Périodicité d'une suite modulo 10

Rois-Céti DIMBAMBA

10 Décembre 2023

1 Périodicité d'une suite modulo 10

Soit la suite $(x_n)_{n\in\mathbb{N}}$ définie par :

$$\begin{cases} x_0 = 0 \\ x_n = 1^1 + 2^2 + 3^3 + \ldots + n^n, \ \ n \in \mathbb{N}^* \end{cases}$$

Pour tout $n \in \mathbb{N}$, désignons par b_n, le chiffre des unités de x_n.
Montrer que la suite $(b_n)_{n\in\mathbb{N}}$ ainsi construite est $100-$périodique.

1.0.1 Preuve :

Le but de l'exercice est de montrer que : $\forall\, n \in \mathbb{N}, \ b_{n+100} = b_n$.
<u>*Notation*</u> : Si r désigne le reste de la division euclidienne de $a \in \mathbb{N}$ par $b \in \mathbb{N}^*$, on adopte la notation (N_1): $r = a\%b$, avec $r \in \{0, 1, \ldots, b-1\}$.
Avec cette notation, on peut écrire : $b_n = x_n\%10, \ \ n \in \mathbb{N}$.
Pour tout $n \in \mathbb{N}$, désignons par $m_n \in \mathbb{N}^*$, le nombre de chiffres de l'entier n, et pour tout $j \in \{0, \ldots, m_n - 1\}$, désignons par $c_n^{(j)}$, le $(j+1)^{me}$ chiffre dans l'écriture de n en base 10, en partant du chiffre des unités. Ainsi, $c_n^{(0)}$ est le chiffre des unités, $c_n^{(1)}$ le chiffre des dizaines, $c_n^{(2)}$ le chiffre des centaines, ..., de l'entier n.
Avec ces notations, on peut écrire : $n = \sum\limits_{j=0}^{m_n-1} c_n^{(j)} 10^j$, avec $c_n^{(j)} \in \{0, 1, \ldots, 9\}$ pour tout $j \in \{0, \ldots, m_n - 1\}$, $c_n^{(m_n-1)}$ étant non nul.
On va adopter la convention (C_1) : $\sum\limits_{j=a}^{b} \ldots = 0$ si $b < a$.
Pour tout $n \in \mathbb{N}$, on va poser : $q_n = \dfrac{1}{4}\sum\limits_{j=2}^{m_n-1} c_n^{(j)} 10^j = \sum\limits_{j=2}^{m_n-1} c_n^{(j)} 5^j 2^{j-2}$ (avec la convention (C_1), $q_n = 0$ si $m_n \leqslant 2$ et de plus, $q_n \in 25\mathbb{N}^*$ si $m_n \geqslant 3$), de façon à ce que n s'écrive de la forme : $n = 4q_n + 10c_n^{(1)} + c_n^{(0)}$.

Pour tout $k \in \mathbb{N}^*$, $k^k = \left(c_k^{(0)}\right)^k [10]$, et donc pour tout $n \in \mathbb{N}^*$, on a :

$$\left.\begin{aligned}
x_n = {} & 1^{4\times0+10\times0+1} + 2^{4\times0+10\times0+2} + 3^{4\times0+10\times0+3} + \ldots + 9^{4\times0+10\times0+9} \\
& + 1^{4\times0+10\times1+1} + 2^{4\times0+10\times1+2} + 3^{4\times0+10\times1+3} + \ldots + 9^{4\times0+10\times1+9} \\
& + \ldots \\
& + 1^{4\times0+10\times9+1} + 2^{4\times0+10\times2+9} + 3^{4\times0+10\times9+3} + \ldots + 9^{4\times0+10\times9+9} \\
& + 1^{4\times25+10\times0+1} + 2^{4\times25+10\times0+2} + 3^{4\times25+10\times0+3} + \ldots + 9^{4\times25+10\times0+9} \\
& + 1^{4\times25+10\times1+1} + 2^{4\times25+10\times1+2} + 3^{4\times25+10\times1+3} + \ldots + 9^{4\times25+10\times1+9} \\
& + \ldots \\
& + 1^{4\times25+10\times9+1} + 2^{4\times25+10\times9+2} + 3^{4\times25+10\times9+3} + \ldots + 9^{4\times25+10\times9+9} \\
& + 1^{4\times50+10\times0+1} + 2^{4\times50+10\times0+2} + 3^{4\times50+10\times0+3} + \ldots + 9^{4\times50+10\times0+9} \\
& + 1^{4\times50+10\times1+1} + 2^{4\times50+10\times1+2} + 3^{4\times50+10\times1+3} + \ldots + 9^{4\times50+10\times1+9} \\
& + \ldots \\
& + \ldots \\
& + 1^{4\times225+10\times9+1} + 2^{4\times225+10\times9+2} + 3^{4\times225+10\times9+3} + \ldots + 9^{4\times225+10\times9+9} \\
& + 1^{4\times250+10\times0+1} + 2^{4\times250+10\times0+2} + 3^{4\times250+10\times0+3} + \ldots + 9^{4\times250+10\times0+9} \\
& + 1^{4\times250+10\times1+1} + 2^{4\times250+10\times1+2} + 3^{4\times250+10\times1+3} + \ldots + 9^{4\times250+10\times1+9} \\
& + \ldots \\
& + \ldots \\
& + \ldots \\
& + \ldots \\
& + 1^{4p_n+10d_n+1} + 2^{4p_n+10d_n+2} + 3^{4p_n+10d_n+3} + \ldots + 9^{4p_n+10d_n+9}
\end{aligned}\right\} \begin{array}{l} \frac{1}{10}(n - c_n^{(0)}) \in \mathbb{N} \text{ paquets} \\ \text{de somme de 9 termes} \end{array}$$

