

Conjecture de Goldbach, Conjecture des nombres premiers jumeaux, test de primalité et sinusoides

Denise Vella-Chemla

27 Mai 2009

1 Introduction

Dans cette note est présentée un algorithme de recherche des décomposants de Goldbach qui a permis d'aboutir à une façon unifiée de considérer la conjecture de Goldbach et la conjecture des nombres premiers jumeaux en utilisant un test de primalité basé sur le calcul de produits de sinus.

2 Présentation d'un exemple qui explicite la méthode des sinusoides

Intéressons nous au cas $2a = 60$ (ou bien $a = 30$). On va noter dans un tableau par des 0 le fait que certains nombres soient divisibles par les nombres impairs compris entre 3 et $2 * \lfloor \sqrt{30} \rfloor + 1$.

57	55	53	51	49	47	45	43	41	39	37	35	33	31	
3	5	7	9	11	13	15	17	19	21	23	25	27	29	
				0	–	–	–	–	–	–	–	0	–	<i>divisibilité par 11</i>
			0	–	–	0	–	–	–	–	–	0	–	<i>divisibilité par 9</i>
		0	–	0	–	–	–	–	0	–	0	–	–	<i>divisibilité par 7</i>
	0	–	–	–	–	0	–	–	–	–	0	–	–	<i>divisibilité par 5</i>
0	–	–	0	–	–	0	–	–	0	–	–	0	–	<i>divisibilité par 3</i>

Sur la dernière ligne, on constate que 3 divisant 30 (ou 60), les nombres dont la somme vaut 60 (comme 3 et 57, ou bien 9 et 51) sont simultanément divisibles par 3.

Même constat sur la ligne montrant la divisibilité par 5.

Par contre, 7, 9 et 11 ne divisant pas 30, la divisibilité des nombres a été codée en noir pour les nombres inférieurs à 30 et en bleu pour ceux supérieurs à 30.

On comprend aisément que si l'on indice les colonnes par les nombres de 1 à $\lfloor \frac{a-1}{2} \rfloor$ au lieu de les indiquer par les nombres impairs comme on l'a fait, on peut

“simuler” les caractères de la divisibilité par les nombres impairs successifs par des sinusoides. On utilisera la sinusoides $\sin\left(\frac{(x-i)\pi}{2j+1}\right)$ quand on aura besoin d’une sinusoides de période $2j+1$ qui s’annule pour la valeur i . Les nombres qui n’annulent aucune des sinusoides ainsi définies sont en bijection immédiate avec les décomposants de Goldbach de $2a$.

Pour les lignes des nombres impairs qui divisent $2a$, on a besoin d’une sinusoides, pour les lignes des nombres impairs qui ne divisent pas $2a$, on a besoin de deux sinusoides.

La question de la conjecture de Goldbach devient : “ a étant supérieur ou égal à 11, pourquoi existe-t-il obligatoirement un nombre entier x supérieur ou égal à \sqrt{a} et inférieur ou égal à $\left\lfloor \frac{a-1}{2} \right\rfloor$ et tel que :

$$\prod_{\substack{1 \leq k \leq \sqrt{a} \\ 1 \leq k' \leq 2k+1 \\ k' \neq k \\ k \in \mathbb{N} \\ k' \in \mathbb{N}}} \sin\left(\frac{(x-k)\pi}{2k+1}\right) \cdot \sin\left(\frac{(x-k')\pi}{2k+1}\right) \neq 0$$

?” Ce produit fait intervenir $2\lfloor\sqrt{a}\rfloor$ sinusoides. La condition $k' \neq k$ est une condition non obligatoire : puisque c’est la non-nullité des sinusoides qui nous intéresse, on peut bêtement élever la sinusoides “centrale” au carré, cela ne modifie pas le résultat obtenu.

