Les suites aliquotes

Travail de maturité réalisé au Lycée Blaise-Cendrars de La Chaux-de-Fonds
sous la direction de M. Jean-Bernard Mathey

Etienne Grezet

Avant-propos

Dès le départ je me suis dit que j'allais faire mon travail de maturité en mathématiques, je n'avais pas d'idée vraiment précise mais dès que j'ai vu sur la feuille des sujets : les nombres parfaits, j'avais trouvé mon sujet. Car en application des maths, on avait déjà dû faire un programme qui trouve un nombre parfait et j'avais trouvé cela magnifique. Alors pouvoir faire mon travail de maturité là-dessus... Mais très vite la question s'est posée : y a-t-il assez de matière pour en faire un travail de maturité ? Alors avec mon mentor nous avons étudié le sujet et il est apparu que grâce aux suites aliquotes je pouvais parler des nombres parfaits, amicaux et sociables. Le titre de mon travail de maturité était alors trouvé.

Introduction sur la théorie des nombres

Définition

La théorie des nombres est une branche des mathématiques qui s'intéresse aux relations et aux propriétés des nombres entiers. Cette branche des mathématiques passionne beaucoup de mathématiciens car elle compte un grand nombre de problèmes qui ouverts, mais dont l'énoncé est assez simple à comprendre. Pour travailler dans la théorie des nombres, nous n'avons généralement besoin que des quatre opérations de base des mathématiques, c'est-à-dire : $+ - \div \cdot$. Son accessibilité est ce qui est principalement intéressant dans la théorie des nombres. Elle s'intéresse par exemple aux critères de divisibilité, à la nature des nombres (premiers ou pas), etc.

Dans cet présentation nous nous intéresserons plus précisément à la fonction sigma et aux suites aliquotes qu'elle permet de définir.

Introduction sur sigma

Définition : La fonction sigma (σ) associe à un nombre entier la somme de ses diviseurs, 1 et n compris. C'est-à-dire, pour tout n $\in$ N, $\sigma(n)=\sum_{d|n} d$. Par exemple : $\sigma(10)=1+2+5+10=18$.

La figure 1 présente le graphe des valeurs de sigma pour tous les nombres plus petits que 500.

Dans certains cas, comme pour les suites aliquotes, on utilise la fonction σ_0 donnée par la somme des diviseurs d'un nombre sans le nombre lui-même : $\sigma_0(n)=\sigma(n)-n$. Par exemple : $\sigma_0(10) = 8$.

Calcul de sigma

Pour calculer sigma, deux méthodes peuvent facilement être envisagées. La première consiste à trouver les diviseurs (en essayant tous les nombres plus petits que le nombre de départ) puis à les additionner.

```
sigma1 := proc(n)
local x, sig;
begin
  sig:=0;
  for x from 1 to n/2 do
    if irem(n,x)=0 then
      sig:=sig+x;
    end_if;
  end_for;
  sig+n;
end:
```

Encadré 1. Programme élémentaire pour le calcul de $\sigma(n)$

Cette méthode s'avère lente d'autant plus que le nombre est grand. Le programme *MuPAD* pour calculer σ de cette manière est donné dans l'encadré 1. Une version moins élémentaire pourrait se contenter d'essayer les diviseurs jusqu'à la racine de n et d'ajouter à *sig* non seulement x, lorsque celui-ci est diviseur de n, mais aussi n/x (et de procéder à un petit ajustement en bout de course lorsque racine de n est un entier).

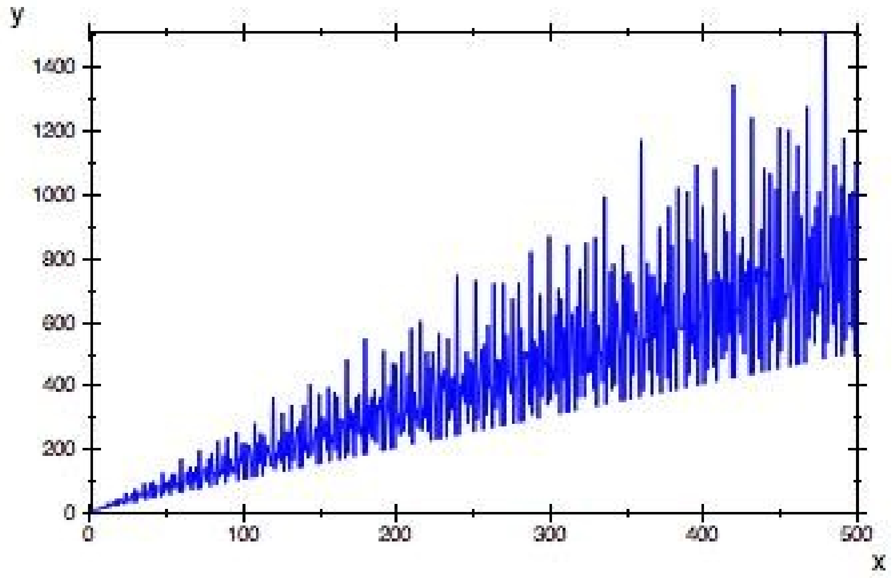

Fig 1. Comportement de σ (axe y) pour n (axe x) jusqu'à 500

La deuxième méthode consiste à utiliser deux propriétés de la fonction σ :

1. σ est une fonction multiplicative : si m et n sont deux nombres premiers entre eux, alors $\sigma(n\cdot m)=\sigma(n)\cdot\sigma(m)$
2. Si p est un nombre premier, alors $\sigma(p^k)=1+p+p^2+\ldots+p^k=\dfrac{p^{k+1}-1}{p-1}$

Démonstration de 1 : Soient n et m deux nombres premiers entre eux. Tous les diviseurs de $n\cdot m$ ont la forme $d=n_i\cdot m_j$ avec n_i diviseur de n et m_j diviseur de m. Alors :

$$\sigma(n\cdot m)=\sum_{i,j} n_i\cdot m_j=\sum_j \left(\sum_i n_i\right)m_j=\sum_j \sigma(n)m_j=\sigma(n)\sum_j m_j=\sigma(n)\cdot\sigma(m)$$

• Démonstration de 2 : les diviseurs de p^k sont $1, p, p^2, \ldots, p^k$. Il s'ensuit que $\sigma(p^k)$ est donné par la série géométrique $1+p+p^2+\ldots+p^k$ de premier terme 1 et de raison p. La somme de cette série vaut : $(p^{k+1}-1)/(p-1)$.