$$+1^{4q_n+10c_n^{(1)}+1} + 2^{4q_n+10c_n^{(1)}+2} + \ldots + \left(c_n^{(0)}\right)^{4q_n+10c_n^{(1)}+c_n^{(0)}} \quad [10]$$

avec la convention que le dernier paquet de somme est nul pour $c_n^{(0)} = 0$, où
$\begin{cases} p_n = q_n \text{ et } d_n = c_n^{(1)} - 1 \text{ si } c_n^{(1)} \in \{1, 2, \ldots, 9\} \\ p_n = (q_n - 25)\mathbb{1}_{m_n \geqslant 3} \text{ et } d_n = 9 \times \mathbb{1}_{m_n \geqslant 3} - \mathbb{1}_{m_n \leqslant 2} \text{ si } c_n^{(1)} = 0 \end{cases}$
Ce qui peut s'écrire en forme condensée, pour tout $n \in \mathbb{N}$:

$$x_n = \underbrace{\underbrace{\sum_{i=0}^{\frac{q_n}{25}-1} \sum_{j=0}^{9} \sum_{k=1}^{9} k^{4\times25i+10j+k}}_{\frac{1}{100}(n-10c_n^{(1)}-c_n^{(0)})\times10 \text{ paquets du type } \sum_{k=1}^{9}\cdots} + \underbrace{\sum_{j=0}^{d_n} \sum_{k=1}^{9} k^{4p_n+10j+k}}_{(d_n+1) \text{ paquets du type } \sum_{k=1}^{9}\cdots}}_{\frac{1}{10}(n-c_n^{(0)}) \text{ paquets du type } \sum_{k=1}^{9}\cdots} + \sum_{k=1}^{c_n^{(0)}} k^{4q_n+10c_n^{(1)}+k} \ [10]$$

$$\left(\text{avec } \frac{1}{100}(n - 10c_n^{(1)} - c_n^{(0)}) = \frac{q_n}{25} \in \mathbb{N}\right)$$

en tenant compte de la convention (C_1).

On va maintenant chercher à déterminer $\left(1^{4a+10b+1} + 2^{4a+10b+2} + 3^{4a+10b+3} + \ldots + 9^{4a+10b+9}\right) \%10$, pour tout $(a, b) \in \mathbb{N} \times \{0, 1, 2, \ldots, 9\}$.
Pour ce faire, on va d'abord commencer par déterminer les restes dans la division euclidienne par 10 de $2^p, 3^p, \ldots, 9^p$, suivant les valeurs de $p \in \mathbb{N}$.

