constant?

The Catalan constant is a mathematical number defined as

$$\mathcal{C} = \sum_{0}^{\infty} \frac{(-1)^k}{(2k+1)^2} = \frac{1}{1^2} - \frac{1}{3^2} + \frac{1}{5^2} + \cdots \approx 0.915965594177219$$

The approximation $\mathcal{C} = 0.915965594177219$ is used in the following calculation.

2.2 How to find the best approximation with a given N?

There are two ways to compare the sizes, difference or quotient. The difference way is selected which is calculated by $\left|\frac{p}{q} - \mathcal{C}\right|$. The closer these two are to each other, the smaller the value will be, and when the approximation equals the Catalan constant, the value will be 0. So our purpose is to calculate the value of $\left|\frac{p}{q} - \mathcal{C}\right|$ for all the satisfied pairs of numbers (p,q) and find the minimum one.

2.2.1 How to traverse through all the pairs(p,q)?

Because there are two variables p and q where the inequality between them is $p+q \leq N$. First, restrict the range of p to $[1:N]$, and through the inequality, get the range of q for each determined p is $[1:N-p]$.
It is easy to traverse through all the (p,q) pairs.

2.2.2 How to find the best pair of numbers?

First, using $a = \left|\frac{p}{q} - \mathcal{C}\right|$ to simplify the expression. In the traversal process, for each pair of numbers calculate a. Since there is no minimum value used for comparison a at the beginning, let $min = a_1 = \left|\frac{p_1}{q_1} - \mathcal{C}\right| = 1 - \mathcal{C}$ with the first pair of numbers $(1, 1)$. Then, for the coming value of a_i, if it is smaller than the stored minimum, update the minimum value and the pair of numbers (p, q).
This leads to a question: because there is only one pair of numbers for the output what if the best approximation can be achieved by multiple pairs of numbers? The answer is to return the pair with the smallest $p + q$ value.
Figure 1 shows the process of finding the best pair of numbers.

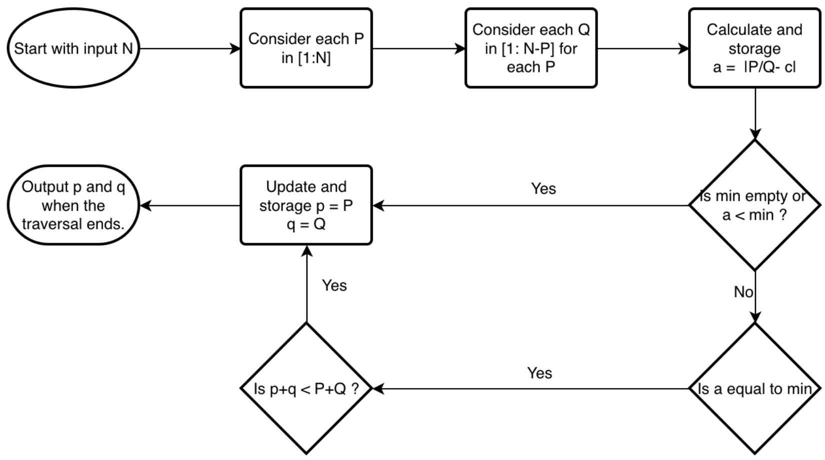


Figure 1: Catalan constant - AppCat

2.3 Result with given $N = 2022$

According to the above steps, the best approximation of the Catalan constant could be reached by the pair of numbers $(109, 119)$ with the difference being only 7.92378×10^{-7}.

3 Task B

Task B is to find the smallest cubic taxicab number M which is greater than or equal to the given positive integer N with positive integers a, b, c and d satisfied $M = a^3 + b^3 = c^3 + d^3$.

3.1 What are the Cubic taxicab numbers?

A cubic taxicab number is a positive integer that can be represented in two or more different ways as the sum of two positive cube numbers, for example, $4104 = 2^3 + 16^3 = 9^3 + 15^3$.

3.2 Solution ideas

Unlike what will be discussed in task C, this section only explicitly needs two pairs of solutions, so the problem turns into seeking two different solutions to a binary cubic equation. According to the previous section, there is a need to run a loop of unknown length until the required solution is found. And the solutions should all be positive integers. Thus, the general idea is to start solving the equation from a given N and increase N by one each time until finding a solution.
The process should be as shown in figure 2.