On calcule k' de la façon suivante :

$$k' = (x \bmod (2k+1)) + k.$$

On peut se passer de la fonction *mod* (qui renvoie le reste modulaire) en calculant k' ainsi

$$k' = x - (2k+1) * \lfloor x/(2k+1) \rfloor.$$

En annexe 1 est fourni le programme qui calcule les décomposants de Goldbach des nombres pairs compris entre 26 et 100 par cette méthode ainsi que son résultat en annexe 2.

3 Conjecture des nombres premiers jumeaux et test de primalité

On réalise que l’on peut utiliser le programme présenté ci-dessus, moyennant quelques petites modifications, d’une part pour tester la primalité d’un nombre et d’autre part, pour voir si ce nombre n’est pas un nombre pair exactement compris entre deux nombres premiers (que l’on appelle des nombres premiers jumeaux) : il suffit pour ce faire de ne tester non pas tous les sinus, mais uniquement les sinus qui interviennent dans la dernière colonne, celle d’indice $colonne = \left\lfloor \frac{a-1}{2} \right\rfloor$. En annexes 3 et 5, sont fournis les deux programmes correspondant et leur résultat sont dans les annexes 4 et 6.

Ces programmes n’ont aucun intérêt pratique : les algorithmes utilisés en cryptographie présentent toutes les qualités d’efficacité nécessitées par un tel domaine d’application. La méthode présentée semble seulement intéressante car elle offre un cadre commun à ces deux problèmes classiques de théorie des

nombres, et permet de les traiter en s'affranchissant de la notion de primalité : on a considéré tous les nombres impairs et non seulement les nombres premiers, dans la mesure où on ne sait pas où ceux-ci sont placés dans la suite numérique.

En annexe 7 sont fournies les sinusoïdes associées aux nombres pairs de 26 à 100, les zéros de la première sinusoïde sont systématiquement codés en noir tandis que ceux de la deuxième (dans le cas où $2k+1$ ne divise pas $2a$) sont codés en bleu. Un regard attentif permet de bien voir le fait que les sinusoïdes bleues "avancent" d'un cran vers la droite à chaque nouveau nombre pair (parfois, les cases bleues "se cachent derrière" les cases noires). On a parfois noté à côté du nombre pair considéré la lettre P pour indiquer qu'il s'agit du double d'un nombre premier, et parfois noté la lettre J pour indiquer qu'il s'agit du double d'un nombre pair qui est entre deux nombres premiers jumeaux.

Bibliographie

- [1] C.F. Gauss, *Recherches arithmétiques*, Ed. Jacques Gabay, 1801.
- [2] L. Euler, *Découverte d'une loi tout extraordinaire des nombres par rapport à la somme de leurs diviseurs*, Commentatio 175 indicis Enestroemiani, Bibliothèqure impartiale 3, 1751, p.10-31.
- [4] C. Laisant, *Sur un procédé de vérification expérimentale du théorème de Goldbach*, Bulletin de la SMF n°25, p.108, 1/12/1897.
- [5] C.P. Bruter, *La construction des nombres*, Ed. Ellipses, 2000.
- [6] D. Vella-Chemla, *Conjecture de Goldbach et résidus minima absolus de Gauss*, mai 2009.