Pour calculer $\sigma(n)$, il suffit donc de factoriser n puis d'appliquer les propriétés 1 et 2. Le programme *MuPAD* qui utilise ces propriétés est donné dans l'encadré 2.

```
sigma2:=proc(n)
local f,s,p,k;
begin
  f:=0;
  s:=1;
  k:=0;
  p:=0;
  f:=op(ifactor(n));
  for p from 2 to nops(f)-1 step 2 do
    k:=(f[p]^(f[p+1]+1)-1)/(f[p]-1);
    s:=s*k;
  end_for;
  s;
end:
```

Encadré 2. Calcul de σ(n) par factorisation en nombres premiers

Pour comparer les deux programmes j'ai noté le temps qu'ils mettent pour calculer σ(12'345'678). Les deux programmes m'ont retourné la bonne solution, c'est à dire 27'319'968, le premier programme en 52 secondes alors que le deuxième en moins d'une seconde. Le passage par la décomposition en facteurs premiers est donc plus efficace que la recherche systématique. Ce fait est en partie dû au fait que MuPAD « connaît » déjà une grande quantité de nombre premiers.

Complexité du calcul de sigma

On a vu que la décomposition en facteur premier permet de calculer « facilement » $\sigma(n)$. Par ailleurs, la connaissance de $\sigma(n)$ permet de trouver « facilement » la décomposition de *n* en facteurs premiers.

En effet, soit $n=p\cdot q$ un produit de deux (grands) nombres premiers. Connaissant *n*, on cherche *p* et *q*. Supposons que l'on connaisse $\sigma(n)$ alors on peut trouver la valeur de *p* et de *q* en résolvant un système de deux équations :

$$n=p\cdot q \quad ; \quad \sigma(n)=\sigma(p\cdot q)=\sigma(p)\cdot\sigma(q)=(p+1)(q+1)=pq+p+q+1$$

qui se réduit en une équation quadratique : $p^2+p(1+n-\sigma(n))+n=0$

Le calcul de sigma est donc intimement lié au calcul de la décomposition en facteurs premiers. Les deux calculs sont de complexité équivalente.

De ce fait, sigma est difficile à calculer puisque, pour le moment, les algorithmes connus de décomposition en facteurs premiers, qui entrent dans le système de cryptographie RSA , demandent des temps de calculs qui augmentent exponentiellement avec la grandeur des données à traiter.

Les suites aliquotes

Définition : Une suite aliquote est une suite (a_n) définie par un terme initial a_1 et la relation suivante : $a_{n+1}=\sigma_0(a_n)$. On la désigne par $A(a_1)$. On représente les suites aliquotes par un graphe avec *n* en abscisse et a_n en ordonnée.

Voici un exemple de suite avec a_1 = 10 : A(10) = [10; 8; 7; 1; 0] . Le graphe de A(10) est donné dans la figure 1.

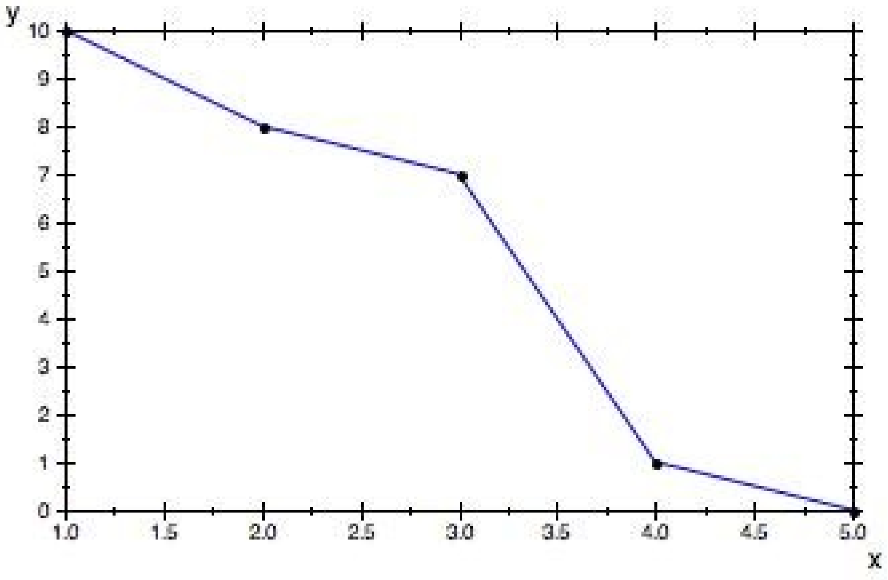

Fig 1. Graphe correspondant à la suite aliquote (10)

A noter que σ n'est pas injective ; un nombre peut être le sigma de plusieurs autres nombres. Par exemple 64 est le sigma de 76 et de 56 : σ(56) =1+2+4+7+8+14+28=64 ; σ(76) =1+2+4+19+38=64.

Ainsi deux suites aliquotes peuvent se « rejoindre », comme nous le montre le graphe de la figure 2 pour les suites A(80) = [80; 106; 56; 64; 63; 41; 1; 0] (en bleu) et A(88) = [88; 92; 76; 64; 63; 41; 1; 0] (en rouge).

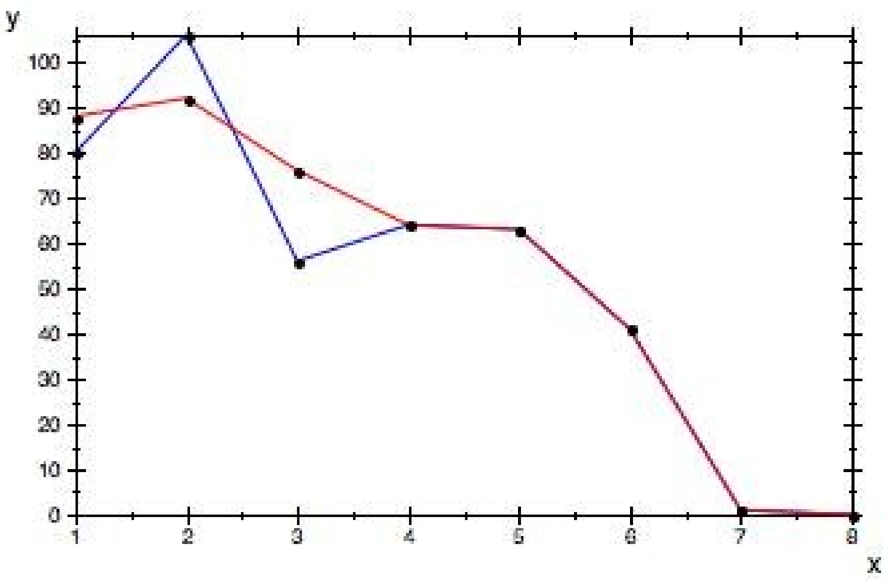

Fig 2. Deux suites aliquotes qui se rejoignent

Abondance et déficience

Définitions : n est abondant si $\sigma_0(n) > n$; n est parfait si $\sigma_0(n) = n$; n est déficient si $\sigma_0(n) < n$.

