Conjecture de Goldbach et mouvement brownien

Denise Vella-Chemla

22/3/14

On se place dans la suite dans un espace cartésien à deux dimensions.

On va associer à chaque nombre pair un *"déplacement global dans le plan"*, qui sera constitué de la suite des déplacements associés aux différentes décompositions de ce nombre pair comme somme de deux nombres impairs [1]. Le déplacement associé à tout nombre pair aura comme origine le point $(0, 0)$.

On codera :

- une décomposition de n de la forme $p + q$ avec p et q premiers et $p \leqslant n/2$ par l'augmentation de 1 de l'abscisse du point sur lequel on se trouve ;
- une décomposition de n de la forme $p + q$ avec p impair composé et q premier et $p \leqslant n/2$ par l'augmentation de 1 de l'ordonnée du point sur lequel on se trouve ;
- une décomposition de n de la forme $p + q$ avec p premier et q impair composé et $p \leqslant n/2$ par la diminution de 1 de l'ordonnée du point sur lequel on se trouve ;
- une décomposition de n de la forme $p + q$ avec p et q impairs composés et $p \leqslant n/2$ par la diminution de 1 de l'abscisse du point sur lequel on se trouve.

Ainsi, des conditions identiques de primalité pour p et pour $n - p$ son complémentaire se traduiront par des déplacements horizontaux (le vecteur $(1, 0)$ correspond à deux nombres premiers [2] et le vecteur $(-1, 0)$ correspond à deux nombres composés [3]). Des conditions différentes de primalité pour p et pour $n-p$ son complémentaire se traduiront quant à elles par des déplacements verticaux (le vecteur $(0, 1)$ correspond aux sommes $p+q$ avec p impair composé $\leqslant n/2$ et q premier [4] et le vecteur $(0, -1)$ correspond aux sommes $p + q$ avec p premier $\leqslant n/2$ et q impair composé [5]).

Exemple : déplacement global associé au nombre pair 48
Le nombre 48 admet 11 décompositions comme somme de deux impairs :

- les décompositions $5+43$, $7+41$, $11+37$, $17+41$, $19+29$, faisant intervenir deux nombres premiers sont codées par 5 déplacements à droite ;

1. On omet la décomposition $1 + (n - 1)$.
2. la lettre a de notes récentes.
3. la lettre d de notes récentes.
4. la lettre b de notes récentes.
5. la lettre c de notes récentes.

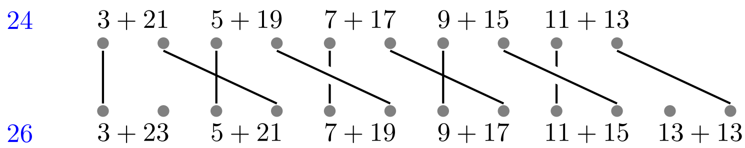
- les décompositions $3 + 45$, $13 + 35$, $23 + 25$ faisant intervenir un nombre premier et un impair composé sont codées par 3 déplacements vers le bas ;
- les décompositions $9 + 39$, $15 + 43$, $21 + 27$, faisant intervenir un impair composé et un nombre premier sont codées par 3 déplacements vers le haut.

On s'est déplacé du point origine $(0, 0)$ au point $(2, -3)$.

En annexe, sont fournis les déplacements associés aux nombres pairs de 10 à 100.

On constate que ce codage permet de trouver très aisément les pairs doubles de nombres premiers : leur "déplacement global" consiste en un unique déplacement vers le bas ou bien en un unique déplacement vers la droite.

Cela se comprend par la théorie des tresses : le passage d'un double de composé à un double de premier consistant à appliquer des permutations suivant toujours le même schéma, du fait de l'ajout artificiel du nombre premier en question dans la décomposition triviale $2p = p + p$; présentons l'exemple des nombres pairs 24 et 26 ci-dessous :



Ce codage permet d'autre part de trouver très aisément les pairs doubles de ces nombres qu'on désigne sous le nom de "pères de jumeaux" (i.e. un père de jumeau est un nombre pair k tel que $k - 1$ et $k + 1$ sont simultanément premiers) : leur "déplacement global" consiste en un unique déplacement à la fois vers le bas et vers la gauche (i.e. $(-1, -1)$, cf. annexe 3).

La proposition ci-dessus est codée par le programme suivant ; son résultat est fourni en annexe.

