High resolution graphs using Maple

By Simon Plouffe

Introduction

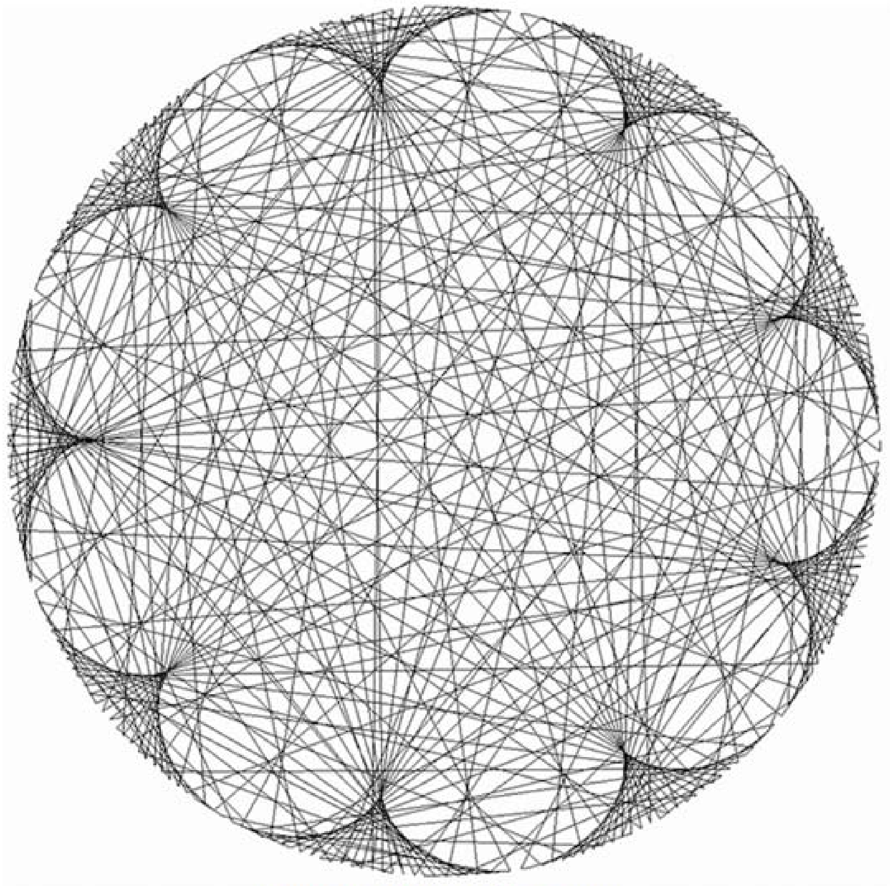
The idea of using a way to represent the variations of a function. Typically this function will go from

$$f(x) \in R \to [0,1]$$

What comes next naturally is to wrap the values from [0,1] to the unit circle, so we will have

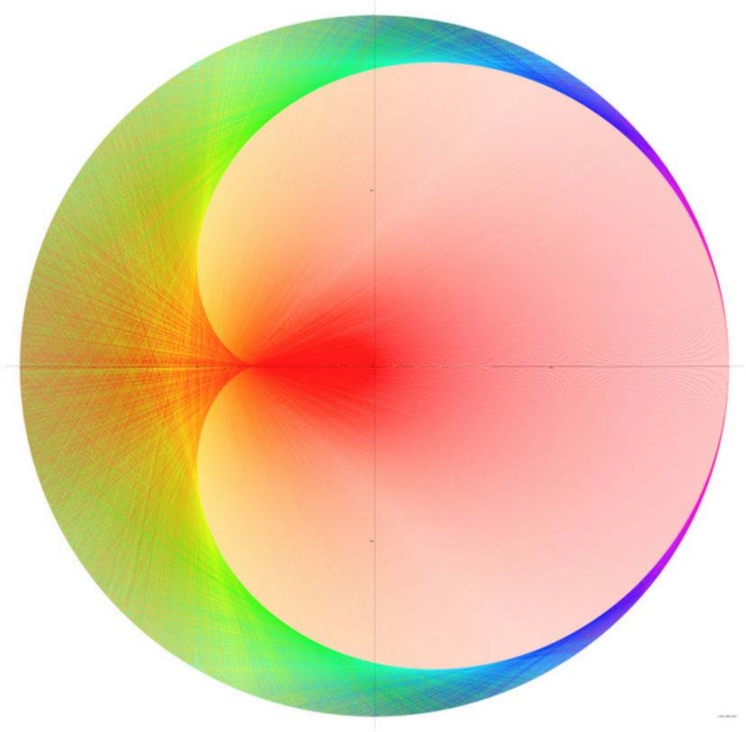
$$f(x) \in R \to [0,1] \to e\hat{}2\pi ix$$

But originally, these graphs were intended to explain the development of rational numbers in base 10 like the number 1/257=0.003891050583... the simple idea is then to represent the development by moving the decimal point to the right. This is the same as joining 2 adjacent points. So by joining 2 successive points we obtain the following graph.



Representation of 1/257 in base 10

The use of the base 10 explains the 9 peaks but what about the other 'peaks'? There are 23 in this case. The first drawing was made on a wall by hand on a height of 1m50Therefore, if we use the base 2 we have 1 point, it is the cardioid.



Expansion of 1/10037 in base 2.

These are the successive lines obtained from the calculation of

$$\frac{2^n \; mod \; 10037}{10037} = \frac{1}{10037}, \frac{2}{10037}, \frac{4}{10037}, \frac{8}{10037}, \dots$$

Which when rolled up on the unit circle gives the points

$$e^{\frac{2I}{10037}\pi}, e^{\frac{4I}{10037}\pi}, e^{\frac{8I}{10037}\pi}, e^{\frac{16I}{10037}\pi}, e^{\frac{32I}{10037}\pi}, e^{\frac{64I}{10037}\pi}, e^{\frac{128I}{10037}\pi}$$

The color is added using the rule: color = length of each segment.

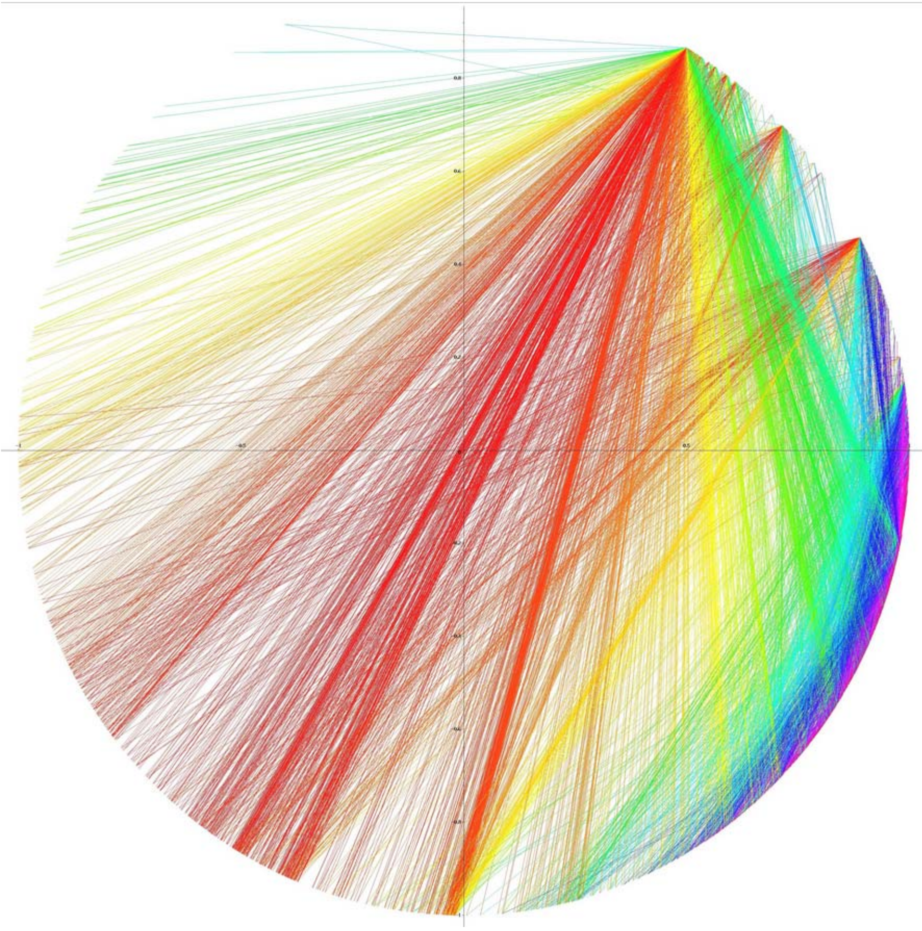
Translated into Maple

```
graphc := proc(s, nom)
local nr, pr, ligne, n, i, g, v, liste, j, t1, fichier,
nn, nj, x1, x2, y1,
y2, p1, z, m, a1;
    v := gg(s);
    nn := nops(v);
    nr := convert(nom, string);
    fichier := cat(nr, `.jpg`);
    interface(plotoutput = fichier);
    liste := [];
    for j to nn do
        x1 := cos(2*pp*v[j][1]);
        y1 := sin(2*pp*v[j][1]);
        x2 := cos(2*pp*v[j][2]);
        y2 := sin(2*pp*v[j][2]);
        ligne[j] := line([x1, y1], [x2, y2],
            color = COLOR(HUE, 1 - 0.5*sqrt((x1 - x2)^2 +
(y1 - y2)^2)));
        liste := [op(liste), ligne[j]]
    end do;
    t1 :=
        plots[textplot]([1, -1, typeset(nr, "   "), font
= [COURIER, 24]]);
    display(op(liste), t1)
end proc
```

