

Aligner dans le plan complexe les images des nombres premiers par une fonction assez bizarre (Denise Vella-Chemla, 2/10/2022)

On revient sur une fonction définie sur les entiers et qui semble aligner les images des nombres premiers et d'eux seuls sur une droite du plan complexe.

Présentons le programme python calculant cette fonction sur la copie d'écran ci-dessous. Les nombres premiers ont pour images par cette fonction des nombres complexes de la forme $a + \frac{1}{2}ai$ tandis que les images des nombres composés semblent être systématiquement extérieures à cette droite.

The image shows two Emacs windows and a terminal window. The left Emacs window contains the following Python code:

```
import mpmath
from mpmath import *

for n in range(6,101):
    print(n, '--> ',end='')
    sommetotale = 0
    for k in range(2,n):
        somme = 0
        for l in range(1,k+1):
            somme += (exp(2*pi*i*j*n*l/k))-1+(((-1)**(n/2))*0.5)
        sommetotale += somme
    print(sommetotale)
```

The right Emacs window shows the output of the program, listing complex numbers for each integer n from 6 to 55. The output is as follows:

```
6 --> (-16.0 - 6.79276928257154e-15j)
7 --> (-20.0 - 10.0j)
8 --> (-7.5 - 9.01414088133518e-15j)
9 --> (-32.0 + 17.5j)
10 --> (-59.0 - 2.70464489395849e-14j)
11 --> (-54.0000000000001 - 27.0j)
12 --> (-17.5 - 3.46503890599909e-14j)
13 --> (-77.0 + 38.500000000001j)
14 --> (-126.0 - 5.89291709804086e-14j)
15 --> (-96.0000000000002 - 52.0000000000002j)
16 --> (-45.5 - 1.39124039803985e-14j)
17 --> (-135.0 + 67.5j)
18 --> (-208.0 - 5.39942318179067e-14j)
19 --> (-170.0 - 85.000000000001j)
20 --> (-73.5 - 1.44960302051633e-13j)
21 --> (-199.0 + 104.5j)
22 --> (-332.0 - 3.37243583173453e-13j)
23 --> (-252.0 + 126.0j)
24 --> (-102.5 - 1.54942081485548e-13j)
25 --> (-294.0 + 149.5j)
26 --> (-471.0 + 2.8876573500039e-13j)
27 --> (-338.000000000001 - 175.0j)
28 --> (-161.5 - 9.04052542853554e-14j)
29 --> (-405.0 + 202.5j)
30 --> (-610.0 - 8.92906677758838e-13j)
31 --> (-464.000000000001 - 232.0j)
32 --> (-217.5 - 1.72931836085753e-13j)
33 --> (-513.0 + 263.5j)
34 --> (-821.0 + 1.05906010144323e-13j)
35 --> (-582.000000000001 - 297.0j)
36 --> (-260.5 - 5.17476442946005e-13j)
37 --> (-665.0 + 332.5j)
38 --> (-1032.0 - 5.56996460722995e-13j)
39 --> (-724.000000000002 - 370.000000000001j)
40 --> (-340.5 - 6.17751517903515e-13j)
41 --> (-818.999999999997 + 409.499999999999j)
42 --> (-1237.0 - 2.55435243315038e-13j)
43 --> (-902.000000000002 - 451.0j)
44 --> (-433.500000000001 - 1.82834104665499e-12j)
45 --> (-956.999999999999 + 494.5j)
46 --> (-1526.0 - 6.6621738067597e-13j)
47 --> (-1080.0 - 540.0j)
48 --> (-488.5 - 7.66679570878079e-13j)
49 --> (-1168.0 + 587.499999999999j)
50 --> (-1794.0 - 1.45644152977536e-13j)
51 --> (-1254.0 - 637.000000000001j)
52 --> (-617.499999999999 + 1.86484946682103e-12j)
53 --> (-1377.0 + 688.499999999999j)
54 --> (-2080.0 - 1.28820243410469e-12j)
55 --> (-1468.0 - 741.999999999999j)
```