Statistique : Sur les 200'000 premiers nombres, il y a 49'481 abondants (environ 25%), 4 parfaits et 150'515 déficients (environ 75%).

La figure 3 montre le graphe du comportement de σ_0 sur les 500 premiers nombres. La droite rouge représente la fonction f(x) = x, fonction qui relie tous les nombres parfaits. Celle-ci est aussi la limite entre abondant et déficient. Ce graphe illustre bien le fait qu'il y a plus de déficients que d'abondants.

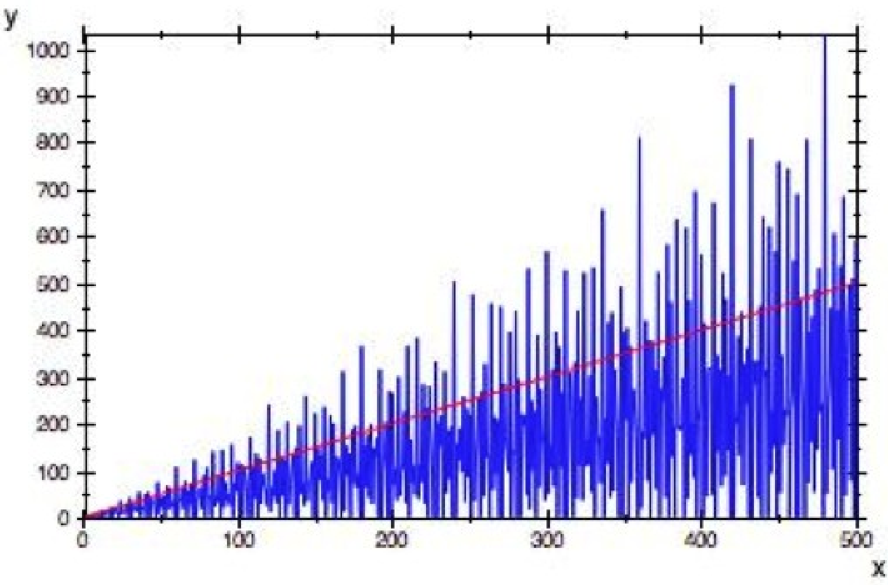

Fig 2. Comportement de σ_0 sur les 500 premiers nombres

Tous les nombres premiers sont déficients car $\sigma_0(p) = 1$. La croissance d'une suite aliquote est donc liée à l'abondance du terme, mais l'abondance de ce terme n'a aucune relation avec l'abondance du prochain terme. Comme nous le montre bien la figure 4 qui a pour nombre de départ 120, où jusqu'au 9e terme la suite est croissante, mais subitement le 9e terme est déficient et la suite décroit.

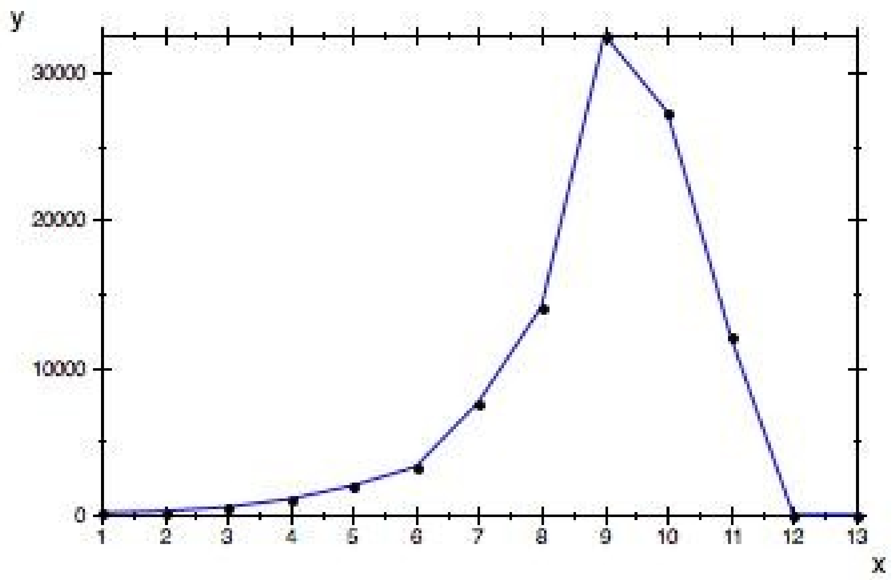

Fig 3. A(120)

Programmes

Quelques programmes ont été utilisés tout au long de ce travail. L'encadré 3 propose celui qui affiche les suites aliquotes (a_n), leur graphe, la suite des logarithmes log(a_{n+1}) + 1 et le graphe de cette suite. Dans ces programmes la fonction sigma utilisée est celle fournie par MuPAD dans la bibliothèque 'numlib'.

```
plotfin:=proc(n)
local L,P,x,y,t,k,j,H,G;
begin
  L:=null(); P:=null(); H:=null(); G:=null();
  y:=n;
  x:=1;
  t:=0;
  j:=float(log(10,y));
  P:=P,[x,y]; H:=H,[x,j]; L:=L,[x,y]; G:=G,[x,j];
  while y<>0 and t<>5 and j<40 do
    t:=0;
    y:=numlib::sigma(y)-y;
    if y<>0 then
      j:=float(log(10,y));
    end_if;
    x:=x+1;
    P:=P,[x,y]; L:=L,[x,y]; H:=H,[x,j]; G:=G,[x,j];
    for k from 1 to nops(P) do
      if [k,y]=P[k] then
        t:=t+1;
      end_if;
    end_for;
  end_while;
  print(P);
  plot(plot::Listplot([L]));
  print(H);
  plot(plot::Listplot([G]));
end:
```

Encadré 3.

L'encadré 4, propose un programme conçu pour calculer toutes les suites aliquotes de premier terme compris entre n et m et sortir toutes les suites spéciales. Son nom est lié au fait qu'il va tourner longtemps !