Before running it is appropriate to prepare the interfaces and the output.

```
printlevel:=5:
interface(plotdevice=jpeg):
interface(plotoptions="height=8192,width=8192"):
Digits:=16:
with(plottools):
with(plots):
pp:=evalf(Pi):
```

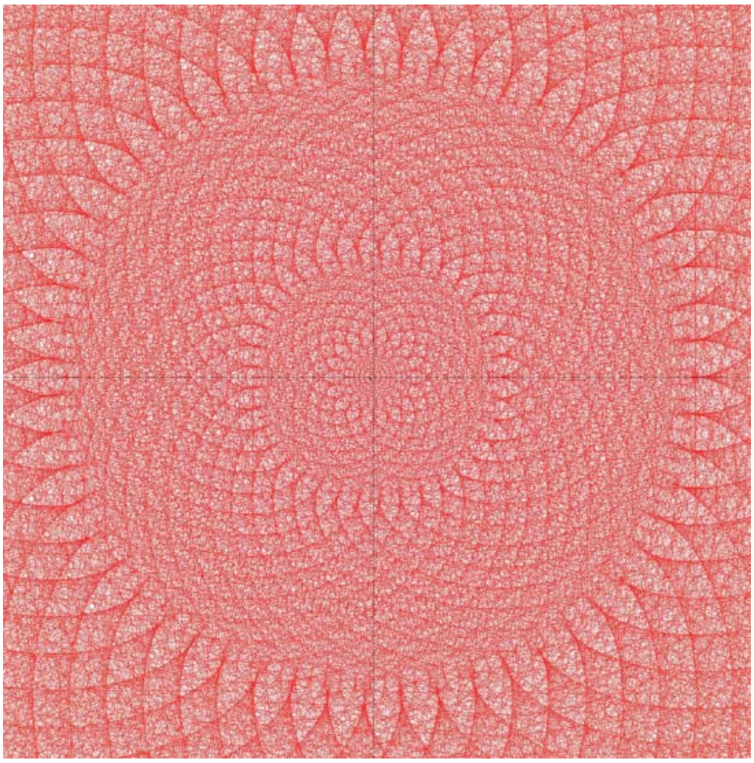
From this point on I started to experiment with everything that is known as functions between 0 and 1. Like for example, the distribution of fractional parts of Bernoulli numbers.



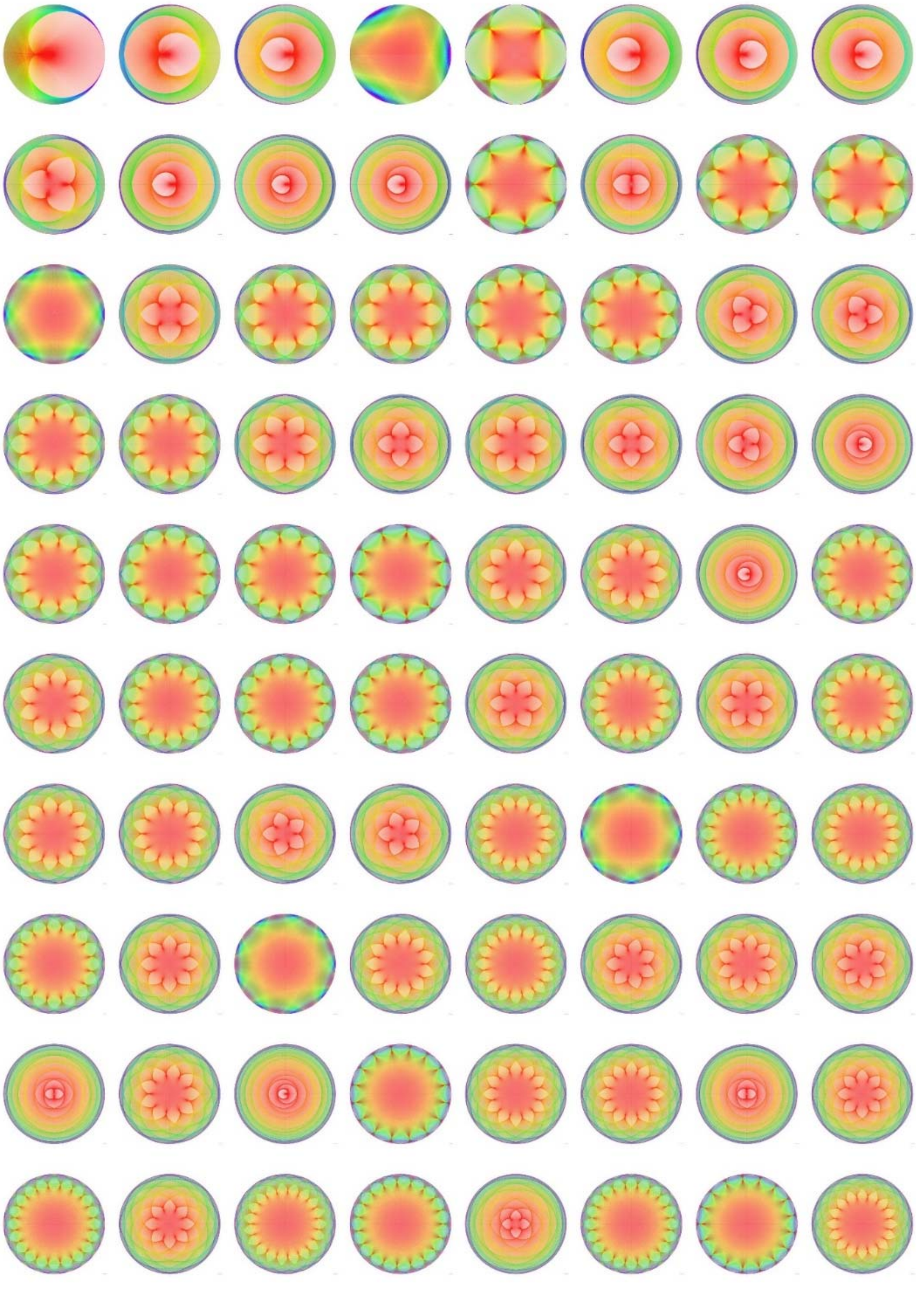
Distribution of values of {B_2k } , and { } is the fractional part.