Annexe 1 : Source du programme de calcul des décomposants de Goldbach des nombres pairs de 26 à 100

```
#include <iostream>
#include <cmath>

int main (int argc, char* argv[])
{
    const double pi=acos(-1) ;
    int x, ligne, colonne, moitie, decomp ;
    float tempo1, tempo2 ;
    int tab[100][100] ;

    for (x=13 ; x <= 50 ; x++)
    {
        std::cout << "\n" ;
        moitie = (x-1)/2 ;
        for (ligne = 1 ; ligne <= floor(sqrt(x)) ; ligne++)
            for (colonne = 1 ; colonne <= moitie ; colonne++)
                {
                    tab[ligne][colonne] = 1 ;
                    tempo1 = ((colonne-ligne)*pi)/(2*ligne+1) ;
                    if ((sin(tempo1) < 0.00001) && (sin(tempo1) > -0.00001))
                        tab[ligne][colonne] = 0 ;
                    tempo2 = ((colonne-((x % (2*ligne+1))+ligne))*pi)/(2*ligne+1) ;
                    if ((sin(tempo2) < 0.00001) && (sin(tempo2) > -0.00001))
                        tab[ligne][colonne] = 0 ;
                }
        for (colonne = 1 ; colonne <= moitie ; colonne++)
        {
            decomp = 1 ;
            for (ligne = 1 ; ligne <= floor(sqrt(x)) ; ligne++)
                if (tab[ligne][colonne] == 0) decomp = 0 ;
            if (decomp == 1 )
            {
                std::cout << 2*colonne+1 << " est un décomposant de " ;
                std::cout << 2*x << "\n" ;
            }
        }
    }
}
```

Annexe 2 : Résultat du programme de calcul des décomposants de Goldbach des nombres pairs de 26 à 100

```
13 est un décomposant de 26
11 est un décomposant de 28
11 est un décomposant de 30
13 est un décomposant de 30
13 est un décomposant de 32
11 est un décomposant de 34
17 est un décomposant de 34
13 est un décomposant de 36
17 est un décomposant de 36
19 est un décomposant de 38
11 est un décomposant de 40
17 est un décomposant de 40
11 est un décomposant de 42
13 est un décomposant de 42
19 est un décomposant de 42
13 est un décomposant de 44
```

17 est un décomposant de 46
23 est un décomposant de 46

11 est un décomposant de 48
17 est un décomposant de 48
19 est un décomposant de 48

13 est un décomposant de 50
19 est un décomposant de 50

23 est un décomposant de 52

13 est un décomposant de 54
17 est un décomposant de 54
23 est un décomposant de 54

13 est un décomposant de 56
19 est un décomposant de 56

17 est un décomposant de 58
29 est un décomposant de 58

13 est un décomposant de 60
17 est un décomposant de 60
19 est un décomposant de 60
23 est un décomposant de 60
29 est un décomposant de 60

19 est un décomposant de 62
31 est un décomposant de 62

17 est un décomposant de 64
23 est un décomposant de 64

13 est un décomposant de 66
19 est un décomposant de 66
23 est un décomposant de 66
29 est un décomposant de 66

31 est un décomposant de 68

17 est un décomposant de 70
23 est un décomposant de 70
29 est un décomposant de 70

19 est un décomposant de 72
29 est un décomposant de 72
31 est un décomposant de 72

31 est un décomposant de 74
37 est un décomposant de 74

17 est un décomposant de 76
23 est un décomposant de 76
29 est un décomposant de 76

17 est un décomposant de 78
19 est un décomposant de 78
31 est un décomposant de 78
37 est un décomposant de 78

19 est un décomposant de 80
37 est un décomposant de 80

23 est un décomposant de 82
29 est un décomposant de 82
41 est un décomposant de 82

17 est un décomposant de 84
23 est un décomposant de 84
31 est un décomposant de 84
37 est un décomposant de 84
41 est un décomposant de 84

19 est un décomposant de 86
43 est un décomposant de 86

17 est un décomposant de 88
29 est un décomposant de 88
41 est un décomposant de 88

17 est un décomposant de 90
19 est un décomposant de 90
23 est un décomposant de 90
29 est un décomposant de 90
31 est un décomposant de 90
37 est un décomposant de 90
43 est un décomposant de 90

19 est un décomposant de 92
31 est un décomposant de 92

23 est un décomposant de 94
41 est un décomposant de 94
47 est un décomposant de 94

17 est un décomposant de 96
23 est un décomposant de 96
29 est un décomposant de 96
37 est un décomposant de 96
43 est un décomposant de 96

19 est un décomposant de 98
31 est un décomposant de 98
37 est un décomposant de 98