Il y a deux grandes parties dans ce programme : la première jusqu’au « end_while; » est la partie où le programme calcul la suite aliquote. La seconde partie est là pour classer chaque suite grâce à une suite de tests.

```
longtemps:=proc(n,m)
local L,k,x,r,t,p,c1,c2,c3,c4,c5,c6,c7,C1,C2,C3,C4,C5,C6,C7;
begin
  c1:=0; c2:=0; c3:=0; c4:=0; c5:=0; c6:=0; c7:=0;
  C1:=null(); C2:=null(); C3:=null();   C4:=null(); C5:=null(); C6:=null(); C7:=null();

  for k from n to m do
    // Calcul de la suite aliquote de premier terme k
    x:=k;
    if irem(k,250)=0 then
      print(k);
    end_if;
    r:=0;
    L:=null(); L:=L,x;
    p:=0;
    while x<>0 and r<5 and p<30 do
      x:=numlib::sigma(x)-x;
      p:=float(log(10,x+1))+1;
      L:=L,x;
      for t from 1 to nops(L)-1 do
        if L[t]=x then
          r:=r+1;
        end_if;
      end_for;
    end_while;

    // Classement de la suite calculée
    // La suite se termine par 0
    if x=0 then
      c1:=c1+1; C1:=C1,k;
    end_if;

    // La liste ne semble ne pas se terminer
    if p>=30 then
      c2:=c2+1; C2:=C2,k;
    end_if;

    // Le nombre de départ est un nombre parfait (suite constante)
    if L[1]=L[2] then
      print(Unquoted,"".x." est un nombre parfait");
      c3:=c3+1; C3:=C3,k;
    end_if;

    // La suite se termine par un nombre parfait
    if L[1]<>L[2] and L[nops(L)]=L[nops(L)-1] then
      c6:=c6+1; C6:=C6,k;
    end_if;

    if nops(L)>2 then
      // Le nombre de départ fait partie d'une paire de nombres amicaux
      if L[1]=L[3] and L[1]<>L[2] then

        print(Unquoted, "les nombres ".L[1].",".L[2]. " sont des nombres amicaux ");
        c4:=c4+1; C4:=C4,k;
      else
      // La suite se termine sur une paire de nombres amicaux
        if L[1]<>L[3] and L[1]<>L[2] and L[nops(L)-2]=L[nops(L)] and
           L[nops(L)]<>L[nops(L)-1] then
          c7:=c7+1; C7:=C7,k;
        end_if;
        //  La suite finit par un cycle
        if x<>0 and L[nops(L)]<>L[nops(L)-1] and p<30 and L[nops(L)]<>L[nops(L)-2] then
          print(Unquoted, "il y a une boucle qui commence par ".k."");
          c5:=c5+1; C5:=C5,k;
        end_if;
      end_if;
    end_if;
  end_for;
  print(Unquoted, "                             VOICI LES RESULTATS");
  print(Unquoted, "".c1. " se terminent par zéro ");
  print(Unquoted, "".c2. " ne se terminent pas ",C2);
  print(Unquoted, "".c3. " nombres parfaits ",C3);
  print(Unquoted, "".c4. " amicaux ",C4);
  print(Unquoted, "".c5. " donnent des boucles ",C5);
  print(Unquoted, "".c6." nombres tombent sur des nombres parfaits ",C6);
  print(Unquoted, "".c7." nombres tombent sur des nombres amicaux ",C7);
end:
```

Encadré 4. Calcul et classement de suites aliquotes

Ce programme a fonctionné plus de 26 heures pour calculer les suites jusqu'à A(200'000). Le chapitre suivant, notamment les rubriques « Statistiques », rend compte de ces résultats.

Différentes sortes de suites

Les suites qui se terminent par 0

Derrière, une suite aliquote qui se termine par 0, se cache un nombre premier. En effet, dès qu'une suite aliquote tombe sur un nombre premier son σ_0 sera égal à 1 et fatalement, la suite aura comme prochain terme 0. L'étude de ces suites se résume donc à l'étude de l'apparition des nombres premiers dans les suites aliquotes.

En reprenant le programme « longtemps » et en lui ajoutant un compteur de nombres premiers, il est possible de construire un histogramme du nombre d'apparitions de chaque nombre premier. La figure 4 propose cet histogramme construit pour toutes les suites aliquotes commençant par des nombres jusqu'à 300.

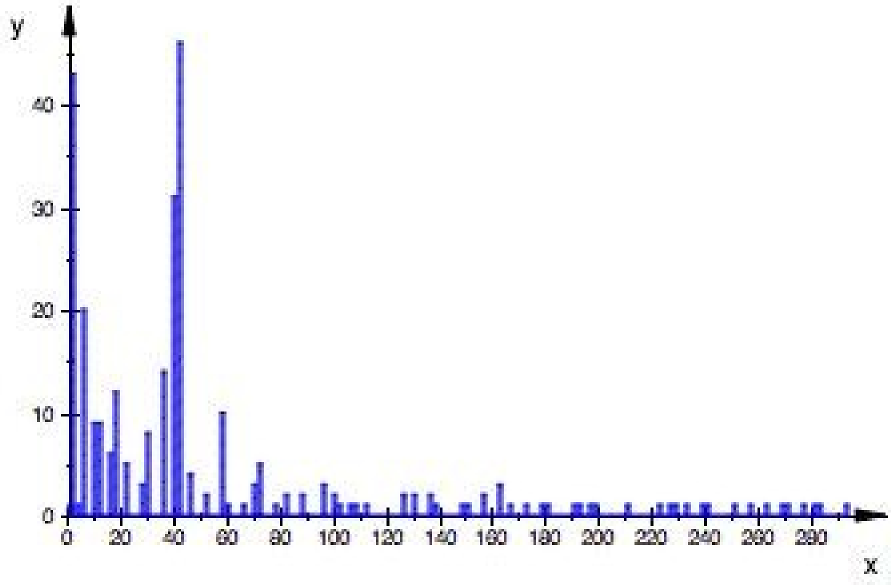

Fig 4. Statistique des nombres premiers apparaissant dans une suite aliquote

Etude de l'histogramme

Les deux nombres premiers les plus représentés sont 43 avec 46 occurences et 3 avec 43 apparitions. Nous n'avons pas d'explication de ce fait peut-être dû au seul hasard.

Statistiques

Sur 200'000 suites, il y en a 166'622 qui se terminent par zéro, c'est donc une grande majorité, cela représente environ 83%,

Les suites qui ne se terminent pas

Ce sont les suites sans boucles identifiées par le programme comme augmentant « beaucoup ». De fait, on ne sait pas si elles se terminent ou pas. Le problème est que plus un nombre est grand, plus cela va prendre du temps pour calculer son sigma, et comme ces suites ne semblent pas redescendre, on peut faire l'hypothèse qu'elles tendent vers l'infini. Le premier nombre qui engendre une suite de ce type est 276.

Le graphe de sa suite est donné dans la figure 5. Celui-ci a la forme d'une exponentielle. Pour mieux voir la progression, la figure 6 présente le graphe de $\log(a_{n+1}) + 1$ pour la suite aliquote (a_n). Cette formule donne la taille du nombre, c'est-à-dire le nombre de chiffres dont il est composé.

Ce graphe, qui a demandé 34 heures à l'ordinateur, semble montrer que la suite continue d'augmenter. Va-t-elle retomber à zéro ? Impossible de la savoir à l'aide d'un calcul sur ordinateur !

