Sommes des cubes des chiffres d'un entier naturel

Patrick ROYIS

13 décembre 2022

Résumé

On se propose, dans cet article, d'étudier les propriétés de la suite des sommes des cubes des chiffres d'un entier naturel.

Mots-clés Sommes — Cubes

De la suite des sommes des cubes des chiffres d'un enntier naturel

Prouvons le

Théorème 1 (la pêche miraculeuse) *Soit p un entier naturel non nul et soit*

$$n = \sum_{k=0}^{p-1} a_k 10^k \quad \text{avec} \quad a_k \in [\![0, 9]\!] \quad \forall k \in [\![0, p-1]\!] \quad \text{et} \quad a_{p-1} \neq 0$$

un entier naturel non nul à p chiffres donné. Posons

$$f(n) = \sum_{k=0}^{p-1} a_k^3$$

Soit $(u_m)_{m\in\mathbb{N}}$ la suite définie par $u_0 = n$ et, pour chaque entier naturel m, $u_{m+1} = f(u_m)$. Alors,

1. *Si n est un multiple de 3, la suite $(u_m)_{m\in\mathbb{N}}$ est constante et égale à 153 à partir d'un certain rang.*
2. *Si n est congru à 1 modulo 3, la suite $(u_m)_{m\in\mathbb{N}}$ est, à partir d'un certain rang, soit constante et égale à 1 ou à 370, soit cyclique, de cycle appartenant à l'ensemble*
$$\{(55, 250, 133),\ (136, 244),\ (160, 217, 352),\ (919, 1459)\}$$
3. *Si n est congru à 2 modulo 3, la suite $(u_m)_{m\in\mathbb{N}}$ est constante à partir d'un certain rang et égale à 371 ou à 407.*

Preuve

1. Soit, pour chaque entier naturel k, u_k le terme de rang k de la suite $(u_m)_{m\in\mathbb{N}}$, et soit p_k le nombre de chiffres de ce terme. On a alors $u_k \geq 10^{p_k-1}$. L'on a par ailleurs $u_{k+1} = f(u_k) \leq 9^3 p_k = 729p_k$. Or, si p_k est supérieur ou égal à 5, l'on a $729p_k < 10^{p_k-1}$, et donc $u_{k+1} < u_k$. Ainsi, si l'entier naturel u_0 possède au moins 5 chiffres, la suite $(u_m)_{m\in\mathbb{N}}$ décroit strictement dans un premier temps, de sorte que pour un certain entier naturel non nul k_0, le terme u_{k_0} de cette suite possède au plus 4 chiffres. Ainsi, il suffit de montrer que le théorème 1 est satisfait par toutes les suites $(u_m)_{m\in\mathbb{N}}$ pour lesquelles u_0 est un entier naturel non nul et inférieur à 9999, ce qui peut aisément se faire grâce à un algorithme. Le code Javascript suivant

```
<script>

function Test()
{
var cong, nn, n, mat, mat1, mat2, mat3, niter, finiter, nit, nitmax,
text, text1, text2, text3 ;
text = "" ;
mat1 = [] ;
mat2 = [] ;
mat3 = [] ;
  for (nn=0 ; nn<=9996 ; nn+=3)
  {
    for (cong=1 ; cong<=3 ; cong++)
    {
    mat = [] ;
    n = nn + cong ;
    mat.push(n) ;
    text = text + "Terme initial égal  " + n + " : " ;
    niter = 1 ;
    finiter = 0 ;
      while (finiter == 0)
      {
      niter = niter + 1 ;
      n = scub(n) ;
      mat.push(n) ;
      nit = 0 ;
      nitmax = niter -2
        while((nit <= nitmax) && (finiter == 0))
        {            if (mat[nit] == n)
          {
          finiter = 1
          text = text + n ;
            if (cong == 1)
            {
```

```
            mat1.push(n) ;
            }
            if (cong == 2)
            {
            mat2.push(n) ;
            }
            if (cong == 3)
            {
            mat3.push(n) ;
            }
            if (n == scub(n))
            {
            text = text + " est un point fixe.<br>"
            }
            else
            {
            text = text + " est le début d'un cycle de longueur
" +
            (niter-nit-1) + ".<br>"
            }
          }
        nit = nit + 1 ;
        }
      }
    }
  }
mat1 = mattri(matred(mat1)) ;
mat2 = mattri(matred(mat2)) ;
mat3 = mattri(matred(mat3)) ;
text = text + "<br>%%%%%%%%%%%%%%%%%%%" ;
text = text + "<br>%%%%% R&#233capitulatif %%%%" ;
text = text + "<br>%%%%%%%%%%%%%%%%%%%<br>" ;
text = text + "<br> Nombres congrus  1 modulo 3 :<br>" + cycle(mat1) ;
text = text + "Nombres congrus  2 modulo 3 :<br>" + cycle(mat2) ;
text = text + "Nombres multiples de 3 :<br>" + cycle(mat3) ;
document.getElementById("result").innerHTML = text ;
}

function scub(n)
{var cub ;
cub = 0 ;
  while (n > 0)
  {
  u = n % 10 ;
  cub = cub + u*u*u ;
  n = Math.floor(n/10) ;
```

```
  }
return cub ;
}

function matred(mat)
{var long, matr, i, mi, j, ifin ;
long = mat.length - 1 ;
matr =[] ;
matr.push(mat[0])
  for (i=1 ; i<=long ; i++)
  {
  mi = mat[i] ;
  j = 0 ;
  ifin = 1 ;
    while ((j <= i-1) && (ifin == 1))
    {
      if (mat[j] == mi)
      {
      ifin = 0 ;
      }
    j = j + 1 ;
    }
    if (ifin == 1)
    {
    matr.push(mi) ;
    }
  }
return matr ;
}

function mattri(mat)
{
var long, matt, i, mi, j, mj ;
long = mat.length - 1 ;
matt = mat ;
  for (i=0 ; i<=long-1 ; i++)
  {
    for (j=i+1 ; j<=long ; j++)
    {
    mi = matt[i] ;
    mj = matt[j] ;
      if (mj < mi)
      {
      matt[i] = mj ;
      matt[j] = mi
      }
    }
```

```
  }
return matt ;
}

function cycle(mat)
{var text, ltext, long, i, mi, nn, nit ;
long = mat.length - 1 ;
text = "" ;
  for (i=0 ; i<=long ; i++)
  {
  mi = mat[i] ;
  text = text + "{" + mi + ", " ;
  nn = scub(mi)
  nit = 1 ;
    while (nn != mi)
    {
    text = text + nn + ", " ;
    nn = scub(nn) ;
    nit = nit + 1 ;
    }
  ltext=text.length ;
    if (nit == 1)
    {
    text = text.substr(0,ltext-2) + "} est un point fixe.<br>" ;
    }
    else
    {
    text = text.substr(0,ltext-2) + "} est un cycle de longueur
" + nit +
    ".<br>" ;
    }
  }
text = text + "<br>" ;
return text ;
}

</script>
```

montre que le théorème 1 est satisfait par toutes les suites $(u_m)_{m\in\mathbb{N}}$ telles que u_0 est un entier naturel non nul et inférieur à 9999, ce qui achève sa preuve. □