Un calcul rapide nous donne les résultats ci-après :

$$2^p\%10 = \begin{cases} 1 \text{ si } p=0 \\ 6 \text{ si } p=4k,\ k\in\mathbb{N}^* \\ 2 \text{ si } p=4k+1,\ k\in\mathbb{N} \\ 4 \text{ si } p=4k+2,\ k\in\mathbb{N} \\ 8 \text{ si } p=4k+3,\ k\in\mathbb{N} \end{cases} \qquad 3^p\%10 = \begin{cases} 1 \text{ si } p=4k,\ k\in\mathbb{N} \\ 3 \text{ si } p=4k+1,\ k\in\mathbb{N} \\ 9 \text{ si } p=4k+2,\ k\in\mathbb{N} \\ 7 \text{ si } p=4k+3,\ k\in\mathbb{N} \end{cases}$$

$$4^p\%10 = \begin{cases} 1 \text{ si } p=0 \\ 6 \text{ si } p=2k,\ k\in\mathbb{N}^* \\ 4 \text{ si } p=2k+1,\ k\in\mathbb{N} \end{cases} \qquad 5^p\%10 = \begin{cases} 1 \text{ si } p=0 \\ 5 \text{ si } p\in\mathbb{N}^* \end{cases} \qquad 6^p\%10 = \begin{cases} 1 \text{ si } p=0 \\ 6 \text{ si } p\in\mathbb{N}^* \end{cases}$$

$$7^p\%10 = \begin{cases} 1 \text{ si } p=4k,\ k\in\mathbb{N} \\ 7 \text{ si } p=4k+1,\ k\in\mathbb{N} \\ 9 \text{ si } p=4k+2,\ k\in\mathbb{N} \\ 3 \text{ si } p=4k+3,\ k\in\mathbb{N} \end{cases} \qquad 8^p\%10 = \begin{cases} 1 \text{ si } p=0 \\ 6 \text{ si } p=4k,\ k\in\mathbb{N}^* \\ 8 \text{ si } p=4k+1,\ k\in\mathbb{N} \\ 4 \text{ si } p=4k+2,\ k\in\mathbb{N} \\ 2 \text{ si } p=4k+3,\ k\in\mathbb{N} \end{cases}$$

$$9^p\%10 = \begin{cases} 1 \text{ si } p=2k, k\in\mathbb{N} \\ 9 \text{ si } p=2k+1, k\in\mathbb{N} \end{cases}$$

Comme $b\in\{0,1,2,\dots,9\}$, alors il peut s'écrire sous la forme $b=2s$ ou $b=2s+1$ avec $s\in\{0,1,2,3,4\}$.
On a :

$$\sum_{k=1}^{9} k^{4a+10b+k} = \begin{cases} \sum_{k=1}^{9} k^{4(a+5s)+k} & \text{si } b=2s \\ \sum_{k=1}^{9} k^{4(a+5s)+10+k} & \text{si } b=2s+1 \end{cases}$$
$$= \begin{cases} 1+2^{4(a+5s)+2}+3^{4(a+5s)+3}+4^{2(2a+10s+2)}+5^{4(a+5s)+5}+6^{4(a+5s)+6}+7^{4(a+5s+1)+3}+8^{4(a+5s+2)}+ \\ +9^{2(2a+10s+4)+1} \quad \text{si } b=2s \\ 1+2^{4(a+5s+3)}+3^{4(a+5s+3)+1}+4^{2(2a+10s+7)}+5^{4(a+5s)+15}+6^{4(a+5s)+16}+7^{4(a+5s+4)+1}+8^{4(a+5s+4)+2}+ \\ +9^{2(2a+10s+9)+1} \quad \text{si } b=2s+1 \end{cases}$$
$$= \begin{cases} 1+4+7+6+5+6+3+6+9\,[10]=27\,[10]=7\,[10] & \text{si } b=2s \\ 1+6+3+6+5+6+7+4+9\,[10]=27\,[10]=7\,[10] & \text{si } b=2s+1 \end{cases}$$

Il s'ensuit que pour tout $(a,b)\in\mathbb{N}\times\{0,1,2,\dots,9\}$:

$$\left(\sum_{k=1}^{9} k^{4a+10b+k}\right)\%10 = 7$$

On en déduit que pour tout $n\in\mathbb{N}$:

$$x_n = \left(\frac{1}{10}(n-c_n^{(0)})\times 7 + \sum_{k=1}^{c_n^{(0)}} k^{4q_n+10c_n^{(1)}+k}\right)\ [10]$$