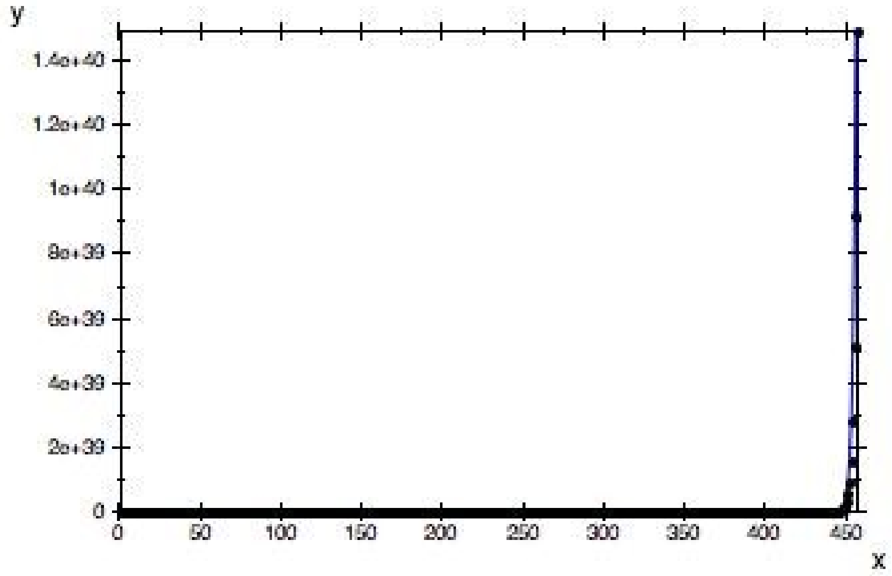

Fig 5. A(276)

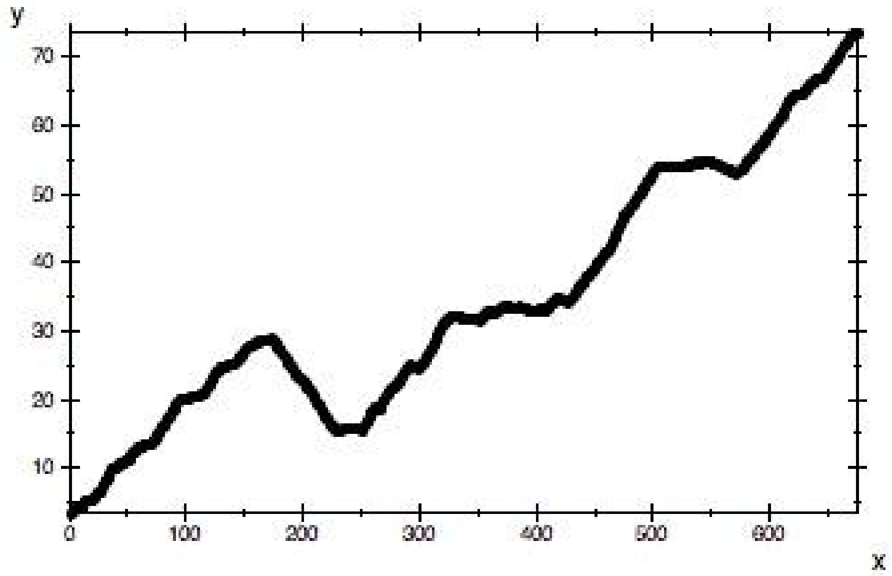

Fig 6. Graphe en log de la suite A(276)

Parmi les nombres inférieurs à 1000 le programme en identifie 13 qui ne se terminerait pas. Ces suites commencent par : [276; 306; 396; 696]; [552; 888]; [564; 780; 996]; [660; 828]; 840; 966.

Les nombres regroupés par une accolade font partie de la même suite. Soit ils la rejoignent ou ils en font partie. Par exemple A(306) rejoint A(276) à son deuxième terme. Parmi ces nombres, on identifie les cinq de Lehmer : 276; 552; 564; 660; 966 qui font partie des candidats de suites infinies apériodiques.

Quant à la suite A(840), elle se termine après un pic de 49 chiffres. Son graphe en « log » est donné dans la figure 7.

La suite A(3'630) pose également problème au programme. C'est une suite finie[1] mais dont les termes deviennent trop grand pour mon ordinateur (après 3 jours de calcul). Son graphe en « log » jusqu'au 1'037e terme est donné dans la figure 8.

[1] http://www.factordb.com/sequences.php

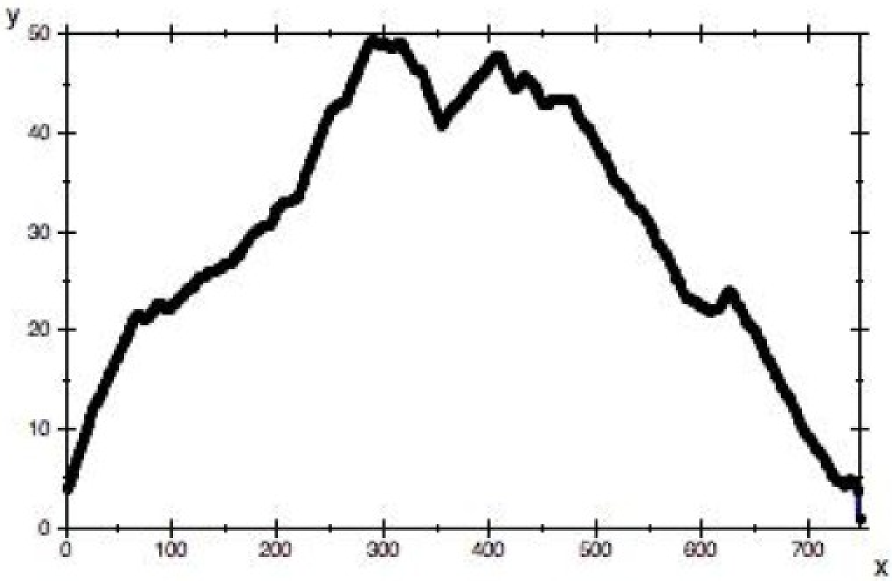

Fig 7. Graphe en log de la suite A(840)

De ces résultats, j'en déduit que certainement une partie des suites que j'ai qualifiées de montant à l'infini finissent, peut-être toutes. On se trouve ici face au mystère et la question principale concernant les suites aliquotes.