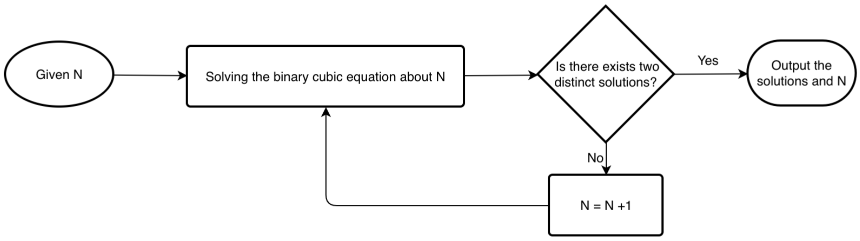
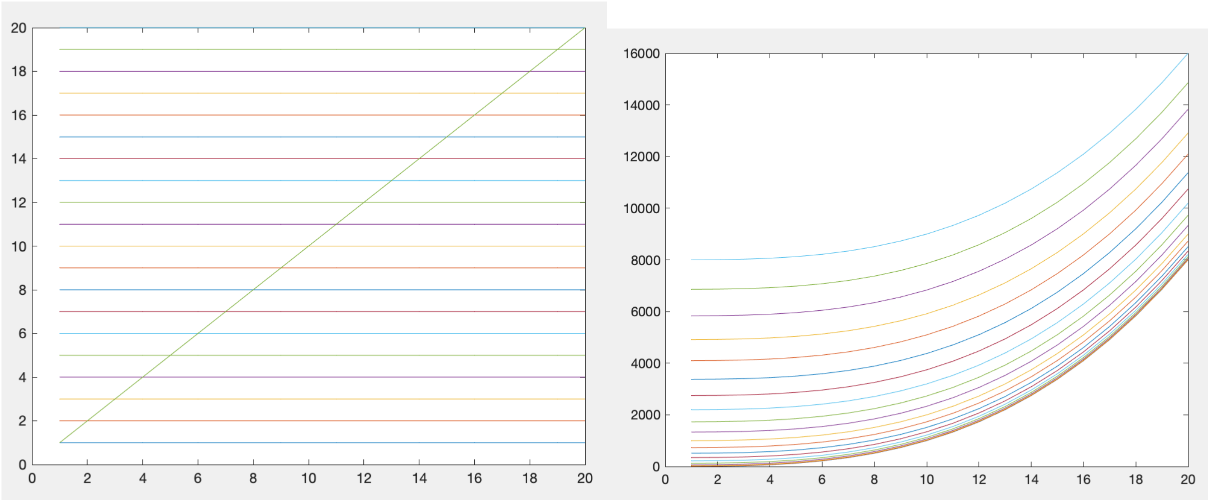


Figure 2: Cubic taxicab numbers - CubicTaxicabNum

But if conventional solving a binary cubic equation, there will inevitably be complex and fractional solutions. In most cases, numerous solutions are discarded to find the right one. Thus, try a different way to set the value of the solution to see if it satisfied the equation with the given N.
First, the upper bound of solutions is limited to the `n = floor(N^(1/3))` (floor is the function for rounded down, and 10*eps should be added in code to avoid unexpected outcomes as a result of floating-point operations) according to the equation.
Then, for the numbers in the range, using x and y to represent the unknowns of the equation, once there are at least two groups of (x, y) reaching N in the loop, choose them as the output.
So in MATLAB, let `n = floor(N^(1/3)+10*eps);` be the upper bound of solutions. Thus, the range for x and y becomes $[1 : n]$. Using `[x,y] = meshgrid(x,y);` to create a square grid coordinates with grid size n-by-n (like the one shown in Figure 3(a)). Then using `z = x.^3 + y.^3;` (Figure 3(b) shows the graph for z), where z is a set contains all the values of $x^3 + y^3$. After that, compare the number in z with N to see whether they are the same. Using the code `k = find(z==N);` find the solution set which contains all the values for outputting since a, b, c and d are equivalence in equations. Once the number of elements in `x(k);` is greater than or equal to 4 meets the requirements of outputting. Because it shows that there are different values in the set that could be assigned to a, b, c and d.



(a) [x,y] grid coordinates with n=20 (b) value of z with increasing x and y

Figure 3: meshgrid graphics

3.3 Result with given N=36031

According to the above ideas, the smallest cubic taxicab number M greater than the given lower bound 36031 is 39312 and is equal to $2^3 + 34^3 = 15^3 + 33^3$.

4 Task C

This task is a further study of task B, a new notation T_n is defined as the smallest positive number that can be written as the sum of two positive integers cubed in n different ways. For example,

$$T_1 = 2 = 1^3 + 1^3;$$

$$T_2 = 1729 = 1^3 + 12^3 = 9^3 + 10^3.$$

This task aims to find T_3 outputting the corresponding pairs of integers, and it is extendable to T_n with just a few changes to the previous code.

4.1 Improvement ideas

Because the number of pairs for outputting has increased from 2 pairs to n pairs while the input changes from a given lower bound N to the number of distinct ways n in taxicab number, change the conditions for the loop. Find the solution in the same way as above, but construct a new variable lensol(an abbreviation for `length(solution)`) to store the number of solutions. The length of the solution set should be twice the input n. Then the condition becomes "Is the value of lensol less than $2n$?", if the answer is no, continue the loop, else output results.
The process should be as shown in figure 4.