17 est un décomposant de 100
29 est un décomposant de 100
41 est un décomposant de 100
47 est un décomposant de 100

Annexe 3 : Source du programme de test de la primalité des nombres de 13 à 99

```
#include <iostream>
#include <cmath>

int main (int argc, char* argv[])
{
    const double pi=acos(-1) ;
    int x, ligne, colonne, moitie, premier ;
    float tempol, tempo2 ;

    for (x=13 ; x <= 99 ; x=x+2)
    {
        moitie = (x-1)/2 ;
        premier = 1 ;
        for (ligne = 1 ; ligne <= floor(sqrt(x)) ; ligne++)
        {
            tempol = ((moitie-ligne)*pi)/(2*ligne+1) ;
            if ((sin(tempol) < 0.00001) && (sin(tempol) > -0.00001))
                premier = 0 ;
            tempo2 = ((moitie-((moitie % (2*ligne+1))+ligne))*pi)/(2*ligne+1) ;
            if ((sin(tempo2) < 0.00001) && (sin(tempo2) > -0.00001))
                premier = 0 ;
        }
        if (premier == 1) std::cout << x << " _\n";
    }
}
```

Annexe 4 : Résultat du programme de test de la primalité des nombres de 13 à 99

```
13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89
97
```

Annexe 5 : Source du programme de recherche des nombres pairs entre deux nombres premiers jumeaux

```
#include <iostream>
#include <cmath>

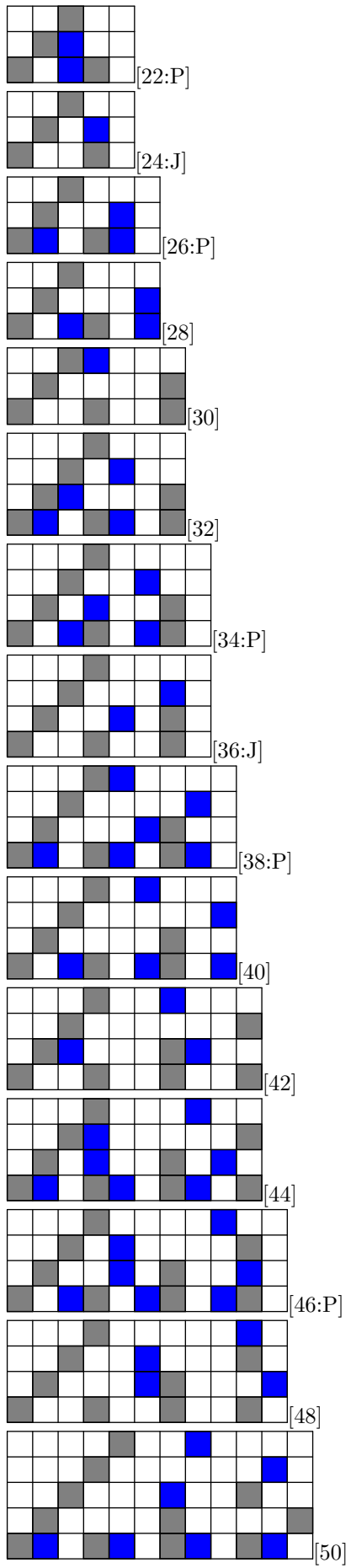
int main (int argc, char* argv[])
{
    const double pi=acos(-1) ;
    int x, ligne, colonne, moitie, jumeaux ;
    float tempo1, tempo2 ;

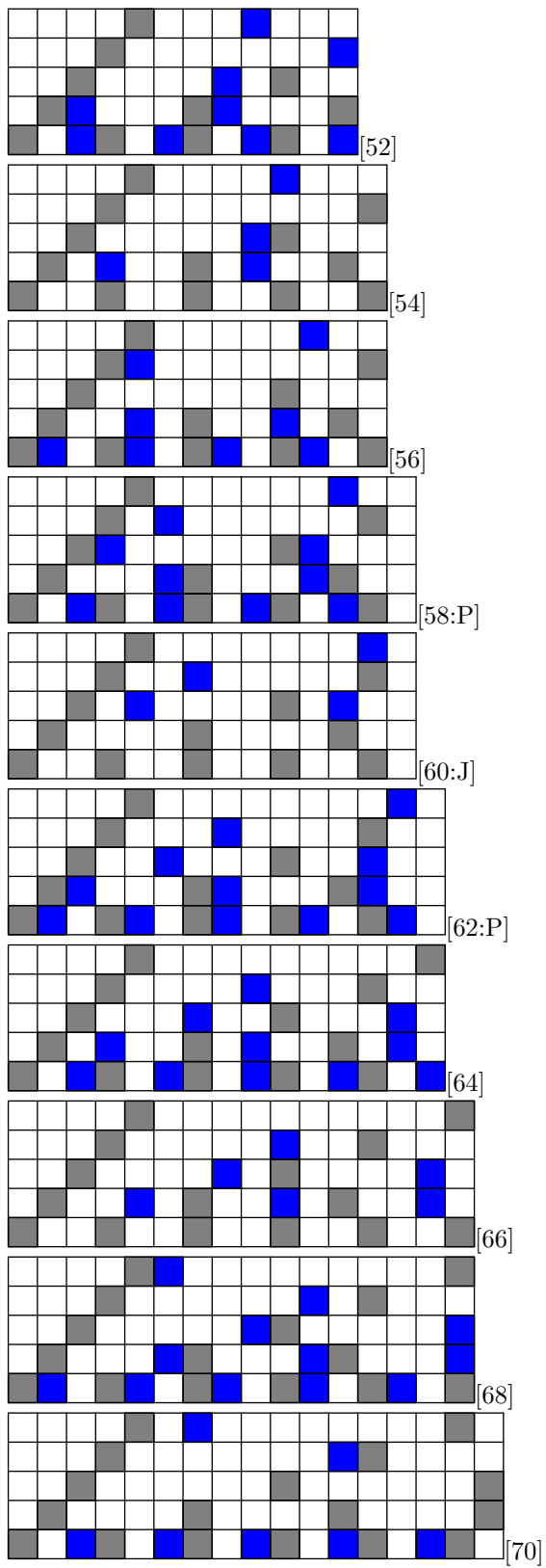
    for (x=14 ; x <= 100 ; x=x+2)
    {
        moitie = (x-1)/2 ;
        jumeaux = 1 ;
        for (ligne = 1 ; ligne <= floor(sqrt(x)) ; ligne++)
        {
            tempo1 = ((moitie-ligne)*pi)/(2*ligne+1) ;
            if ((sin(tempo1) < 0.00001) && (sin(tempo1) > -0.00001))
                jumeaux = 0 ;
            tempo2 = ((moitie-((x % (2*ligne+1))+ligne))*pi)/(2*ligne+1) ;
            if ((sin(tempo2) < 0.00001) && (sin(tempo2) > -0.00001))
                jumeaux = 0 ;
        }
        if (jumeaux == 1) std::cout << x << "_est_entre_deux_nombres_premiers.\n" ;
    }
}
```