The terminal window shows the execution of the Python script:

```
denise@denise-Inspiron-3501:~/Bureau$ python3 prem-sur-droite-complexe-pente-zero-point-cinq.py
denise@denise-Inspiron-3501:~/Bureau$ python3 resultat-prem-sur-droite-complexe-pente-zero-point-cinq.py
denise@denise-Inspiron-3501:~/Bureau$
```

On avait étudié cette fonction après des travaux autour de la somme des diviseurs,

suite à la lecture d'un article d'Euler *Découverte d'une loi tout extraordinaire des nombres par rapport à la somme de leurs diviseurs.*

On peut calculer la somme des diviseurs des nombres comme une somme de cosinus.

Pour $n \geq 1$, on définit :

$$\sigma(n) = \sum_{k=1}^n \sum_{l=1}^k \cos\left(\frac{2i\pi nl}{k}\right)$$

Pour $1 \leq k \leq n$, on a :

$$\sigma_k(n) = \begin{cases} \sum_{l=1}^k e^{\frac{2i\pi nl}{k}} = k & \text{si } k \mid n \\ \sum_{l=1}^k e^{\frac{2i\pi nl}{k}} = \sum_{l=1}^k \left(e^{\frac{2i\pi n}{k}}\right)^l = \frac{\left(e^{\frac{2i\pi n}{k}}\right)^{k+1} - 1}{e^{\frac{2i\pi n}{k}} - 1} = e^{\frac{2i\pi n}{k}} \left(\frac{e^{2i\pi n} - 1}{e^{\frac{2i\pi n}{k}} - 1}\right) = 0 & \text{sinon} \end{cases}$$

On en déduit que $\sigma(n)$ est la somme des diviseurs de n :

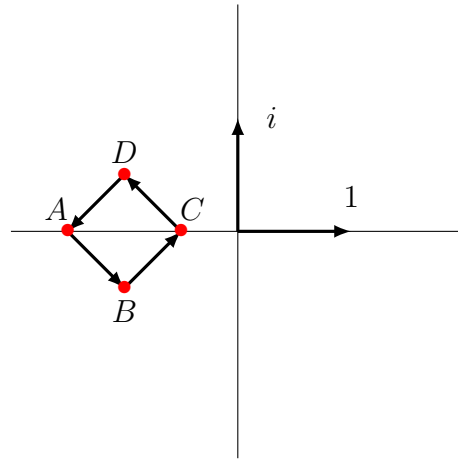
$$\sigma(n) = \sum_{k=1}^n \sigma_k(n) = \sum_{k \mid n} k$$

Fin 2018, on avait eu comme idée¹ de calculer une somme alternée des mêmes cosinus, mais en voulant simplifier cette formule et en commettant plusieurs erreurs, on aboutit à la définition suivante d'une fonction $f_D(n)$ définie sur les entiers (pour $n \geq 1$) :

$$f_D(n) = \left[\sum_{k=2}^{n-1} \sum_{l=1}^k \cos \frac{2\pi nl}{k} \right] + \left[-1 + \frac{1}{2} \left((-1)^{\frac{n}{2}} \right) \right]$$

L'ajout du terme bizarre $\left[-1 + \frac{1}{2} \left((-1)^{\frac{n}{2}} \right) \right]$ en fin de formule consiste à ajouter à la somme des diviseurs l'un des 4 complexes qui sont les affixes des sommets du carré ($A = -1.5$, $B = -1 - 0.5i$, $C = -0.5$, $D = -1 + 0.5i$) (*remarque* : on a soustrait de la somme des diviseurs les deux diviseurs triviaux que sont 1 et n : pour effectuer cette soustraction la somme englobante d'indice k est prise pour k variant de 2 à $n - 1$).

¹Voir <http://denise.vella.chemla.free.fr/premiers-image-un-demi.pdf>.



On ne sait pas pour l'instant pourquoi la fonction $f_D(n)$ envoie les nombres premiers sur la droite complexe $a + \frac{1}{2}ai$.