

# Goldbach's Conjecture

Nicolas Desan

February 2024

## 1 Introduction

Goldbach's conjecture is one of the oldest and best-known unsolved problems in number theory and mathematics. The endeavor and fascination around the Goldbach conjecture is beautifully portrayed in the movie "Le Théorème de Marguerite." Celebrated at the Cannes Festival 2023 and awarded the "Grand Prix du Public" at the Cabourg Festival, with a nomination for the 2024 César Awards, the movie masterfully weaves the elegance of mathematical theories with the profound human endeavor to uncover the mysteries of the universe.

The Goldbach's conjecture states that every even natural number greater than 2 is the sum of two prime numbers.

The conjecture has been shown to hold for all integers less than  $4.10 * 10^{18}$  but remains unproven despite considerable effort. The purpose here is to prove the Goldbach's conjecture for the maximum number of even natural numbers with a Python code and then a C++ code and see how we can improve the performances.

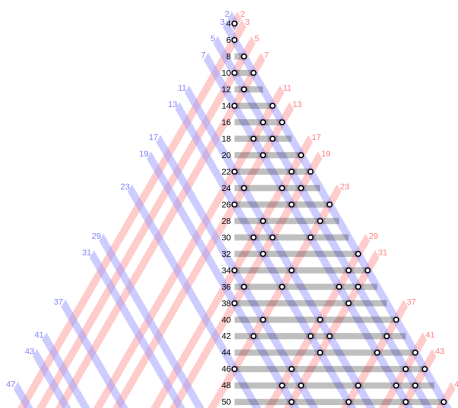


Figure 1: Pyramide of Goldbach that illustrates his conjecture



Figure 2: Le Théorème de Marguerite

## 2 Implementation

### 2.1 List of prime numbers

First of all, we made a python code that gets all the prime numbers under  $n$  :

```
def premier(n):
    if n % 2 != 0 or n <= 2:
        return False

    l=[2]
    for i in range(3,n,2):
        #only even numbers under n to increase performances
        j=1
        k=0
        while(math.gcd(j,i)==1 and k<len(l)-1):
            j=l[k]
            k+=1
        if(k==len(l)-1):
            l.append(i)
    return l
```

## 2.2 Conjecture proved at range 2N

Here, we consider an even number and we want to prove that it verifies the Goldbach conjecture. Then we only need to generate the list of the prime numbers with `premier(2N)` function and to find a pair that verifies the conjecture ( $2N = p + q$  with  $p$  and  $q$  two prime numbers).

In order to do so, we just need to find an  $i$  in the prime list number such that  $(2N-i)$  is also a prime number.

```
def goldbach(n, liste_premiers):
    if n%2 != 0:
        print("Il faut que n soit paire")

    for j in liste_premiers:
        if (n-j) in liste_premiers:
            print(f"{n} = {j} + {n - j}")
            return True
    return False
```

## 2.3 Proving the conjecture for every even natural number under 2N

We just make an iteration of the previous code with a prime list generated before :

```
def test(n):
    primes=premier(n)
    for i in range(4,n+1,2):
        if not(goldbach(i, primes)):
            return False
    return True
```

With this code, we could prove the Goldbach conjecture for each even number under 800 000.