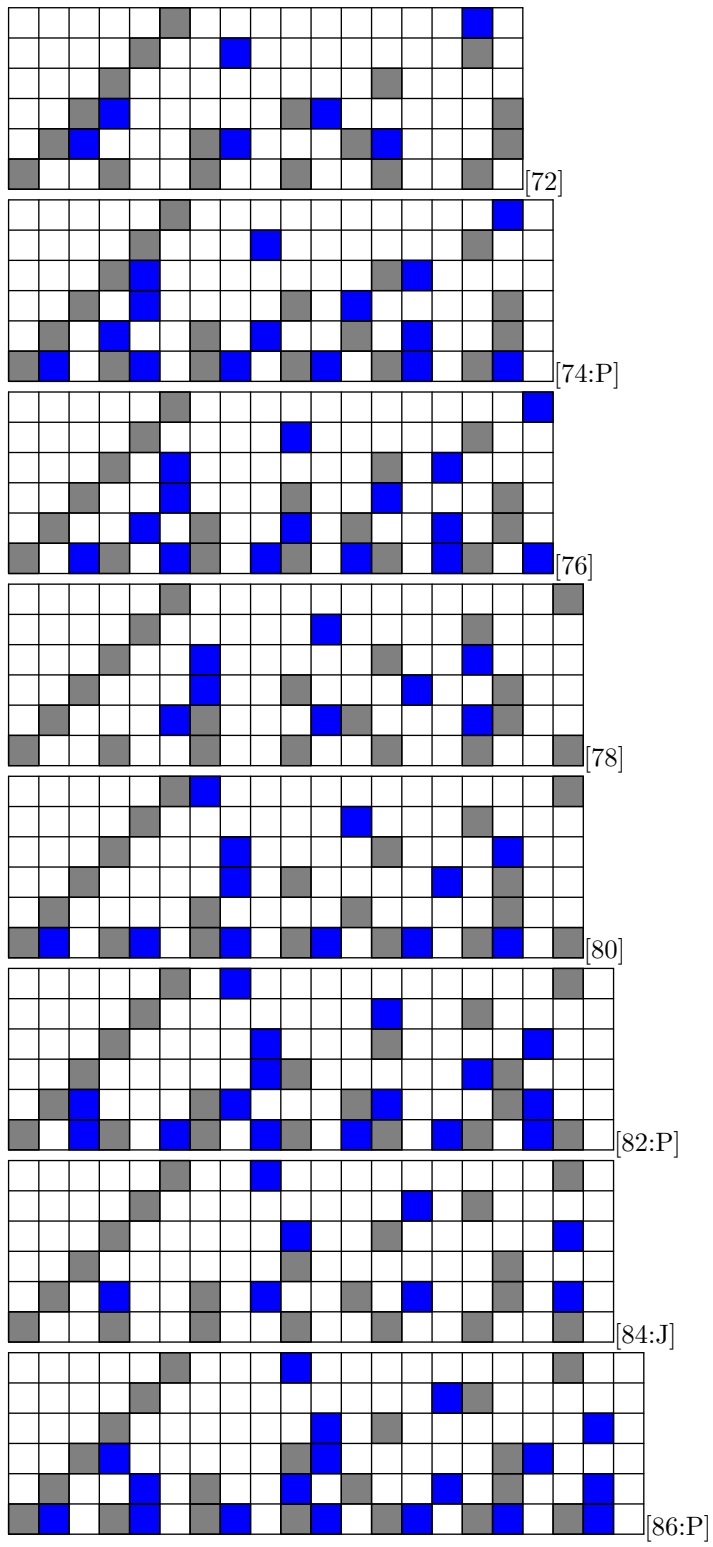
Annexe 6 : Résultat du programme de recherche des nombres pairs entre deux nombres premiers jumeaux