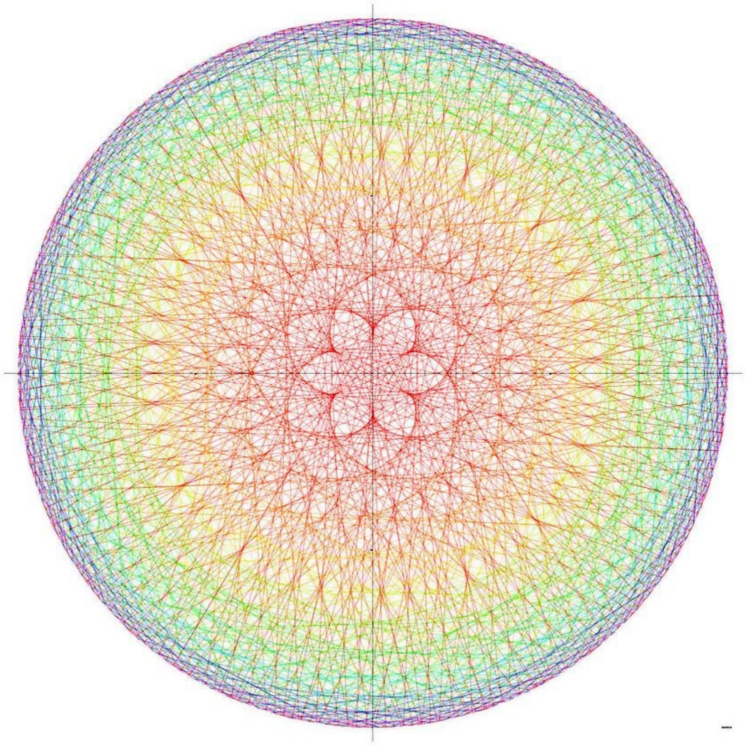
To come back to the graphs seen above, I explored all the ways to generate them and even to enlarge in the center (10 x zoom) to see them more clearly.

Here are some examples of inverses of primes in base 60 and 240.

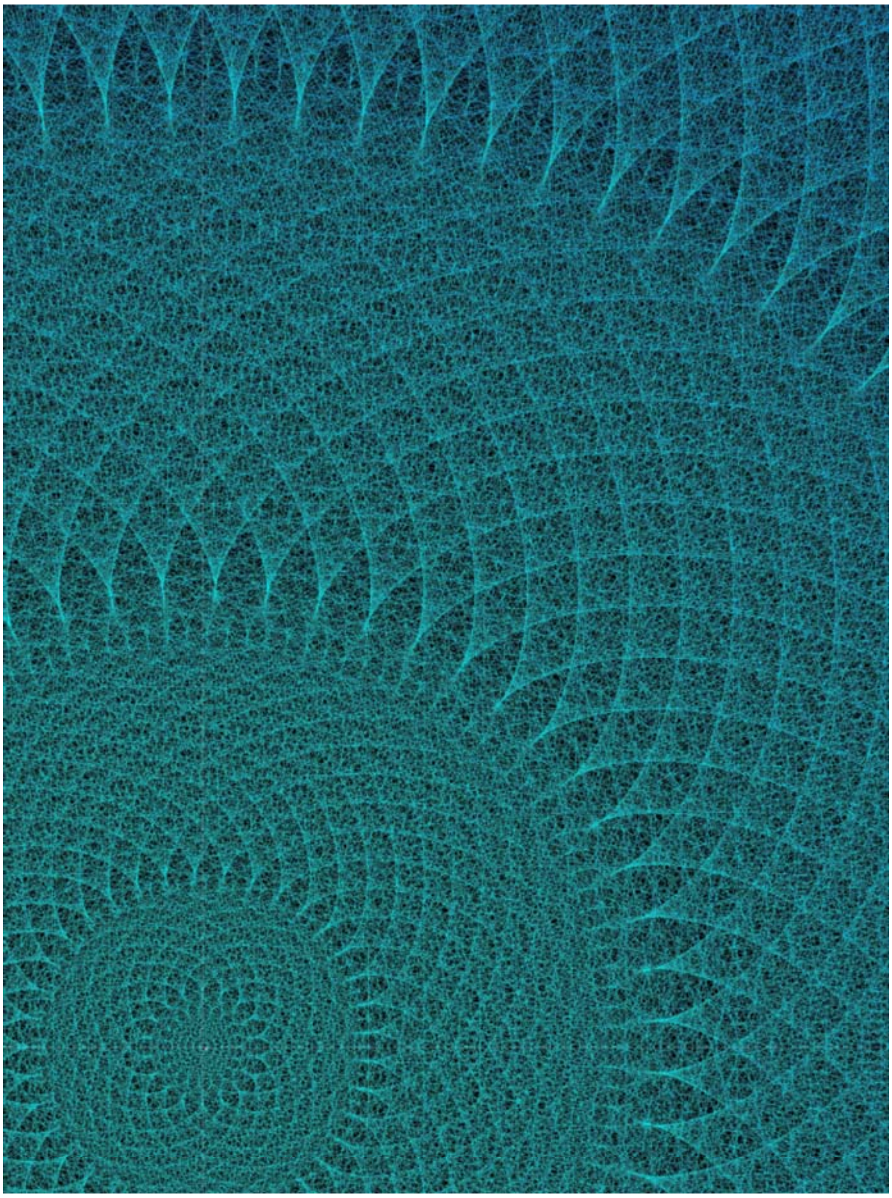


With $240^n \ mod \ 14009$, we obtain $P_0 = 92$ and $P_1 = 239$ the center of the huge 1 billion pixels gives 18 and 19 spikes.



