BROUILLON - SOMMER LES CARRÉS DES CHIFFRES D'UN NATUREL

CHRISTOPHE BAL

Document, avec son source LaTeX, disponible sur la page
https://github.com/bc-writing/drafts.

Table des matières

1. Faire une tête au carré à tous les entiers naturels

Voici un procédé facile à faire à l'aide d'une calculatrice. Considérons un entier naturel n, puis calculons la somme de ses chiffres élevés au carré. Ceci nous donne un nouveau naturel auquel on peut appliquer le même procédé. Voici deux exemples.

Exemple 1.1. *Pour $n = 19$, nous obtenons :*

- $1^2 + 9^2 = 82$
- $8^2 + 2^2 = 68$
- $6^2 + 8^2 = 100$
- $1^2 + 0^2 + 0^2 = 1 \rightarrow$ *Rien de nouveau à attendre.*

Exemple 1.2. *Pour $n = 1\,234\,567\,890$, après $1^2+2^2+3^2+4^2+5^2+6^2+7^2+8^2+9^2+0^2 = 285$ nous obtenons :*

- $2^2 + 8^2 + 5^2 = 93$
- $9^2 + 3^2 = 90$
- $9^2 + 0^2 = 81$
- $8^2 + 1^2 = 65$
- $6^2 + 5^2 = 61$
- $6^2 + 1^2 = 37$
- $3^2 + 7^2 = 58$
- $5^2 + 8^2 = 89$
- $8^2 + 9^2 = 145$
- $1^2 + 4^2 + 5^2 = 42$
- $4^2 + 2^2 = 20$
- $2^2 + 0^2 = 4$
- $4^2 = 16$
- $1^2 + 6^2 = 37 \rightarrow$ *Dèjà rencontré.*

Dans le 1er cas, au bout d'un moment le procédé ne produit que des 1. Ce sera le cas dès que l'on commence avec une puissance de 10. Le 2e exemple montre que le mieux que l'on puisse espérer c'est que le procédé devienne périodique à partir d'un moment *(on parle de phénomène ultimement périodique).*

Date: 6 Juin 2018 – 10 Déc. 2018.

On peut explorer le comportement de ce procédé sur plusieurs valeurs grâce à un programme. Voici un code possible non optimisé[1] écrit en Python 3.7 qui prend un peu de temps pour vérifier que pour tous les naturels $n \in [\![1; 10^6]\!]$, le procédé devient ultimement périodique.

```
NMAX    = 10**6
MAXLOOP = 10**20

for n in range(1, NMAX + 1):
    nbloops = 0
    results = []

    while nbloops < MAXLOOP and n not in results:
        nbloops += 1
        results.append(n)
        n = sum(int(d)**2 for d in str(n))

    if n not in results:
        print(f"Test raté pour n = {n}.")

print("Tests finis.")
```

Une fois lancé, le code précédent affiche juste `Tests finis`. Il reste à voir ce qu'il se passe dans le cas général. La section qui suit démontre que pour tout naturel n, le procédé sera toujours ultimement périodique.

2. Une preuve

On introduit les notations suivantes.

- Pour un naturel n, $n = [\, c_{d-1}c_{d-2} \cdots c_1c_0 \,]_{10} \overset{\text{def}}{=} \sum_{k=0}^{d-1} c_k 10^k$, avec $c_{d-1} \neq 0$, désigne l'écriture décimale propre de n.
- On pose enuite $sq(n) = \sum_{k=0}^{d-1} (c_k)^2$ et $\texttt{taille}(n) = d$ sera appelé *« taille de n »*.
- Pour $(n\,;k) \in \mathbb{N}^2$, on définit $\boxed{n}_{\,0} = n$ et $\boxed{n}_{\,k} = sq^k(n) \overset{\text{def}}{=} sq \circ sq \circ \cdots \circ sq(n)$ avec $(k-1)$ compositions si $k > 0$.

  Autrement dit, nous avons $\boxed{n}_{\,0} = n$ et $\boxed{n}_{\,k+1} = sq\left(\boxed{n}_{\,k}\right)$.
- Enfin on note $V_n = \left\{ \boxed{n}_{\,k} \mid k \in \mathbb{N} \right\}$ l'ensemble des valeurs prises par la suite $\left(\boxed{n}_{\,k}\right)_k$.