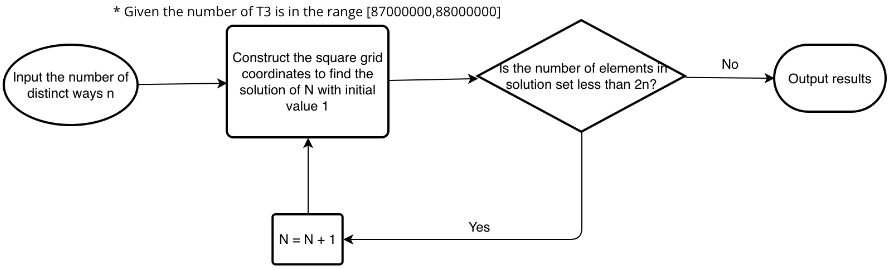


Figure 4: Taxicab numbers - Tn

4.2 Result for n = 3

Combine the numbers in the solution set in two-by-two order from the ends to the middle since the elements in the set are arranged by default from smallest to largest. Then the answer comes out: the taxi number for $n = 3$ is 87539319, and the corresponding pairs of numbers could be shown in the equation below:

$$T_3 = 87539319 = 167^3 + 436^3 = 228^3 + 423^3 = 255^3 + 414^3$$

5 Appendix - MATLAB code

Outputs are shown by `format compact`

01 - AppCat

```
function [p,q] = AppCat(N)
% APPCat Approximates the catalan constant by p/q
% among all pairs of positive integers (p,q) such that p + q <= N
catconst =0.915965594177219;% Catalan constant
min = NaN;P = NaN;Q = NaN; % Set initial values to NaN
for p = 1:N
    for q = 1:(N-p)
        value = abs(p/q-catconst);% Aim to find the smallest one
        if isnan(min) || value < min
            min = value;
            P = p; Q = q;% Update the output
        elseif value == min
            if p+q < P+Q % Select the one with smaller sum
                P = p;Q = q;
            end
        end
    end
end
p = P;q = Q;
disp(['The best approximates to the catalan constant ' ...
```

```
    'satisfies p+q<=' num2str(N) ' is p=' num2str(p) ' and q=' ...
    num2str(q) ' with difference ' num2str(min) '.'])
end
```

02 - running AppCat from command line

```
>> [p,q]=AppCat(2022)
The best approximates to the Catalan constant satisfies p+q<=2022 is
   p=109 and q=119 with difference 7.9238e-07.
p =
   109
q =
   119
```

03 - CubicTaxicabNum

```
function [a,b,c,d,M] = CubicTaxicabNum ( N )
% CUBICTAXICABNUM returns the smallest cubic taxicab number M
% M=a^3+b^3=c^3+d^3 greater than or equal to N
t = N; % storage of the initial N for disp
lensol = 0;% set the initial number for the loop
while lensol<4
    n = floor(N^(1/3)+10*eps);% the upper bound for a,b,c,d
    x = 1:n;y = 1:n;
    [x,y] = meshgrid(x,y);
    z = x.^3 + y.^3;
    k = find(z==N);
    lensol = length(x(k));
    N = N+1;
end
sol = x(k); % the solution set
a = sol(1);b = sol(end);c = sol(2);d = sol(end-1);
M = N-1; % there is an extra 1 in the last loop
disp(['The smallest cubic taxicab number M greater than ' ...
    num2str(t) ' is ' num2str(M) ' and is equal to ' ...
    num2str(a) '^3+' num2str(b) '^3=' num2str(c) '^3+' ...
    num2str(d) '^3.'])
end
```

04 - running CubicTaxicabNum from command line

```
The smallest cubic taxicab number greater than 36031 is 39312 and is
   equal to 2^3+34^3=15^3+33^3.
a =
     2
b =
    34
```

```
c =
    15
d =
    33
M =
       39312
```

05 - Tn

```
function [xy,M] = Tn (n)
% Tn returns the smallest taxicab number M that can be expressed
% as the sum of two positive integers cubed in n distinct ways.
lensol = 0;% set the initial number for the loop
N = 1;
xy = '';
if n == 1 % 1 cannot show twice in the vector using this way
    M = 2;% so list such cases separately
    xy=[xy, '(1,1)'];
else
    while lensol<2*n
        if n == 3 && N == 1
            N = 87000000;% the lower bound for n=3
        end
        j = floor(N^(1/3)+10*eps);% the upper bound for solutions
        x = 1:j;y = 1:j;
        [x,y] = meshgrid(x,y);
        z = x.^3 + y.^3;
        k = find(z==N);
        lensol = length(x(k));% storage the length of soluiton set
        N = N+1;
    end
    s = x(k);
    for i = 1:(length(s)/2)
        xy = [xy,'(',num2str(s(i)),',',num2str(s(end+1-i)),')'];
    end
    M = N-1;
end
disp(['For T' num2str(n) ', the smallest integer is ' ...
    num2str(M) ', and the corresponding pairs of numbers are ' ...
    xy ['. For each pair of numbers (xi,yi) satisfying ' ...
    '(xi)^3+(yi)^3='] num2str(M) '.'])
end
```

06 - running taxiTn from command line

```
>> [Pairs,M] = Tn (3)
```

```
For T3, the smallest integer is 87539319, and the corresponding pairs of
   numbers are (167,436)(228,423)(255,414). For each pair of numbers
   (xi,yi) satisfying (xi)^3+(yi)^3=87539319.
xy =
    '(167,436)(228,423)(255,414)'
M =
       87539319
```