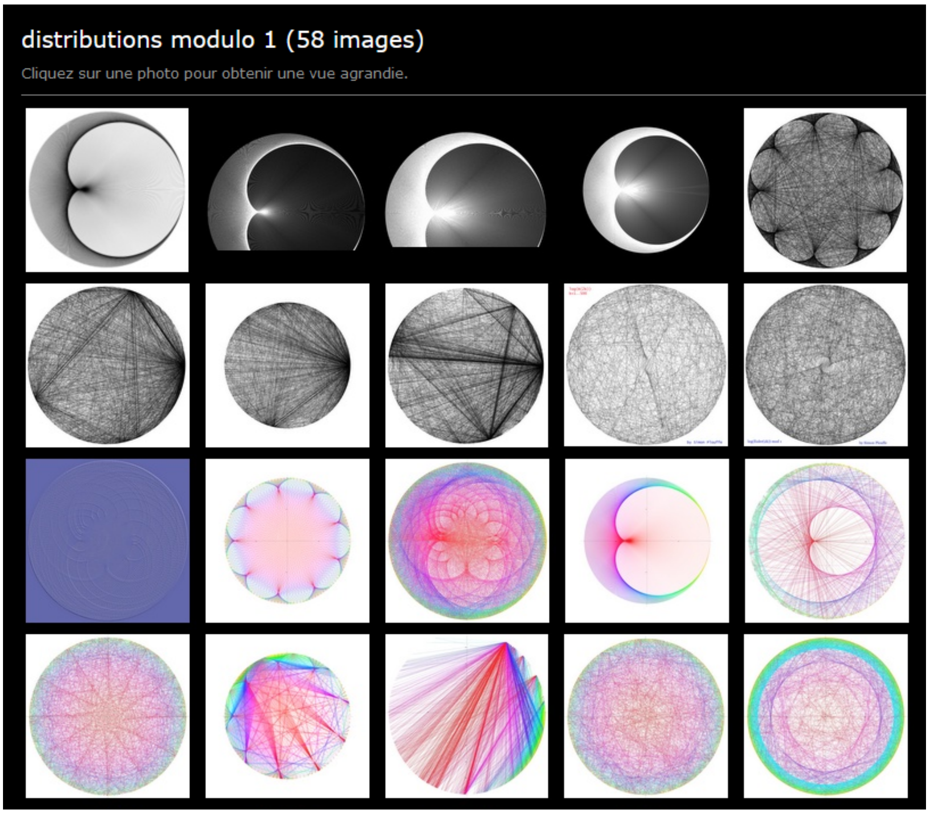


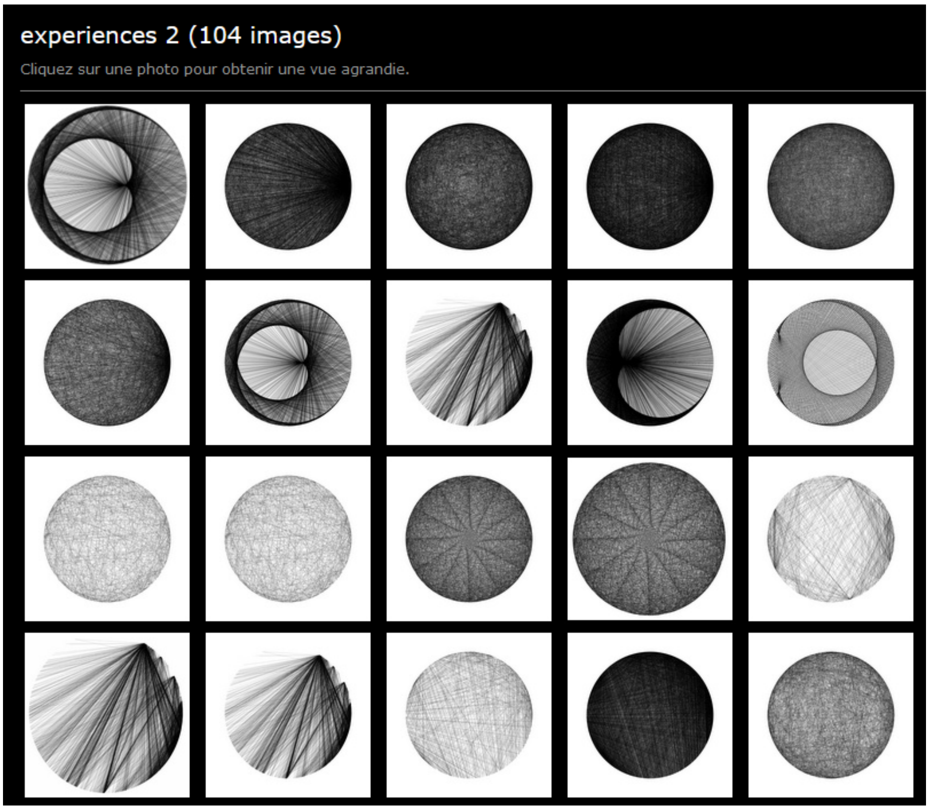
Simon Plouffe 2020: base = 240 , prime = 991, P1 = 204, harmonics = 6, 12, 17, 18, 23, 29, 35, 41, 47,



Near the center of 240^n mod 26437 in inverted color for more visibility
$P_0 = 239, P_1 = 92$
The number of spikes are within the sequence of harmonics : 1, 17, 18, 19, 20, 35, 37, 54, 55, 56, 57, 72, 73, 74, 75, 91, 92

Other experiments :
http://plouffe.fr/distributions%20modulo%201/





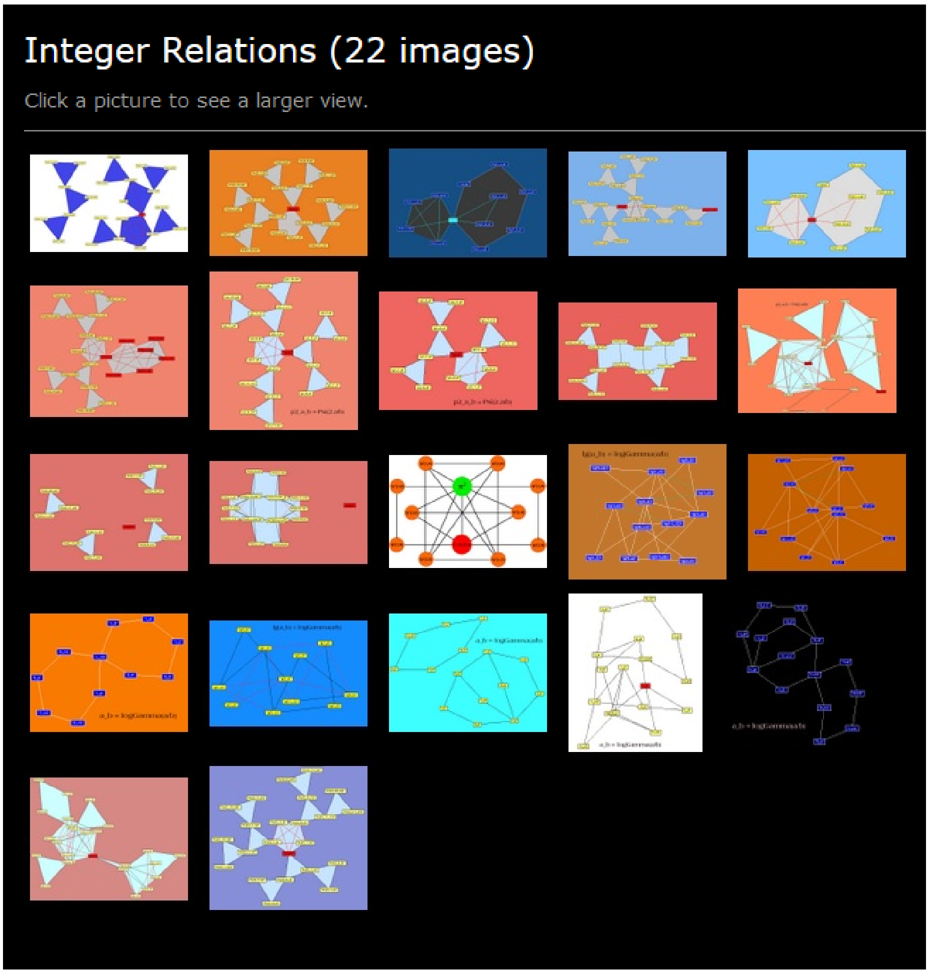
Pages at http://plouffe.fr/experiences%202/

Other sources of graphs :