Fait 2.1. *$\forall n \in \mathbb{N}$, $sq(n) \leqslant 81d$ où $d = \texttt{taille}(n)$.*

Preuve. *Si $n = [\, c_{d-1}c_{d-2} \cdots c_1c_0 \,]_{10}$ alors $sq(n) = \sum_{k=0}^{d-1} (c_k)^2 \leqslant \sum_{k=0}^{d-1} 9^2 = 81d$.*

1. Voir la section proposant des graphiques pour découvrir un code qui explore un peu plus efficacement les périodes.

Fait 2.2. *$\forall n \in \mathbb{N}$, notant $d = \texttt{taille}(n)$, nous avons les résultats suivants :*

(1) Si $d \geqslant 4$ alors $sq(n) < n$.

(2) Si $d \leqslant 3$ alors $sq(n) < 10^3$.

Preuve. *Comme $n \geqslant 10^{d-1}$ et compte tenu du fait précédent, nous cherchons à comparer 10^{d-1} et $81d$. Pour cela, nous allons considérer sur $\mathbb{R}_+$ la fonction $\Delta(x) = 10^{x-1} - 81x$. Comme $\Delta'(x) = \ln 10 \times 10^{x-1} - 81$, nous avons $\Delta'(x) > 0$ si et seulement si $x > 1 + \dfrac{1}{\ln 10} \ln\left(\dfrac{81}{\ln 10}\right)$ où le dernier nombre α vaut environ $2{,}546 < 3$.*

Comme $10^3 > 81 \times 4$, nous avons donc $10^{d-1} > 81d$ dès que $d \geqslant 4 > \alpha$ d'où nous déduisons $n \geqslant 10^{d-1} > 81d \geqslant sq(n)$ puis $n > sq(n)$ dès que $d \geqslant 4$. Ceci prouve le 1^er^ point.

Le 2nd point pour $d \leqslant 3$ découle directement de $sq(999) = 243$.

Fait 2.3. *$\forall n \in \mathbb{N}$, l'ensemble V_n est fini et donc la suite $\left(\boxed{n}_{\,k}\right)_{k\in\mathbb{N}}$ est ultimement périodique, i.e. périodique à partir d'un certain rang.*

Preuve. *Le 2nd point dépend directement du 1er point via le principe des tiroirs et la définition récursive de la suite $\left(\boxed{n}_{\,k}\right)_k$.*

Pour le 1er point, pour $n \leqslant 999$, on a directement $V_n \subset [0; 999]$, sinon il suffit de montrer que $V_n \subset \left[0; 10^{\texttt{taille}(n)}\right]$ pour $n \geqslant 10^4$ via une petite récurrence descendante finie.

3. Coder - Étudier la « période » d'un naturel

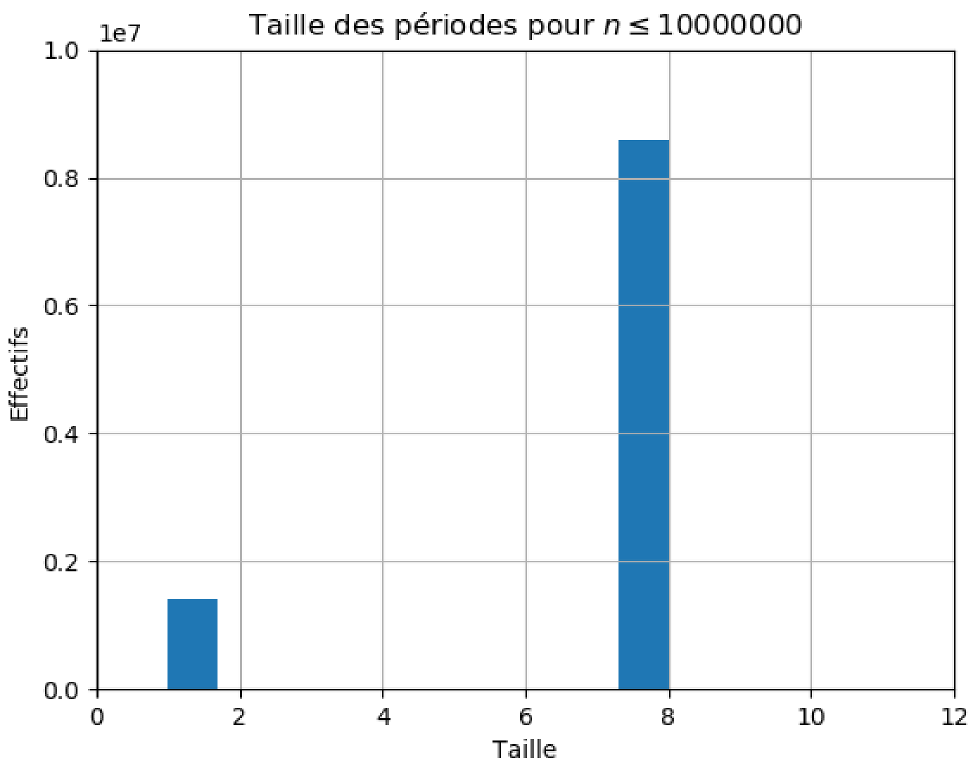
Quand il ne se fige pas, le code suivant donne la *« période »* d'un naturel auquel on applique le procédé.

```
n     = 20181209
nmemo = n

results = []

while n not in results:
    results.append(n)
    n = sum(int(d)**2 for d in str(n))

print(f"{nmemo} a la période suivante :")
print(results[results.index(n):])

print()

before = results[:results.index(n)]

if before:
    print("Avant la 1ère période nous avons :")
    print(before)
else:
    print("On commence directement par la période.")
```