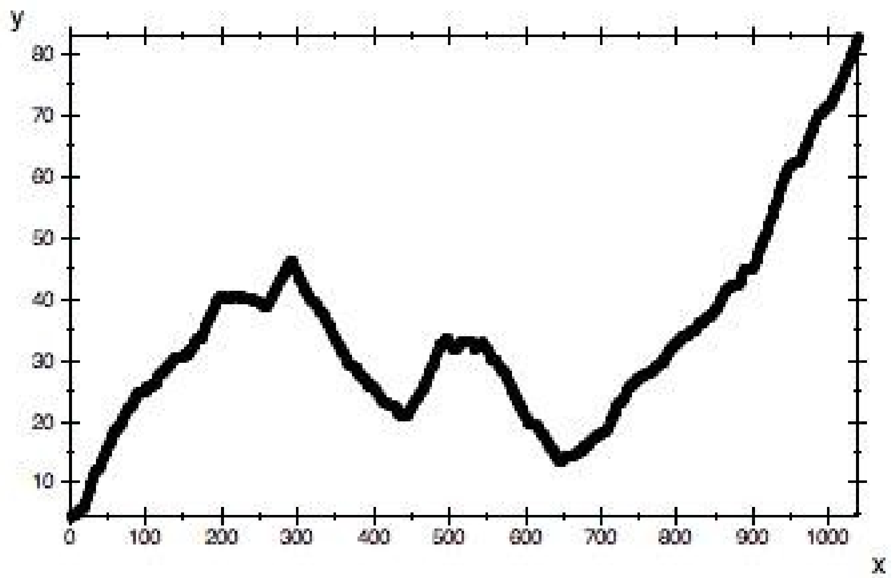

Fig 8. Graphe en log de la suite A(3630) jusqu'au terme 1037

Statistiques

Le programme trouve 29'166 suites qui ne se terminent pas (infinie apériodique), ce qui représente 14,6% des suites. C'est la deuxième sorte de suites la plus représentée.

Les nombres parfaits

Définition : Un nombre parfait N est un nombre tel que $\sigma_0(N) = N$, c'est-à-dire que la somme de ses diviseurs stricts est égale à lui-même. La suite et donc le graphe de sa suite seront constants. Les quatre premiers nombres parfaits sont : 6, 28, 496, 8128.

Recherche des nombres parfaits

Euclide a trouvé une méthode pour trouver des nombres parfaits pairs. Euler a démontré que cette méthode donne tous les nombres parfaits pairs. Cette méthode utilise les nombres dits de Mersenne qui sont les nombres premiers de la forme $M_p = 2^p - 1$ avec p premier. Avec cette notation, on peut montrer que $N = 2^{p-1} \cdot M_p$ est parfait.

Exemple avec p = 2 : $M_2 = 2^2 - 1 = 3$; 3 étant premier nous obtenons alors un nombre parfait pair : $N = 2^{2-1} \cdot M_2 = 2 \cdot 3 = 6$

Démonstration : Nous allons démontrer que $\sigma(N) = 2\mathrm{N}$.

Tout d'abord $\sigma(N) = \sigma(2^{p-1} \cdot M_p) = \sigma(2^{p-1}) \cdot \sigma(M_p)$ en utilisant la première propriété de sigma, la multiplicativité, puisque les deux facteurs sont premiers entre eux (M_p est impair).

Puis par le deuxième propriété on a : $\sigma(2^{p-1}) = 2^p - 1$ et $\sigma(M_p) = 1 + M_p = 2^p$

En définitive : $\sigma(N) = \sigma(2^{p-1}) \cdot \sigma(M_p) = (2^p - 1) \cdot 2^p = 2 \cdot (2^{p-1} \cdot (2^p - 1)) = 2 \cdot (2^{p-1} \cdot M_p) = 2\mathrm{N}$

Par ailleurs, Euler a prouvé qu'un nombre parfait pair est obligatoirement du type : $n = 2^{p-1} \cdot M_p$

Le programme de l'encadré 5, recherche les *n* premiers nombres parfait. Dans un temps il recherche les nombres de Mersenne. Il utilise ensuite la propriété découverte par Euclide pour former un nombre parfait pair. A noter que MyPAD offre les fonctions ithprime(i) (i-ème nombre premier) et isprime(p) !

```
parfait:=proc(n)
local h,p,m,x,k;
begin k:=0;
  h:=1; p:=1; m:=1; x:=1;
  while k<>n do
    h:=ithprime(p);
    m:=(2^p)-1;
    p:=p+1;
    if isprime(m)=TRUE then
      x:=(m*(m+1))/2;
      k:=k+1;
      print(x);
    end_if;
  end_while;
end:
```

Encadré 5. Recherche de nombres parfaits

Statistiques

Les nombres parfaits sont très peu représentés, seulement 4 nombres pour 200'000 testés (0.002%). Aujourd'hui on en connaît une quarantaine et comme les nombres de Mersenne semblent se présenter avec une certaine régularité, les mathématiciens font la conjecture de l'existence d'une infinité de nombres parfaits. On note qu'aucun nombre parfait impair n'est apparu. Leur existence est une question ouverte.

Les nombres amicaux

Définition : Deux nombres *N* et *M* sont dits amicaux si $\sigma_0(N) = M$ et $\sigma_0(M) = N$.

Exemples : (220, 284) ; (1184, 1210) sont des paires de nombres amicaux. La figure 9 propose le graphe pour la paire (220, 284).

Recherche de nombres amicaux

Il n'y a pas de méthode pour trouver tous les nombres amicaux comme c'est le cas pour les nombres parfaits pairs. Mais il existe des « recettes de cuisine » pour pouvoir trouver quelques paires. En voici une, c'est la méthode de Abu-I-Hasan Thabit ibn Querra :

Soient $p = (3 \cdot 2^{n-1}) - 1$; $q = (3 \cdot 2^n) - 1$; $r = (9 \cdot 2^{2n-1}) - 1$

Si p, q et r sont premiers alors : $M = 2^n \cdot p \cdot q$ et $N = 2^n \cdot r$ sont deux nombres amicaux.

Exemple pour n = 2 : p = 5 ; q = 11 ; r = 71 ; M = 220 ; N = 284

Cette méthode nous donne effectivement des nombres amicaux mais elle ne nous les donne pas tous. La paire suivante obtenue par cette méthode est pour n = 4 : M = 18416 et N = 17296. Cette paire n'est pas la 3e, elle est la 8e.