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

int prime(int p) {
  bool notfound=true ;
  unsigned long k = 2 ;

  if (p == 1) return 0;
  if (p == 2) return 1;
  if (p == 3) return 1;
  if (p == 5) return 1;
  if (p == 7) return 1;
  while (notfound) {
    if ((k * k) > p) return 1;
    else if ((p % k) == 0) { return 0 ; }
         else k++;
  }
}

int main (int argc, char* argv[]) {
  int n, k, x, y, xprec, yprec ;

  x = 0 ;
  y = 0 ;
  for (n=14 ; n <= 1000 ; n=n+2) {
    xprec = x ;
    yprec = y ;
    x = 0 ;
    y = 0 ;
    for (k=3 ; k <= n/2 ; k=k+2) {
      if (prime(k) && prime(n-k)) x=x+1 ;
      else if (prime(k) && (not(prime(n-k)))) y=y-1 ;
      else if ((not(prime(k))) && prime(n-k)) y=y+1 ;
      else if ((not(prime(k))) && (not(prime(n-k)))) x=x-1 ;
    }
    std::cout << n << " : x = " << x << " y = " << y << "\n" ;
    if ((((x-xprec) == 1) && ((y-yprec) == 0))
        || (((x-xprec) == 0) && ((y-yprec) == -1)))
      std::cout << "je descends ou vais à droite pour " << n/2 << "\n" ;
  }
}
```

Annexe 1 : déplacements associés aux nombres pairs de 10 à 100

10	$2(1,0)+0(-1,0)+0(0,1)+1(0,-1)$	$(2,0)$
12	$1(1,0)+0(-1,0)+0(0,1)+1(0,-1)$	$(1,-1)$
14	$2(1,0)+0(-1,0)+0(0,1)+1(0,-1)$	$(2,-1)$
16	$2(1,0)+0(-1,0)+0(0,1)+1(0,-1)$	$(2,-1)$
18	$2(1,0)+1(-1,0)+0(0,1)+1(0,-1)$	$(1,-1)$
20	$2(1,0)+0(-1,0)+1(0,1)+1(0,-1)$	$(2,0)$
22	$3(1,0)+0(-1,0)+1(0,1)+1(0,-1)$	$(3,0)$
24	$3(1,0)+1(-1,0)+0(0,1)+1(0,-1)$	$(2,-1)$
26	$3(1,0)+0(-1,0)+1(0,1)+2(0,-1)$	$(3,-1)$
28	$2(1,0)+0(-1,0)+1(0,1)+3(0,-1)$	$(2,-2)$
30	$3(1,0)+2(-1,0)+0(0,1)+2(0,-1)$	$(1,-2)$
32	$2(1,0)+0(-1,0)+2(0,1)+3(0,-1)$	$(2,-1)$
34	$4(1,0)+1(-1,0)+1(0,1)+2(0,-1)$	$(3,-1)$
36	$4(1,0)+2(-1,0)+0(0,1)+2(0,-1)$	$(2,-2)$
38	$2(1,0)+0(-1,0)+2(0,1)+5(0,-1)$	$(2,-3)$
40	$3(1,0)+1(-1,0)+1(0,1)+4(0,-1)$	$(2,-3)$
42	$4(1,0)+3(-1,0)+0(0,1)+3(0,-1)$	$(1,-3)$
44	$3(1,0)+1(-1,0)+2(0,1)+4(0,-1)$	$(2,-2)$
46	$4(1,0)+1(-1,0)+2(0,1)+4(0,-1)$	$(3,-2)$
48	$5(1,0)+3(-1,0)+0(0,1)+3(0,-1)$	$(2,-3)$
50	$4(1,0)+2(-1,0)+2(0,1)+4(0,-1)$	$(2,-2)$
52	$3(1,0)+1(-1,0)+3(0,1)+5(0,-1)$	$(2,-2)$
54	$5(1,0)+4(-1,0)+1(0,1)+3(0,-1)$	$(1,-2)$
56	$3(1,0)+1(-1,0)+4(0,1)+5(0,-1)$	$(2,-1)$
58	$4(1,0)+2(-1,0)+3(0,1)+5(0,-1)$	$(2,-2)$
60	$6(1,0)+5(-1,0)+0(0,1)+3(0,-1)$	$(1,-3)$
62	$3(1,0)+1(-1,0)+4(0,1)+7(0,-1)$	$(2,-3)$
64	$5(1,0)+3(-1,0)+2(0,1)+5(0,-1)$	$(2,-3)$
66	$6(1,0)+5(-1,0)+1(0,1)+4(0,-1)$	$(1,-3)$
68	$2(1,0)+1(-1,0)+5(0,1)+8(0,-1)$	$(1,-3)$
70	$5(1,0)+4(-1,0)+3(0,1)+5(0,-1)$	$(1,-2)$
72	$6(1,0)+5(-1,0)+2(0,1)+4(0,-1)$	$(1,-2)$
74	$5(1,0)+3(-1,0)+4(0,1)+6(0,-1)$	$(2,-2)$
76	$5(1,0)+3(-1,0)+4(0,1)+6(0,-1)$	$(2,-2)$
78	$7(1,0)+6(-1,0)+2(0,1)+4(0,-1)$	$(1,-2)$
80	$4(1,0)+3(-1,0)+5(0,1)+7(0,-1)$	$(1,-2)$
82	$5(1,0)+3(-1,0)+5(0,1)+7(0,-1)$	$(2,-2)$
84	$8(1,0)+7(-1,0)+1(0,1)+4(0,-1)$	$(1,-3)$
86	$5(1,0)+3(-1,0)+5(0,1)+8(0,-1)$	$(2,-3)$
88	$4(1,0)+3(-1,0)+5(0,1)+9(0,-1)$	$(1,-4)$
90	$9(1,0)+9(-1,0)+0(0,1)+4(0,-1)$	$(0,-4)$
92	$4(1,0)+3(-1,0)+6(0,1)+9(0,-1)$	$(1,-3)$
94	$5(1,0)+4(-1,0)+5(0,1)+9(0,-1)$	$(1,-4)$
96	$7(1,0)+7(-1,0)+2(0,1)+7(0,-1)$	$(0,-5)$
98	$3(1,0)+4(-1,0)+6(0,1)+11(0,-1)$	$(-1,-5)$
100	$6(1,0)+6(-1,0)+4(0,1)+8(0,-1)$	$(0,-4)$

Annexe 2 : déplacements des doubles de premiers ($(1,0)$ ou $(0,-1)$)