Le code précédent, où `n = 20181209`, nous affiche :

```
20181209 a la période suivante :
[16, 37, 58, 89, 145, 42, 20, 4]

Avant la 1ère période nous avons :
[20181209, 155, 51, 26, 40]
```

Amusons-nous maintenant à représenter un histogramme des effectifs pour la taille des « périodes ». Le code utilisé se trouve à l'adresse `https://github.com/bc-writing/drafts` : voir le fichier `squareint-sizeplots.py` dans le dossier `squares-int`. Le traitement des données a été amélioré pour éviter de refaire des calculs déjà rencontrés *(pour plus de précisions, le lecteur curieux se reportera aux commentaires du code)*. Voici ce que l'on obtient [2].



Voilà quelque chose de frappant ! Il semblerait que l'on ait soit des périodes de taille 1, penser à 0 et 1, soit des périodes de taille 8 comme pour $37-58-89-145-42-20-4-16\ldots$ Magie ou coïncidence ? Les résultats de la section 2 vont nous permettre de le savoir. Dans la suite, nous reprenons les notations de la dite section.

Tout d'abord, comme $\texttt{taille}(sq(n)) < \texttt{taille}(n)$ dès que $\texttt{taille}(n) \geqslant 4$ d'après le fait 4.2, la périodicité n'arrivera que lorsque $\texttt{taille}\left(\boxed{n}_{\,k}\right) \leqslant 3$.

De plus, nous savons aussi que $\texttt{taille}(sq(n)) \leqslant 3$ dès que $\texttt{taille}(n) \leqslant 3$.

2. À l'adresse `https://github.com/bc-writing/drafts` dans le dossier `squares-int` vous trouverez l'image `befores.png` qui est un histogramme des effectifs pour la taille des suites de naturels juste avant la 1$^{\text{re}}$ *« période »*.

Tout ceci nous permet d'analyser brutalement via un programme ce qu'il se passe pour les périodes des naturels appartenant à $[\![0; 999]\!]$. Nous pouvons pour cela utiliser le code suivant, qui n'est absolument pas optimisé mais fait le travail immédiatement.

```
nmax = 999

periodsfound = []

for n in range(nmax + 1):
    results = []

    while n not in results:
        results.append(n)
        n = sum(int(d)**2 for d in str(n))

    period = results[results.index(n):]

    if period not in periodsfound:
        periodsfound.append(period)

for oneperiod in periodsfound:
    print(oneperiod)
```

Ce code nous fournit toutes les périodes possibles.

```
[0]
[1]
[4, 16, 37, 58, 89, 145, 42, 20]
[37, 58, 89, 145, 42, 20, 4, 16]
[89, 145, 42, 20, 4, 16, 37, 58]
[16, 37, 58, 89, 145, 42, 20, 4]
[20, 4, 16, 37, 58, 89, 145, 42]
[58, 89, 145, 42, 20, 4, 16, 37]
[42, 20, 4, 16, 37, 58, 89, 145]
[145, 42, 20, 4, 16, 37, 58, 89]
```

Et là cela devient joli car nous notons au passage que trois types de périodes : `[0]`, `[1]` et `[4, 16, 37, 58, 89, 145, 42, 20]` avec toutes ses *« permutées circulaires »*.