Par ailleurs, puisque pour tout $n \in \mathbb{N}$, $n + 100 = n\,[100]$, alors $n + 100$ et n ont les mêmes chiffres des unités et des dizaines, donc $c_n^{(0)} = c_{n+100}^{(0)}$ et $c_n^{(1)} = c_{n+100}^{(1)}$.
Ainsi, pour tout $n \in \mathbb{N}$:

$$\begin{aligned} x_{n+100} &= \left(\frac{1}{10}(n + 100 - c_{n+100}^{(0)}) \times 7 + \sum_{k=1}^{c_{n+100}^{(0)}} k^{4q_{n+100}+10c_{n+100}^{(1)}+k} \right) [10] \\ &= \left(\frac{1}{10}(n - c_n^{(0)}) \times 7 + \sum_{k=1}^{c_n^{(0)}} k^{4q_{n+100}+10c_n^{(1)}+k} \right) [10] \end{aligned}$$

Donc, pour tout $n \in \mathbb{N}$:

$$x_{n+100} - x_n = \left(\sum_{k=1}^{c_n^{(0)}} k^{10c_n^{(1)}+k}(k^{4q_{n+100}} - k^{4q_n}) \right) [10]$$

On va distinguer deux cas :

- Si $m_n \geqslant 3$, alors $q_n \in \mathbb{N}^*$ (car $c_n^{(2)}$ est non nul) et donc aussi $q_{n+100} \in \mathbb{N}^*$. Ainsi, pour tout $k \in \{1, 2, \ldots, 9\}$, $k^{4q_{n+100}} = k^{4q_n}\,[10]$ en utilisant les résultats précédents.
- Si maintenant $m_n \leqslant 2$, alors $q_n = 0$ et $q_{n+100} = 25$. Ainsi, pour tout $k \in \{1, 2, \ldots, 9\}$:

$$k^{4q_{n+100}} - k^{4q_n} = k^{4\times 25} - 1 = \begin{cases} 0\,[10] \text{ si } k \in \{1, 3, 7, 9\} \\ 5\,[10] \text{ si } k \in \{2, 4, 6, 8\} \\ 4\,[10] \text{ si } k = 5 \end{cases}$$

Et donc,

$$k^{10c_n^{(1)}+k}(k^{4q_{n+100}} - k^{4q_n}) = k^{10c_n^{(1)}+k}(k^{4\times 25} - 1) = \begin{cases} 0\,[10] \text{ si } k \in \{1, 3, 7, 9\} \\ 2K \times 5\,[10] = 0\,[10] \text{ si } k \in \{2, 4, 6, 8\} \\ 5K' \times 4\,[10] = 0\,[10] \text{ si } k = 5 \end{cases}$$

$$\text{où} \quad \begin{cases} K = \frac{k}{2} k^{10c_n^{(1)}+k-1} \in \mathbb{N}^* \text{ si } k \in \{2, 4, 6, 8\} \\ K' = 5^{10c_n^{(1)}+4} \in \mathbb{N}^* \text{ si } k = 5 \end{cases}$$

Dans tous les cas, on a que pour tout $k \in \{1, 2, \ldots, 9\}$, $k^{10c_n^{(1)}+k}(k^{4q_{n+100}} - k^{4q_n}) = 0\,[10]$.
D'où, l'on déduit que pour tout $n \in \mathbb{N}$, $x_{n+100} = x_n\,[10]$, et donc $b_{n+100} = b_n$. Ce qui achève la preuve.

2 Ilustrastion via un mini-code python

On va commencer par écrire une fonction `unit_digit` qui prend en entrée un couple $(a, p) \in \mathbb{N}^2$ et qui renvoie le chiffre des unités du nombre a^p.
Il y a un réel gain-machine à ne pas calculer la puissance a^p dans le code, et cela est possible grâce aux restes dans la division euclidienne par 10 de $2^p, 3^p, \ldots, 9^p$ suivants les valeurs de $p \in \mathbb{N}$, que nous avons déterminés plus haut. Nous allons donc utiliser ces restes afin de contourner le calcul de la puissance a^p qui a un énorme coût-machine lorsque a et p deviennent de plus en plus grand.
Voici le code :

[85]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import nbconvert
```

[1]:
```python
def unit_digit(a, p):
    '''Cette fonction détermine le reste dans la division euclidienne par 10 de␣
↪a^p
    où a et p sont des entiers naturels non simultanément nuls, auquel cas la␣
↪puissance a^p n'aurait plus de sens.'''