http://plouffe.fr/Inverseofprimes/160/

http://plouffe.fr/Inverseofprimes/240/

http://plouffe.fr/Inverseofprimes/premiers%20base%20240C/



Images at http://plouffe.fr/simon/IntegerRelations/

Graphiques haute résolution faits avec Maple

par Simon Plouffe

Introduction

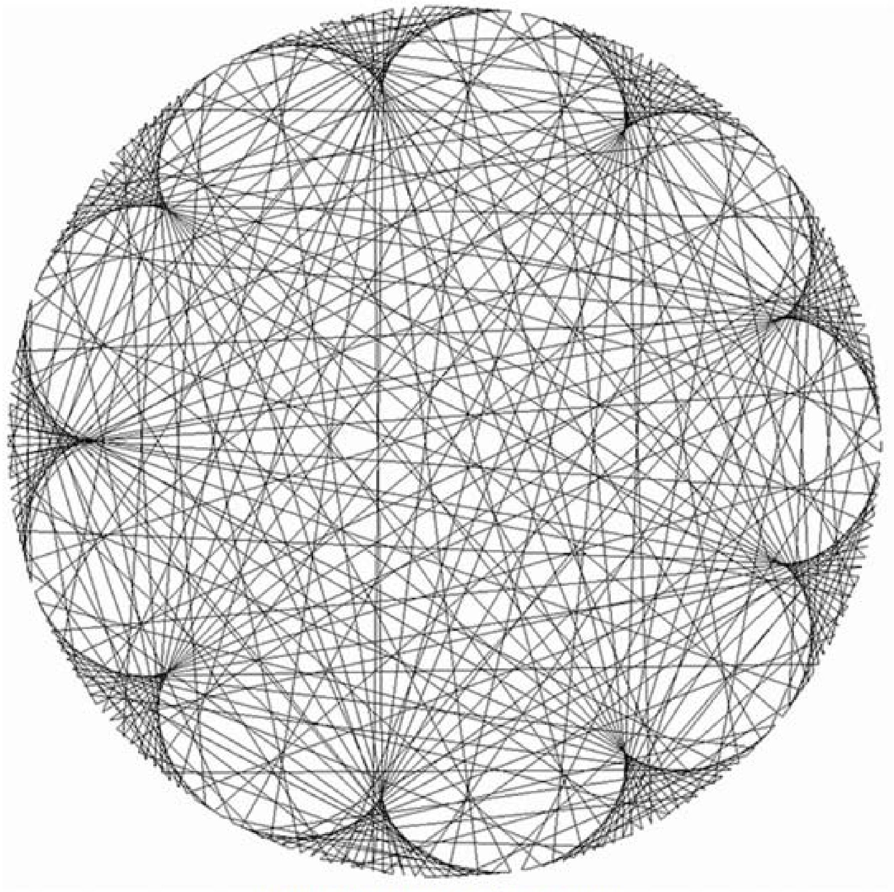
L'idée d'utiliser un moyen de représenter les variations d'une fonction. Typiquement cette fonction ira de

$$f(x) \in \mathbb{R} \to [0,1]$$

Ce qui vient ensuite naturellement est d'envelopper les valeurs de $[0,1]$ vers le cercle unité, donc on aura

$$f(x) \in \mathbb{R} \to [0,1] \to e^{2\pi i x}$$

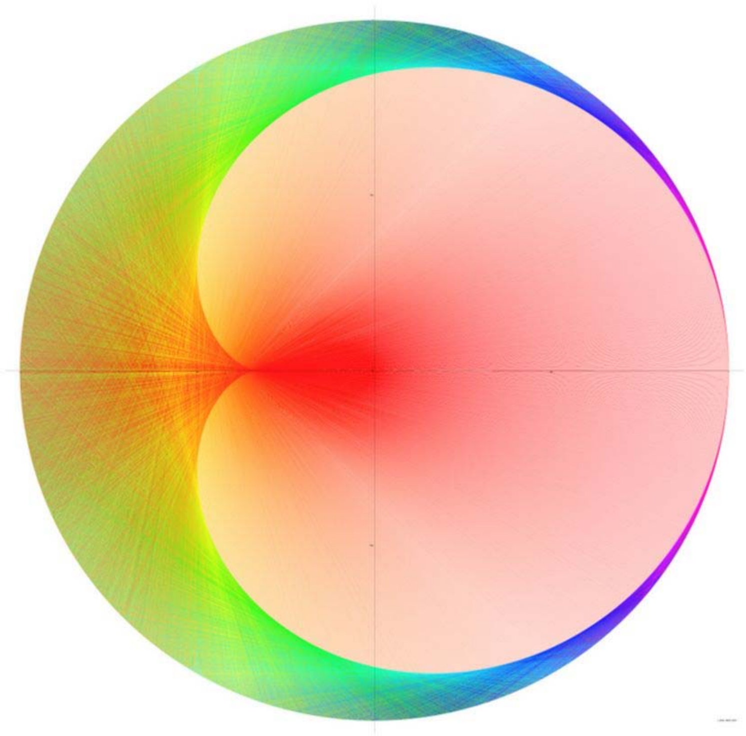
Mais au départ, ces graphes étaient destinés à expliquer le développement de nombres rationnels en base 10 comme le nombre $\frac{1}{257} = 0{,}003891050583\ldots$ l'idée simple est alors de représenter le développement en déplaçant le point décimal à droite. Ce qui revient à joindre 2 points adjacents. Donc en joignant 2 points successifs on obtient le graphe suivant.



Representation of 1/257 in base 10

L'emploi de la base 10 explique les 9 pointes mais qu'en est-il des autres 'pointes' ? On en compte 23 dans ce cas. Le premier dessin a été fait sur un mur à la main sur une hauteur de 1m50.

Par conséquent, si on utilise la base 2 on a 1 pointe, c'est la cardioïde.



Développement de 1/10037 en base 2.

Ce sont donc les droites successives obtenues à partir du calcul de

$$\frac{2^n \, mod \, 10037}{10037} = \frac{1}{10037}, \frac{2}{10037}, \frac{4}{10037}, \frac{8}{10037}, \ldots$$

Qui une fois enroulées sur le cercle unité donne les points

$$e^{\frac{2 I}{10037}\pi}, e^{\frac{4 I}{10037}\pi}, e^{\frac{8 I}{10037}\pi}, e^{\frac{16 I}{10037}\pi}, e^{\frac{32 I}{10037}\pi}, e^{\frac{64 I}{10037}\pi}, e^{\frac{128 I}{10037}\pi}$$

La couleur est ajoutée en utilisant la règle : couleur = longueur de chaque segment.

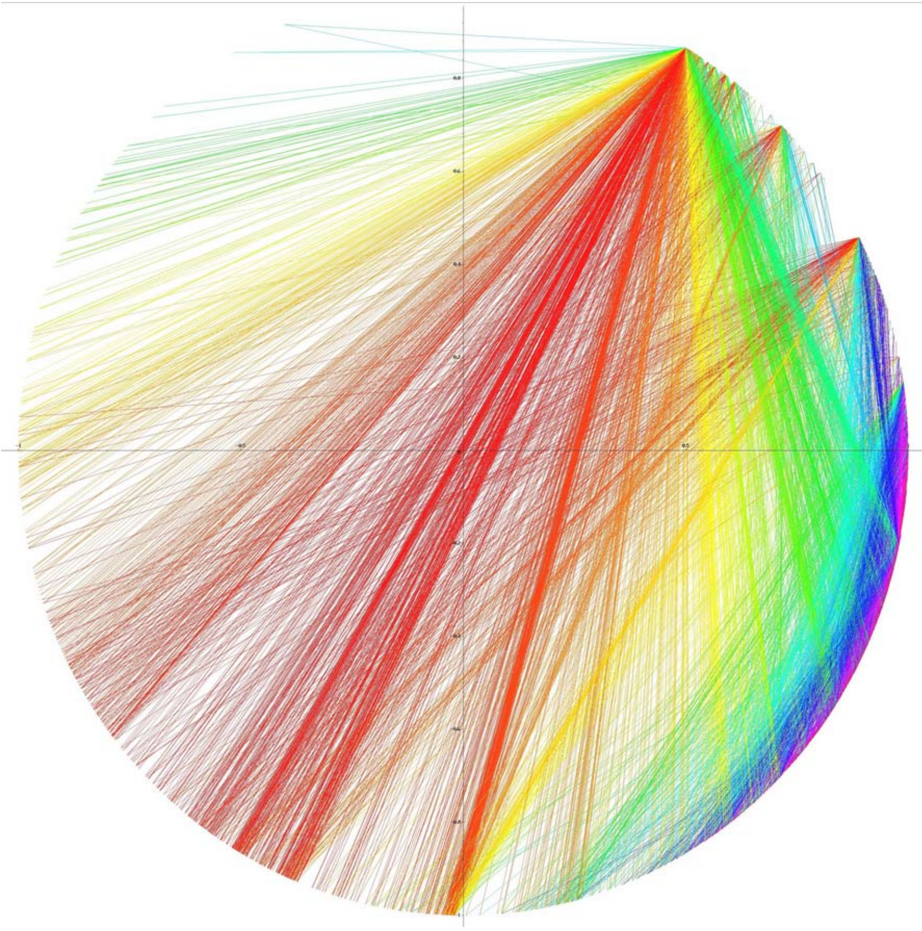
Traduit en Maple

```
graphc := proc(s, nom)
local nr, pr, ligne, n, i, g, v, liste, j, t1, fichier,
nn, nj, x1, x2, y1,
y2, p1, z, m, a1;
    v := gg(s);
    nn := nops(v);
    nr := convert(nom, string);
    fichier := cat(nr, `.jpg`);
    interface(plotoutput = fichier);
    liste := [];
    for j to nn do
        x1 := cos(2*pp*v[j][1]);
        y1 := sin(2*pp*v[j][1]);
        x2 := cos(2*pp*v[j][2]);
        y2 := sin(2*pp*v[j][2]);
        ligne[j] := line([x1, y1], [x2, y2],
            color = COLOR(HUE, 1 - 0.5*sqrt((x1 - x2)^2 +
(y1 - y2)^2)));
        liste := [op(liste), ligne[j]]
    end do;
    t1 :=
        plots[textplot]([1, -1, typeset(nr, "   "), font
= [COURIER, 24]]);
    display(op(liste), t1)
end proc
```