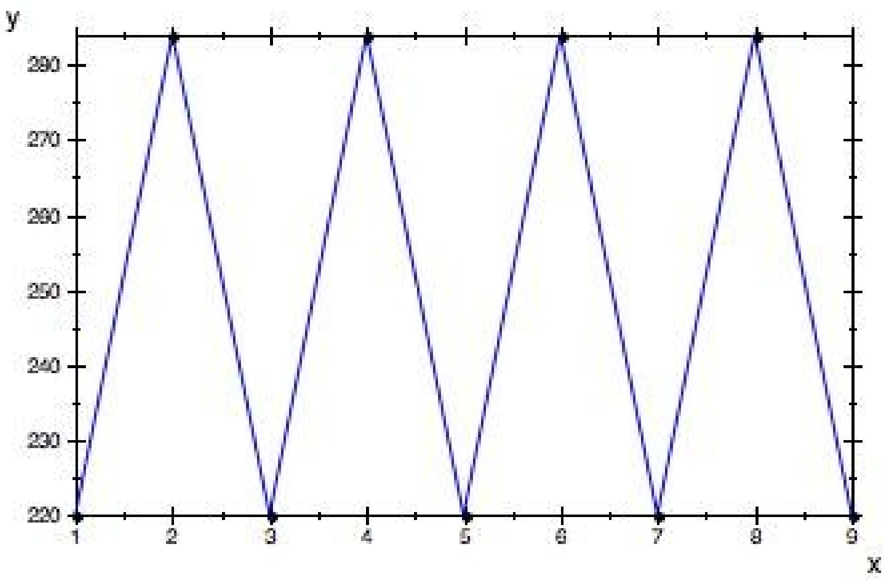

Fig 9. A(220)

Statistiques

Il y a 42 nombres amicaux jusqu'à 200'000, c'est-à-dire 21 paires. Ces nombres sont donc relativement rares (0.02%).

Les boucles

Définition : Une boucle est une suite de nombres dits sociables. C'est-à-dire que le premier nombre sociable est le sigma du dernier. Les suites de nombres sociables sont définies par un des nombres sociables et par leur ordre, c'est-à-dire du nombre de maillons qu'il faut parcourir si l'on veut retomber sur le premier nombre sociable. Les nombres parfaits sont sociables d'ordre 1 et les nombres amicaux sociables d'ordre 2. En formule, une suite d'ordre *n* qui commence par a_1 est donnée par :

$\sigma_0(a_1) = a_2$; $\sigma_0(a_2) = a_3$; ... ; $\sigma_0(a_{n-1}) = a_n$; $\sigma_0(a_n) = a_1$

Exemple d'une suite sociable d'ordre 5 : **12'496 ;** 14'288 ; 15'472 ; 14'535 ; 14'264 ; **12'496**

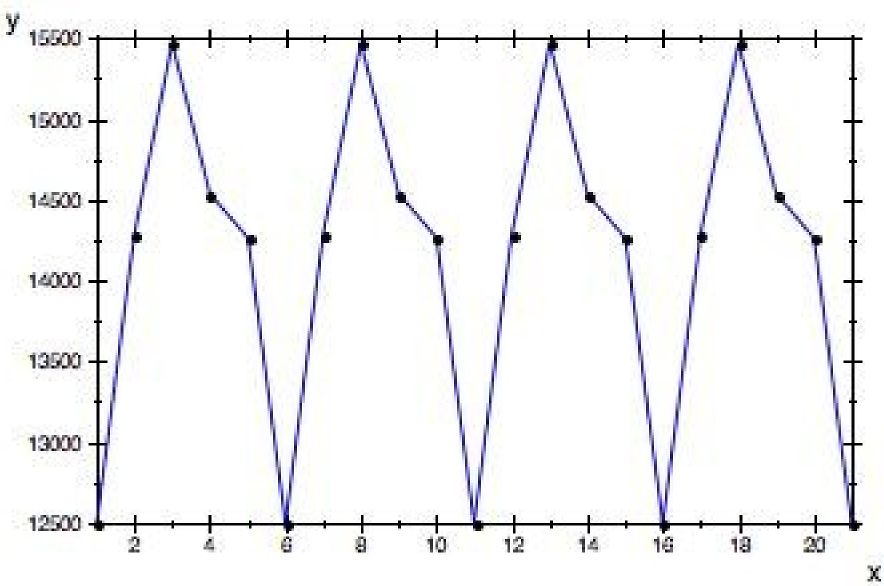

Fig 10. A(12496)

Statistiques

Il n'y a que 33 nombres sociables plus petits que 200'000, ce qui équivaut à environ 0.016%. Mais ce qui est encore plus étonnant c'est qu'ils se répartissent en trois boucles :

1) La boucle d'ordre 5 que nous avons prise comme premier exemple : 12496, 14288, 15472, 14536, 14264 (figure 10).

2) Une boucle d'ordre 4 : 1264460, 1547860, 1727636, 1305184 (figure 11)

3) Une boucle d'ordre 28 : 14316, 19116, 31704, 47616, 83328, 177792, 295488, 629072, 589786, 294896, 358336, 418904, 366556, 274924, 275444, 243760, 376736, 381028, 285778, 152990, 122410, 97946, 48976, 45946, 22976, 22744, 19916, 17716 (figure 12).

Cette boucle est d'ailleurs la boucle la plus longue connue à ce jour, et les chercheurs n'ont trouvé que cette boucle qui est d'un ordre plus grand que 9. Aujourd'hui aucune boucle d'ordre 3 n'a été trouvée, mais beaucoup d'ordre 4.

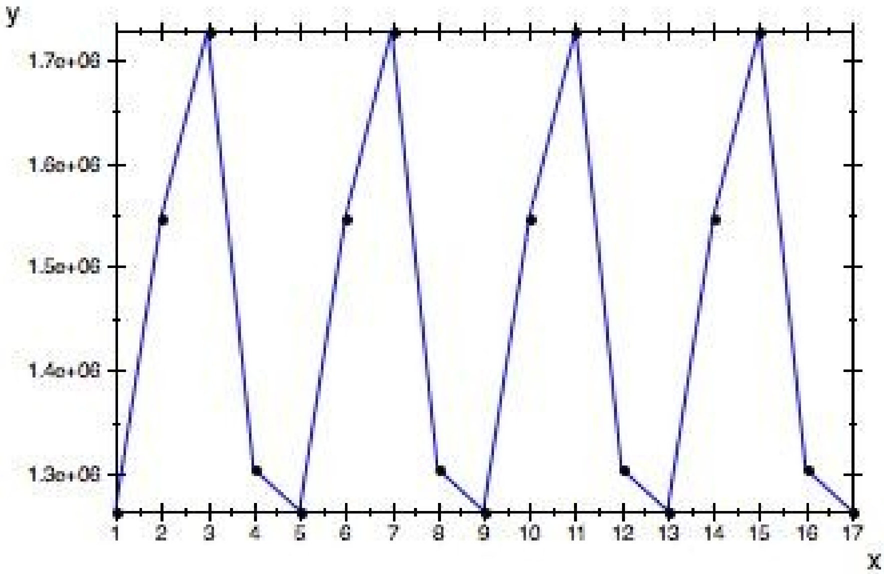

Fig 11. A(1264460)