```
14 : x = 2 y = -1
16 : x = 2 y = -1
18 : x = 1 y = -1
20 : x = 2 y = 0
22 : x = 3 y = 0
je descends ou je vais à droite pour 11
24 : x = 2 y = -1
26 : x = 3 y = -1
je descends ou je vais à droite pour 13
28 : x = 2 y = -2
30 : x = 1 y = -2
32 : x = 2 y = -1
34 : x = 3 y = -1
je descends ou je vais à droite pour 17
36 : x = 2 y = -2
38 : x = 2 y = -3
je descends ou je vais à droite pour 19
40 : x = 2 y = -3
42 : x = 1 y = -3
44 : x = 2 y = -2
46 : x = 3 y = -2
je descends ou je vais à droite pour 23
48 : x = 2 y = -3
50 : x = 2 y = -2
52 : x = 2 y = -2
54 : x = 1 y = -2
56 : x = 2 y = -1
58 : x = 2 y = -2
je descends ou je vais à droite pour 29
60 : x = 1 y = -3
62 : x = 2 y = -3
je descends ou je vais à droite pour 31
64 : x = 2 y = -3
66 : x = 1 y = -3
68 : x = 1 y = -3
70 : x = 1 y = -2
72 : x = 1 y = -2
74 : x = 2 y = -2
je descends ou je vais à droite pour 37
76 : x = 2 y = -2
78 : x = 1 y = -2
80 : x = 1 y = -2
82 : x = 2 y = -2
je descends ou je vais à droite pour 41
84 : x = 1 y = -3
86 : x = 2 y = -3
je descends ou je vais à droite pour 43
88 : x = 1 y = -4
90 : x = 0 y = -4
92 : x = 1 y = -3
94 : x = 1 y = -4
je descends ou je vais à droite pour 47
96 : x = 0 y = -5
98 : x = -1 y = -5
100 : x = 0 y = -4
```

Annexe 3 : déplacements des doubles de “pères de jumeau” $((-1,-1))$

```
24 : x = 2 y = -1
24 : xprec = 3 yprec = 0

36 : x = 2 y = -2
36 : xprec = 3 yprec = -1

60 : x = 1 y = -3
60 : xprec = 2 yprec = -2

84 : x = 1 y = -3
84 : xprec = 2 yprec = -2

120 : x = 0 y = -3
120 : xprec = 1 yprec = -2

144 : x = -2 y = -5
144 : xprec = -1 yprec = -4

204 : x = -5 y = -5
204 : xprec = -4 yprec = -4

216 : x = -7 y = -8
216 : xprec = -6 yprec = -7

276 : x = -11 y = -7
276 : xprec = -10 yprec = -6

300 : x = -13 y = -7
300 : xprec = -12 yprec = -6
```