Avant d'exécuter il est approprié de préparer les interfaces et la sortie.

```
printlevel:=5:
interface(plotdevice=jpeg):
interface(plotoptions="height=8192,width=8192"):
Digits:=16:
with(plottools):
with(plots):
pp:=evalf(Pi):
```

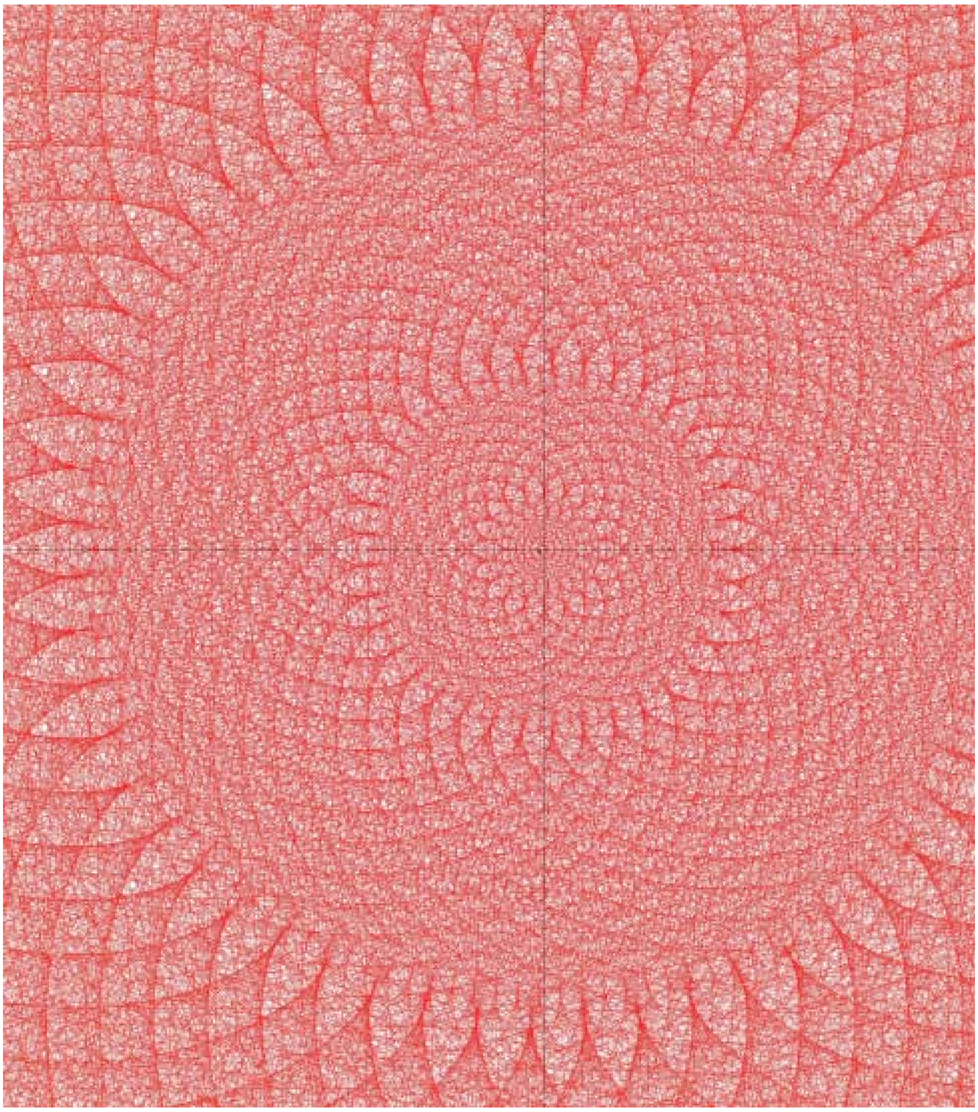
À partir de ce point j'ai commencé à expérimenter tout ce qui est connu comme fonctions entre 0 et 1. Comme par exemple, la distribution des parties fractionnaires des nombres de Bernoulli.



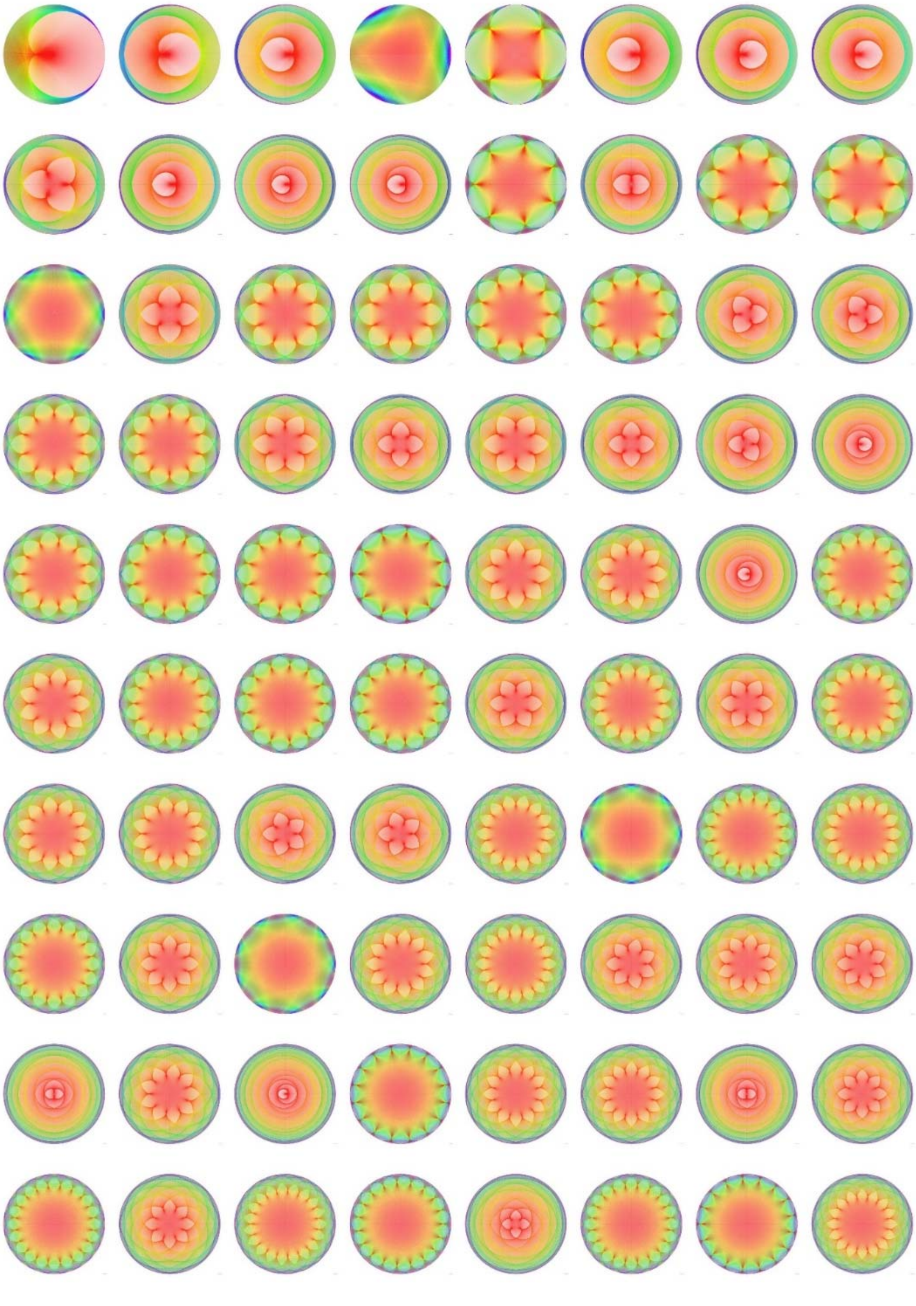
Distribution des valeurs de $\{B_{2k}\}$, et { } est la partie fractionnaire.