    result = None

    if not (isinstance(a, int) & isinstance(p, int)):
        result = "L'une au moins des entrées a (base) ou p (exposant) n'est pas␣
↪un entier naturel"
    elif a<0:
        result = "L'entrée a (base) n'est pas un entier naturel (donc positif)"
    elif p<0:
        result = "L'entrée p (exposant) n'est pas un entier naturel (donc␣
↪positif)"
    elif (a==0) & (p==0):
        result = "La puissance a^p n'est pas définie "
    else:
        d = a % 10 # a**p = d**p [10] avec d=a%10 appartenant à {0,1,...,9}
        if p==0:
            result = 1
        elif d == 0:
            result = 0
        elif d == 1:
            result = 1
        elif d == 2:
            temp = p % 4
            if temp == 0:
                result = 6
            elif temp == 1:
                result = 2
            elif temp == 2:
                result = 4
            else:
                result = 8
        elif d == 3:
            temp = p % 4
            if temp == 0:
                result = 1
            elif temp == 1:
                result = 3
```

```
            elif temp == 2:
                result = 9
            else:
                result = 7
        elif d == 4:
            temp = p % 2
            if temp == 0:
                result = 6
            else:
                result = 4
        elif d == 5:
            result = 5
        elif d == 6:
            result = 6
        elif d == 7:
            temp = p % 4
            if temp == 0:
                result = 1
            elif temp == 1:
                result = 7
            elif temp == 2:
                result = 9
            else:
                result = 3
        elif d == 8:
            temp = p % 4
            if temp == 0:
                result = 6
            elif temp == 1:
                result = 8
            elif temp == 2:
                result = 4
            else:
                result = 2
        else:
            temp = p % 2
            if temp == 0:
                result = 1
            else:
                result = 9

    return result
```

Une fois avoir écrit cette fonction, puisque $x_0 = 0$ et pour tout $n \in \mathbb{N}$, $x_{n+1} = x_n + (n+1)^{n+1}$, on aura alors :

$$\begin{cases} b_0 = 0 \\ b_{n+1} = (b_n + \texttt{unit\_digit}(n+1, n+1))\%10 \;\; \forall\, n \in \mathbb{N} \end{cases}$$

Ci-après, le code pour calculer les termes de la suite $(b_n)_{n\in\mathbb{N}}$, partant de l'initialisation $b_0 = 0$.

[3]:
```
k = 100 # nombre de paquets de 100 termes de la suite (b_n)
n = 100*k # le 100 qui multiplie le k représente la période de la suite (b_n)
b = np.zeros(n) # le vecteur qui va contenir les termes de la suites (b_n)

for i in np.arange(n-1):
    b[i+1] = (b[i] + unit_digit(int(i+1), int(i+1))) % 10
```

On représente les termes de la suite $(b_n)_{n\in\mathbb{N}}$ dans un dataframe pandas de k lignes et 100 colonnes, de sorte que l'élément se trouvant à la ligne i ($i \in \{0,1,\ldots,k-1\}$) et à la colonne j ($j \in \{0,1,\ldots,99\}$) soit b_{100i+j}. Après, on pourra jouer avec la valeur de k dans le code ci-dessus pour rajouter les lignes dans le dataframe et constater la $100-$périodicité de la suite $(b_n)_{n\in\mathbb{N}}$. Visuellement, cela se constate en remarquant que pour une colonne fixée, les éléments correspondants à toutes les lignes sont égaux.

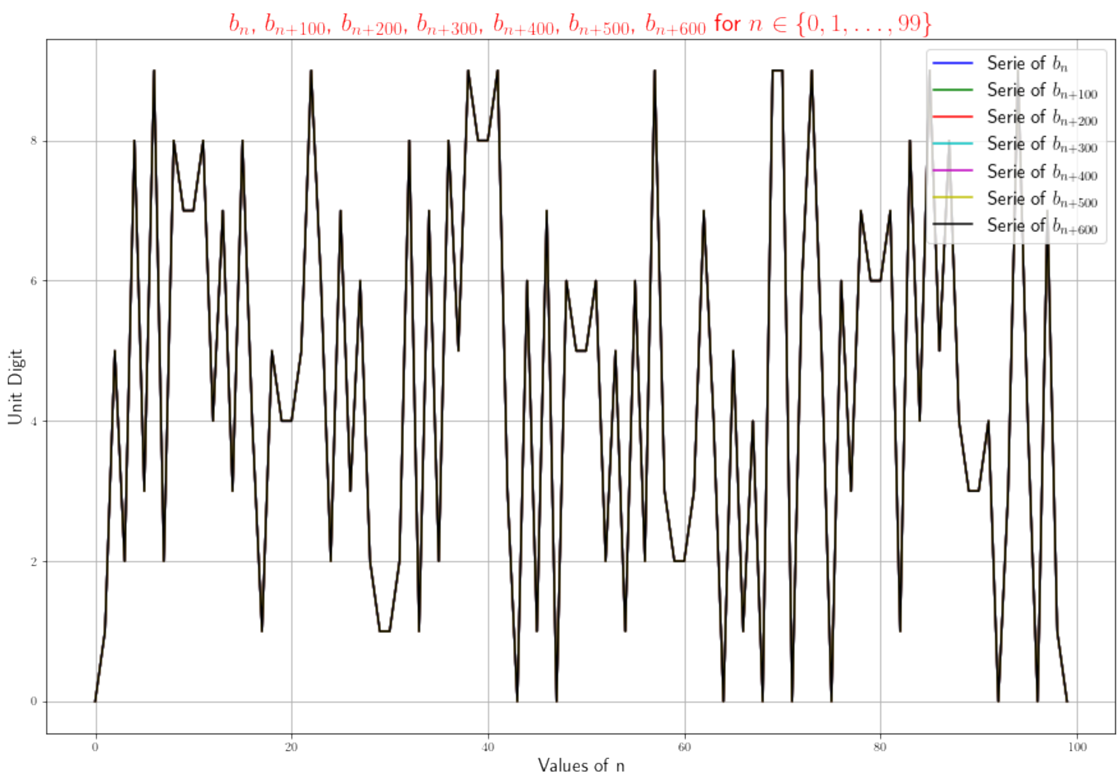
[4]:
```
dicos = {}
for i in np.arange(100):
    dicos[f"{i}"] = b[100*np.arange(k) + i]