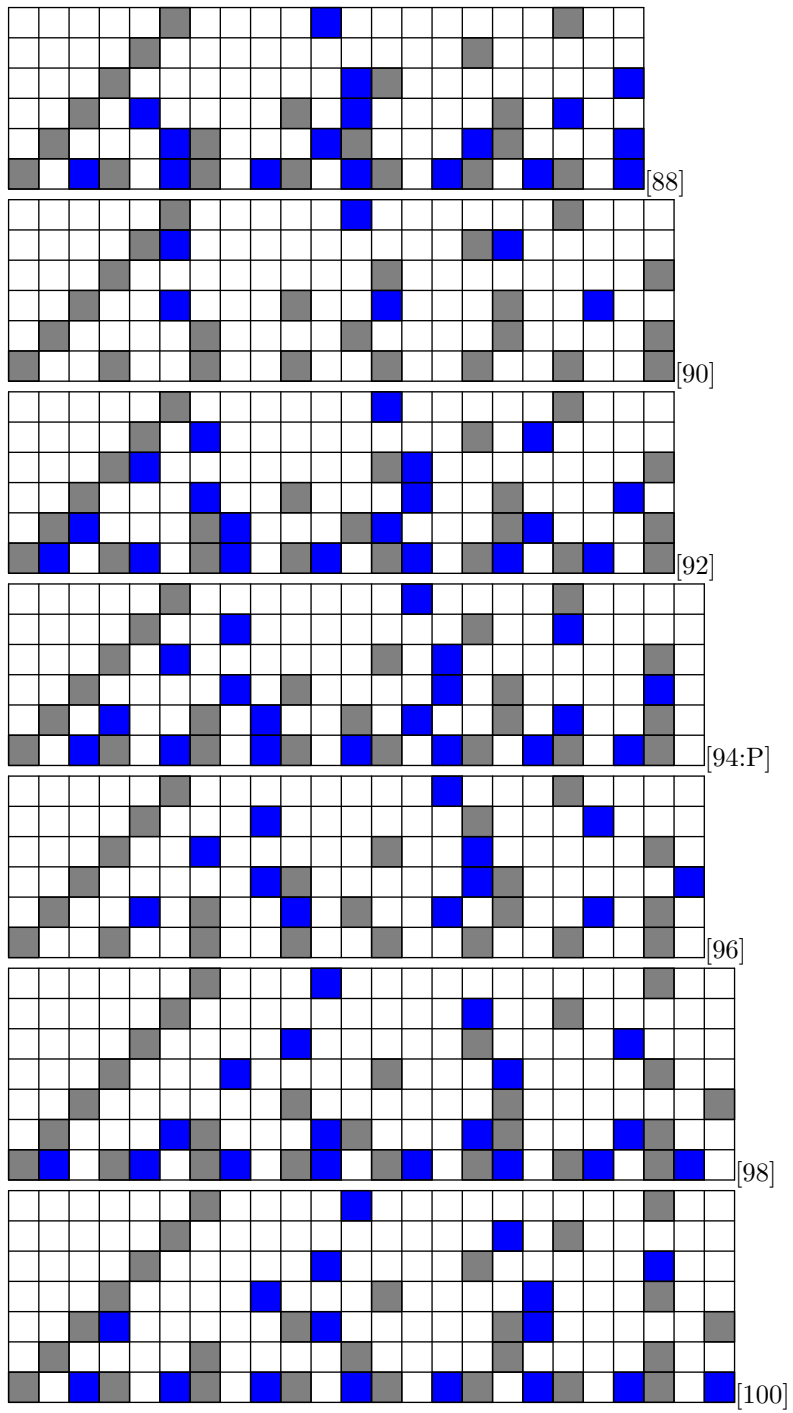
```
18 est entre deux nombres premiers.
30 est entre deux nombres premiers.
42 est entre deux nombres premiers.
60 est entre deux nombres premiers.
72 est entre deux nombres premiers.
```

Annexe 7 : Représentation graphique des sinusôides associées aux nombres pairs de 26 à 100









Annexe 8 : Conjecture de Legendre

La conjecture de Legendre stipule qu'il y a toujours un nombre premier entre n^2 et $(n + 1)^2$. Cela est équivalent à démontrer qu'il y a toujours un double de premier entre $2n^2$ et $2(n + 1)^2$. Dans la représentation présentée ici, le passage au carré suivant consiste à incrémenter de 1 le nombre de lignes des grilles. Il faudrait tester les produits de sinus de toutes les dernières colonnes des grilles ayant un même nombre de lignes. Ce que dit la conjecture de Legendre est équivalent à dire que l'un de ses produits de sinus est non nul.