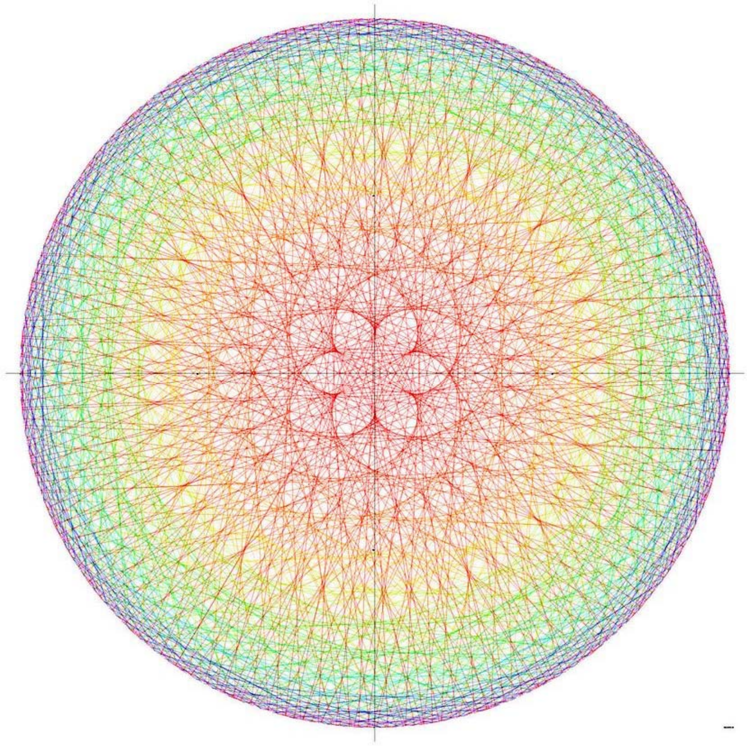
Pour revenir aux graphes vus plus haut, j'ai exploré toutes les façons de les générer et même d'agrandir au centre (zoom 10 x) pour y voir plus clair .

Voici quelques exemples d'inverses de premiers en base 60 et 240.

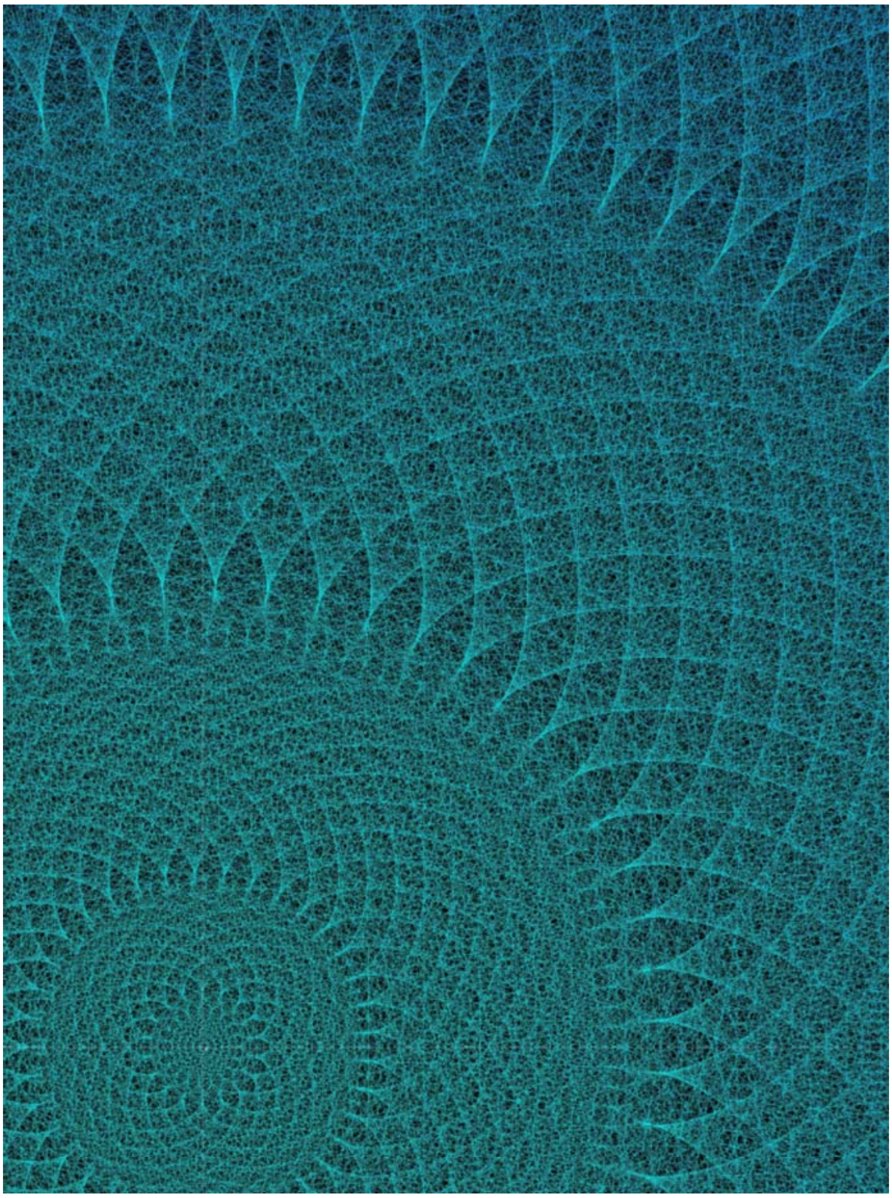


With $240^n \ mod \ 14009$, we obtain $P_0 = 92$ and $P_1 = 239$ the center of the huge 1 billion pixels gives 18 and 19 spikes.