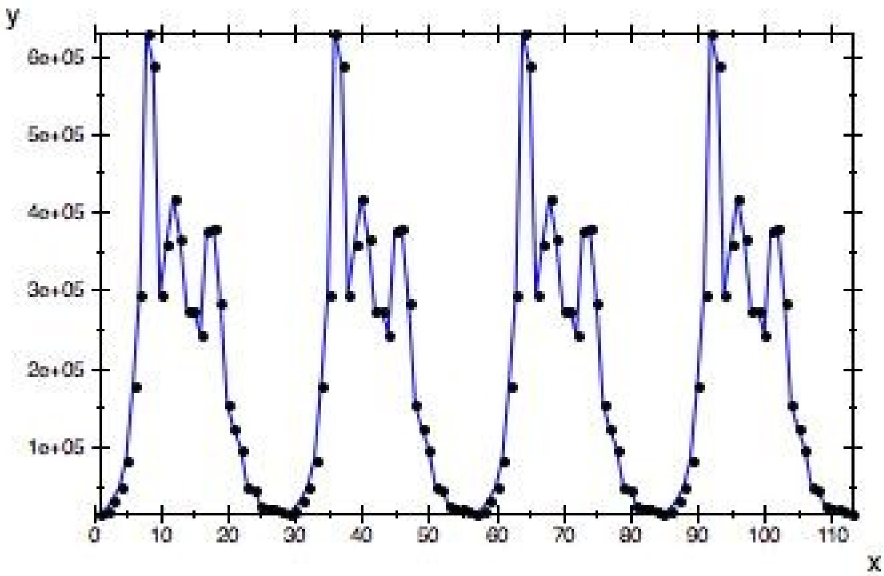

Fig 12. A(1436)

Les suites qui se terminent par un cycle

Définition : Ces suites se terminent par une boucle, d'ordre 1 (nombres parfaits), 2 (nombres amicaux) ou plus grand (boucles). Les nombres qui engendrent ces suites n'entrent dans aucune catégorie particulière.

Exemple : La figure 13 propose le graphe en « log » de la suite commençant par 2'856 qui se finit par la boucle d'ordre 28.

Statistiques

Il y a 4'133 suites qui se terminent par un cycle, dont 1'616 par un nombre parfait, 2'295 par une paire de nombres amicaux et 222 qui se terminent par des boucles. Ce qui représente 2.07% des suites, ce qui est plus que les nombres amicaux, sociables et parfaits réunis.

A noter qu'aucune de ces suite ne tombe sur le nombre parfait 28. A(28) est une « chaîne aliquote isolée ».

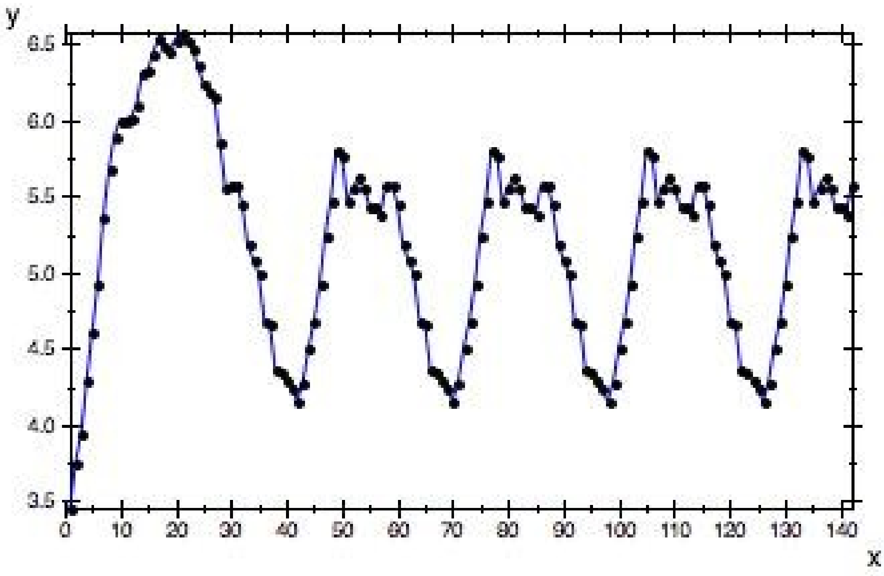

Fig 13. A(2856)

Conclusions

Dans ce travail, j'ai abordé les suites aliquotes et leurs spécificités. J'ai remarqué que l'ordinateur est un élément qui devient presque indispensable dans cette étude par sa vitesse de calcul et ses possibilités graphiques. Mais il a aussi ses limites !

Il est aussi intéressant de constater que l'on repère rapidement des problèmes qui restent encore ouverts. Existe-il :

- Une loi pour les nombres premiers qui finissent les suites ?
- Une suite aliquote qui ne se termine pas ?
- Un nombre parfait impair ?
- Une méthode pour trouver tous les nombres amicaux ?

Remerciements : Je tiens avant tout à remercier mon professeur de mathématiques et mentor Monsieur J.-B. Mathey pour ses conseils et son aide. J'ai ensuite une pensée pour ma famille qui m'a apporté beaucoup de soutien, particulièrement mon frère qui a même relevé des résultats de mon ordinateur en plein milieu de la nuit, en rentrant de soirée. Je veux aussi remercier mon professeur d'application des maths, Monsieur O. Jeanneret qui m'a appris l'utilisation de *MuPAD,* et pour finir un grand merci à Monsieur J.-L. Garambois pour ses réponses et ses encouragements.

Bibliographie, webographie et programmes informatiques

Bibliographie

BARUK Stella : *Dictionnaire de Mathématiques élémentaires,* Seuil. coll. Science ouverte, 1992

DALEHAYE Jean-Paul : Nombres amiables et suites aliquotes, *Pour la science,* n°292, Février 2002, p.98-103. Aussi disponible sur internet : http://www.lifl.fr/~lasou/UEs/info154/JPD/PLSSuitesAlicotes.pdf (consulté : novembre 2012)

ORE Oystein : *Number Theory and its History,* McGraw-Hill, 1948, chap. 5.

Webographie

GARAMBOIS Jean-Luc, les suites aliquotes. http://www.aliquotes.com/ (consulté le 20.1.12)

Wikipédia : http://fr.wikipedia.org/wiki/ (consulté le 20.1.12) pour : Nombre parfait, Nombre amical, Nombre sociable, Suite aliquote.

Nombres - curiosité, théorie et usage, les Nombres parfaits. http://villemin.gerard.free.fr/Wwwgvmm/Decompos/Parfdemo.htm (consulté le 20.1.12)

Site de calcul de suites aliquotes. http://www.factordb.com/sequences.php (consulté le 20.1.12)

Programmes

MuPAD pro, programme de calcul et de programmation. Edition 4.0. Fourni par le lycée avec la licence.
