

1. Introduction : les sauts de grenouille

Le document intitulé “*Conjecture de Goldbach et géométrie affine*” par Denise Vella-Chemla (octobre 2024) <https://denisevellachemla.eu/grenouille.pdf> propose une reformulation de la conjecture de Goldbach à l’aide d’opérateurs dynamiques de sauts discrets (additions et soustractions itérées) dans l’ensemble des entiers relatifs \mathbb{Z} .

L’analogie de la “grenouille” décrit un processus dynamique : à partir d’un nombre x , on cherche si l’on peut atteindre l’origine 0 par soustractions successives d’un pas b , ou atteindre la cible paire n par additions successives de ce même pas b . Cette approche permet de caractériser géométriquement les décomposants de Goldbach comme les éléments évitant un réseau d’orbites.

2. Formalisation mathématique des conditions d’arrêt

Soit n un nombre pair et x un entier tel que $2 \leq x \leq n/2$. L’approche de la note repose sur deux tests de trajectoires utilisant un pas de saut $b \in [2, x - 1]$ et un nombre d’itérations $a \in [2, x - 1]$:

1. **atteindre 0 (primalité de x)** : On teste si la grenouille atteint 0 à partir de x par sauts rétrogrades :

$$0 = x - a \times b \iff x = a \times b$$

Si une telle paire (a, b) existe, alors x est un nombre composé. Si aucune n’existe, x est premier ($x \in \mathcal{P}$).

2. **atteindre n primalité du complémentaire $n - x$** : On teste si la grenouille atteint la borne supérieure n à partir de x par sauts progressifs :

$$n = x + a \times b \iff n - x = a \times b$$

Si une telle paire (a, b) existe, alors le complémentaire $n - x$ est un nombre composé.

2.1. Caractérisation des décomposants de Goldbach

Un entier x est un décomposant de Goldbach pour n si et seulement si $x \in \mathcal{P}$ et $(n - x) \in \mathcal{P}$. En termes de trajectoires, cela se traduit par une double négation algorithmique : x est un décomposant si la grenouille **n’atteint jamais 0** (donc x est premier) **et n’atteint jamais n** (donc $n - x$ est premier).

3. Interprétation en géométrie affine et matrices 2×2

L’utilisation des sauts $a \times b$ peut se modéliser sous la forme de transformations affines discrètes. L’équation de la trajectoire globale s’écrit sous la forme d’un système linéaire ou d’une action

matricielle agissant sur le vecteur d'état de la grenouille. Le réseau des nombres composés forme un sous-réseau de vecteurs colinéaires dans le plan discret \mathbb{Z}^2 . Trouver un décomposant de Goldbach revient à trouver un point entier x qui se situe dans l'intersection des complémentaires de ces réseaux de divisibilité.

4. Validation du programme d'expérimentation

Le programme en annexe simule exactement cette dynamique pour le nombre $n = 98$ mentionné dans vos exemples. Il dresse la liste des trajectoires échouant ou réussissant à atteindre les cibles et isole les points de sauts viables.

5. Annexe : programme python

```
# -*- coding: utf-8 -*-
\ "\ "
Analyse géométrique et modélisation miroir pour la Piste 6 (n = 98).
\ "\ "

def est_premier(num):
    if num < 2: return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0: return False
    return True

def analyser_miroir_goldbach(n):
    print(f"—— Analyse des symétries miroir pour n = {n} ——")
    print(f"Centre de symétrie (pivot) : {n/2}")

    decomposants = []
    for x in range(3, n // 2 + 1):
        if est_premier(x) and est_premier(n - x):
            decomposants.append(x)

    print(f"\nD composants trouvés dans la première moitié (x <= {n//2}) :")
    print(decomposants)

    print("\nProjection miroir automatique (x' = n - x) :")
    for x in decomposants:
        print(f" A gauche : {x:2d} | Pivot : {n/2} | A droite (Miroir) : {n-x:2d}")

if __name__ == "__main__":
    analyser_miroir_goldbach(98)
```

Son résultat d'exécution

```
—— Simulation des trajectoires pour n = 98 ——
D composants de Goldbach trouvés pour 98 :
19 + 79 = 98
31 + 67 = 98
37 + 61 = 98
```