df = pd.DataFrame(dicos)
df
```

[4]:
```
      0    1    2    3    4    5    6    7    8    9  ...   90   91   92   93  \
0   0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
1   0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
2   0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
3   0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
4   0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
..  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
95  0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
96  0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
97  0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
98  0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0
99  0.0  1.0  5.0  2.0  8.0  3.0  9.0  2.0  8.0  7.0  ...  3.0  4.0  0.0  3.0

     94   95   96   97   98   99
0   9.0  4.0  0.0  7.0  1.0  0.0
1   9.0  4.0  0.0  7.0  1.0  0.0
2   9.0  4.0  0.0  7.0  1.0  0.0
3   9.0  4.0  0.0  7.0  1.0  0.0
4   9.0  4.0  0.0  7.0  1.0  0.0
..  ...  ...  ...  ...  ...  ...
95  9.0  4.0  0.0  7.0  1.0  0.0
96  9.0  4.0  0.0  7.0  1.0  0.0
97  9.0  4.0  0.0  7.0  1.0  0.0
98  9.0  4.0  0.0  7.0  1.0  0.0
99  9.0  4.0  0.0  7.0  1.0  0.0
```

```
[100 rows x 100 columns]
```

“Un tableau, c’est bien, mais un graphique, c’est mieux”, disait un sage.
C’est la raison pour laquelle on va faire un graphique, représentant quelques paquets (ici 7) de 100 termes de la suite $(b_n)_{n\in\mathbb{N}}$, chaque paquet étant représenté par une courbe.

[84]:
```python
h = np.arange(100)

plt.rcParams['text.usetex'] = True
plt.figure(figsize = (15,10))
color = ['b', 'g', 'r', 'c', 'm', 'y', 'k']
title = ["$b_{n}$"]
for i, col in enumerate(color):
    plt.plot(h, b[100*i:100*(i+1)], label = "Serie of " + title[i], c = col)
    if i < len(color)-1: title.append("$b_{n+"+str(100*(i+1))+"}$")
plt.xlabel("Values of n", fontsize = 15)
plt.ylabel("Unit Digit", fontsize = 15)
plt.title(", ".join(title) + r" for $n\in \{0, 1, \ldots, 99\}$", fontsize = 20, c = "red")
plt.legend(loc = "upper right", fontsize = 15)
plt.grid(True)
plt.show()
```



On voit dans ce graphique que les séquences $(b_n)_{n\in\{100i,100i+1,\ldots,100i+99\}}$ pour $i \in \{0,1,\ldots,6\}$ sont exactement les mêmes car leurs graphiques se superposent, ce qui permet de visualiser graphiquement la $100-$périodicité de la suite $(b_n)_{n\in\mathbb{N}}$.

Par Le Rois-Céti