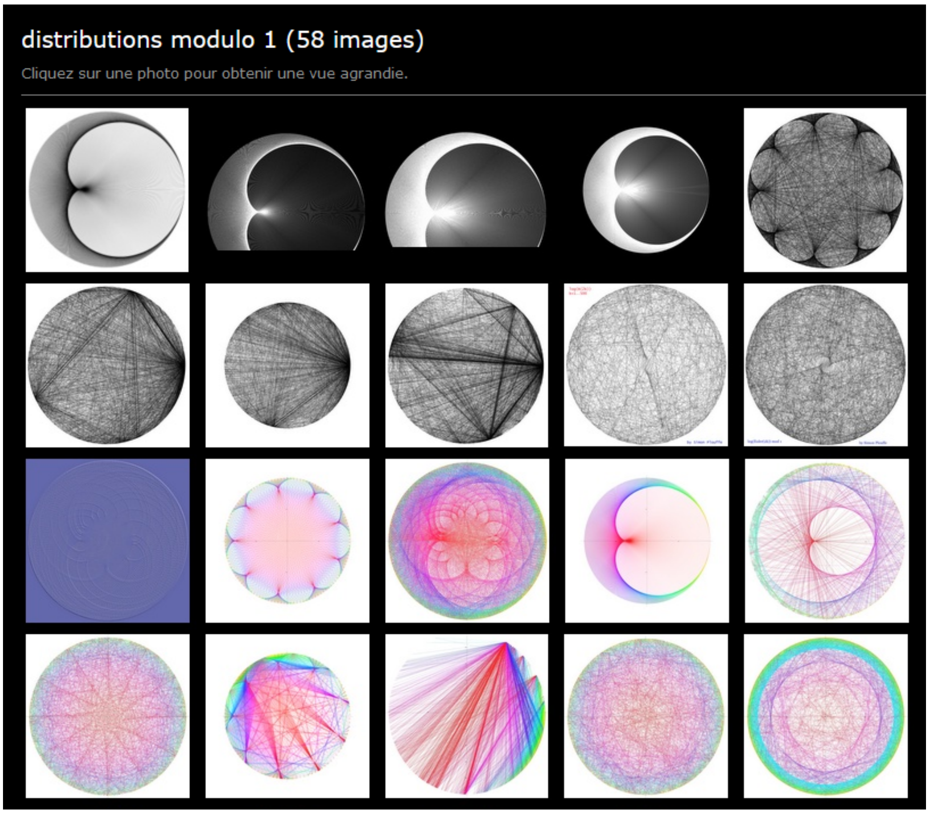


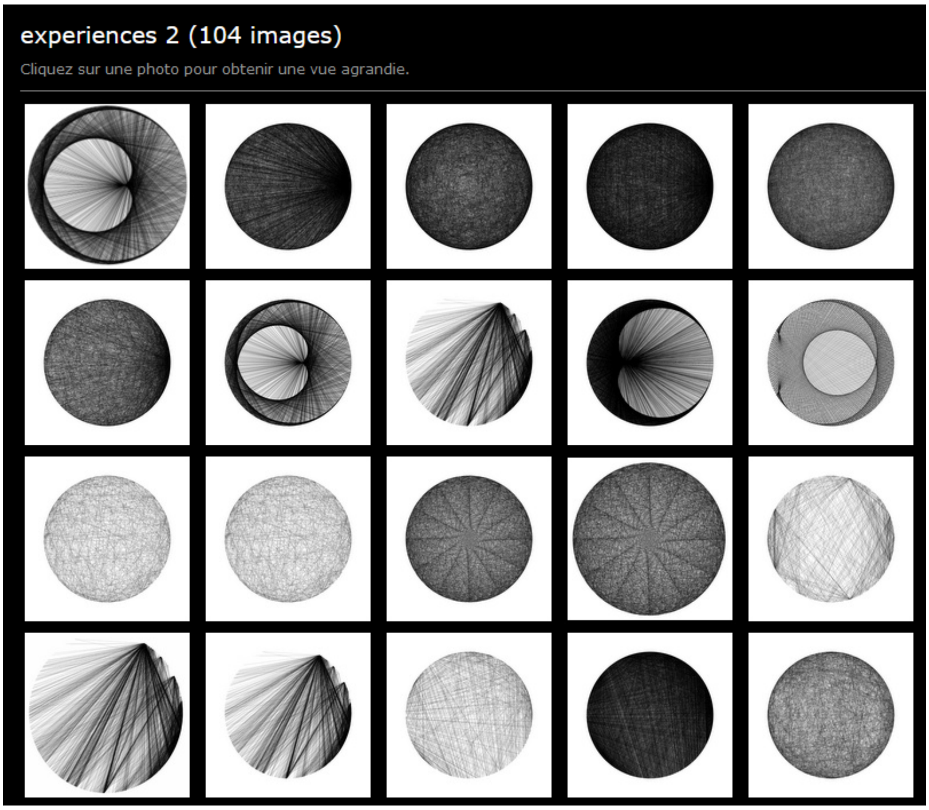
Simon Plouffe 2020: base = 240 , prime = 991, P1 = 204, harmonics = 6, 12, 17, 18, 23, 29, 35, 41, 47,



Near the center of 240^n mod 26437 in inverted color for more visibility
$P_0 = 239, P_1 = 92$
The number of spikes are within the sequence of harmonics : 1, 17, 18, 19, 20, 35, 37, 54, 55, 56, 57, 72, 73, 74, 75, 91, 92

Other experiments :
http://plouffe.fr/distributions%20modulo%201/





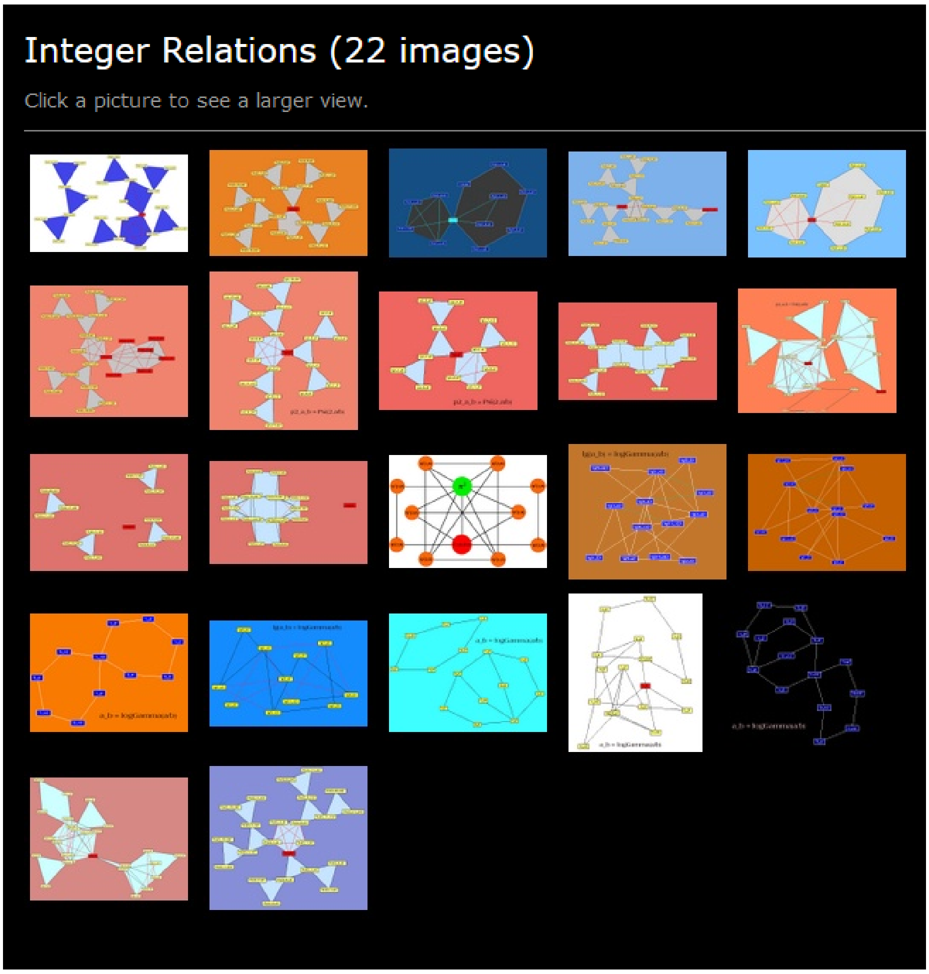
Pages at http://plouffe.fr/experiences%202/

Other sources of graphs :

http://plouffe.fr/Inverseofprimes/160/

http://plouffe.fr/Inverseofprimes/240/

http://plouffe.fr/Inverseofprimes/premiers%20base%20240C/



Images at http://plouffe.fr/simon/IntegerRelations/