4. Peut-on généraliser à un exposant $p \geqslant 3$?

Pour finir, nous allons analyser ce qu'il se passe si l'on somme à la puissance $p \geqslant 3$ au lieu d'élever au carré. Nous reprenons des notations similaires à celles de la section 2.

- Pour un naturel $n = [c_{d-1}c_{d-2}\cdots c_1c_0]_{10}$ avec $c_{d-1} \neq 0$, on pose $pw(n) = \sum_{k=0}^{d-1} (c_k)^p$ et $\texttt{taille}(n) = d$.
- Pour $(n\,;k) \in \mathbb{N}^2$, on définit $\boxed{n}_{\,0} = n$ et $\boxed{n}_{\,k+1} = pw\left(\boxed{n}_{\,k}\right)$.

Fait 4.1. *$\forall n \in \mathbb{N}$, $pw(n) \leqslant 9^p\, d$ où $d = \texttt{taille}(n)$.*

Preuve. *Si $n = [\, c_{d-1}c_{d-2}\cdots c_1c_0\,]_{10}$ alors $pw(n) = \displaystyle\sum_{k=0}^{d-1}(c_k)^p \leqslant \sum_{k=0}^{d-1} 9^p = 9^p\, d$.*

Fait 4.2. *Il existe $d_0 \in \mathbb{N}$ tel que $\forall n \in \mathbb{N}$, $[\,\texttt{taille}(n) \geqslant d_0 \Rightarrow pw(n) < n\,]$.*
On peut en fait choisir $d_0 = 3 + \left\lfloor \dfrac{1}{\ln 10} \ln\left(\dfrac{9^p\, d}{\ln 10}\right)\right\rfloor$ où $\lfloor x \rfloor$ désigne la partie entière de x.

Preuve. *Notons $d = \texttt{taille}(n)$ de sorte que $n \geqslant 10^{d-1}$. Compte tenu du fait précédent, nous cherchons à comparer 10^{d-1} et $9^p\, d$. Comme dans le cas $p = 2$ démontré dans la section 2, on considère sur $\mathbb{R}_+$ la fonction $\Delta(x) = 10^{x-1} - 9^p\, d\, x$ qui vérifie $\Delta'(x) > 0$ si et seulement si $x > 1 + \dfrac{1}{\ln 10}\ln\left(\dfrac{9^p\, d}{\ln 10}\right)$.*

Notant α le réel précédent, nous savons donc que $n \geqslant 10^{d-1} > 9^p\, d \geqslant pw(n)$, puis $n > pw(n)$ dès que $d > \alpha$. Cette dernière condition est vérifiée dès que $d \geqslant d_0$ avec le d_0 proposé un peu plus haut.

Fait 4.3. *$\forall n \in \mathbb{N}$, la suite $\left(\boxed{n}_{\,k}\right)_{k\in\mathbb{N}}$ est ultimement périodique.*

Preuve. *Tout est en fait contenu dans le fait 4.2, dont on reprend la définition de d_0. Expliquons pourquoi.*

- *Le fait 4.2 donne l'existence d'un indice $k_0 \in \mathbb{N}$ tel que $\texttt{taille}\left(\boxed{n}_{\,k_0}\right) < d_0$* (dans le cas contraire, on pourrait construire une suite strictement décroissante de naturels).
- *Si pour tout naturel $k \in [\![k_0; +\infty[\![$, $\texttt{taille}\left(\boxed{n}_{\,k}\right) < d_0$, nous avons l'ultime périodicité via le principe des tiroirs* (si besoin revoir la fin de la section 2).
- *Sinon il existe $k_0' \in]\!]k_0; +\infty[\![$ tel que $\texttt{taille}\left(\boxed{n}_{\,k_0'}\right) \geqslant d_0$. Comme dans le premier point, nous pouvons alors trouver $k_1 \in]\!]k_0'; +\infty[\![$ tel que $\texttt{taille}\left(\boxed{n}_{\,k_1}\right) < d_0$.*
- *En répétant notre raisonnement, on peut aboutir à une situation similaire au 2^e^ point, et c'est gagné.*
  Sinon on arrive à construire une suitre strictement croissante $(k_i)_i$ d'indices tels que $\forall i \in \mathbb{N}$, $\texttt{taille}\left(\boxed{n}_{\,k_i}\right) < d_0$. Le principe des tiroirs s'applique ici aussi !

Remarque 4.1. *La preuve précédente montre que pour rechercher toutes les périodes il* « suffit » *d'étudier les naturels appartenant à $[\![0; 10^{d_0} - 1]\!]$.*