SUITES AVEC MathsOntologie

I/ Approche mathsontologique des suites

1) Fonctions

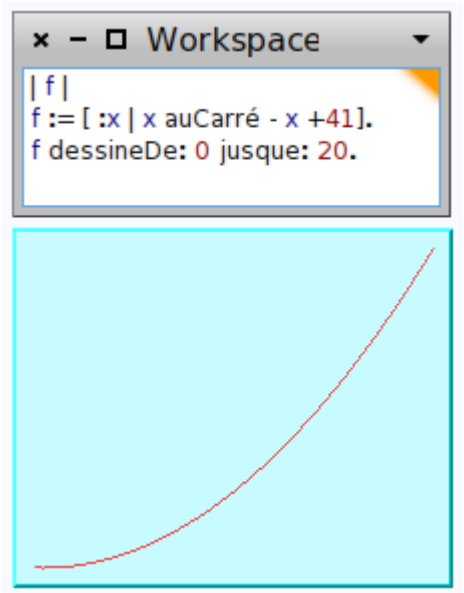
Une suite est une fonction définie sur $\mathbb{N}$. Par exemple, on mesure une grandeur (PIB, capital, population d'un pays, relevé de températures etc) à intervalles réguliers, et on ne connaît donc la valeur de la fonction qu'à ces intervalles, indexés par des entiers. Lorsque la fonction échantillonnée est numérique, on dira que la suite est numérique. Pour approcher la notion de suites avec MathsOntologie, on va donc commencer par décrire la syntaxe des fonctions en MathsOntologie :

Une fonction est décrite par un « bloc » de Smalltalk[1], dont la syntaxe obéit aux règles suivantes :

- La fonction est entre crochets ;
- la liste des variables est séparée du corps de la fonction par un trait vertical[2] « | » ;
- le nom de l'antécédent est précédé d'un double-point (:x au lieu de x pour expliquer à MathsOntologie qu'il ne faut pas calculer x...) ;
- Chaque instruction est suivie d'un point qui la termine ;
- la dernière instruction est la valeur renvoyée par la fonction

Enfin, si une fonction s'appelle f, on obtient l'image de 2 par `f value: 2`

Par exemple, pour définir la fonction $f(x)=x^2-x+41$, on fait ainsi :



La première ligne donne, entre traits verticaux, la liste des variables utilisées dans le script ; ici il n'y en a qu'une, c'est f. La seconde ligne est la définition de f comme décrite précédemment. Et comme f est une fonction numérique, MathsOntologie peut esquisser sa représentation graphique sur [0;20] ce qui est fait dans la troisième ligne. On voit alors un arc de parabole car la fonction f est un trinôme.

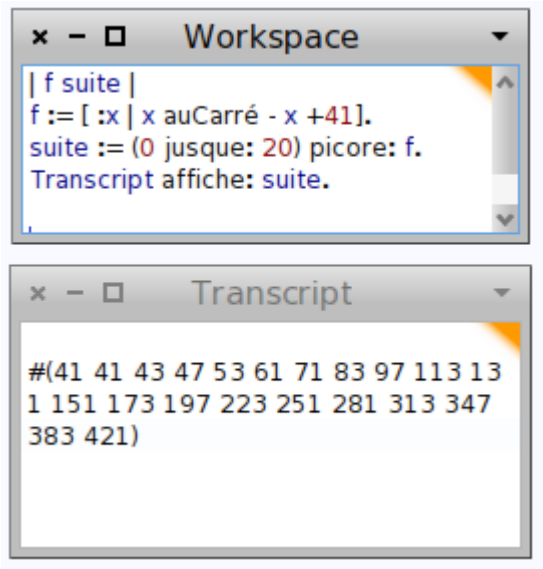
---

1 Voir page 62 de « Pharo par l'exemple » : http://pharobyexample.org/fr/

2 Sur le clavier de l'ordinateur, faire « AltGr » et, simultanément, appuyer sur le 6 du haut du clavier

2) Suites définies explicitement

Pour passer de la fonction à une suite, il suffit de remplacer x réel par n entier et noter $u_n = n^2 - n + 41$. MathsOntologie utilise une image mentale quelque peu originale : Celle d'un oiseau (la fonction) capable de digérer tout réel en le transformant par la fonction, mais qui ne picore que les entiers. On définit alors un nouvel objet appelé suite, et on l'affecte en demandant à la fonction de picorer le tableau des entiers allant de 0 à 20, ce qui crée alors un tableau[3] d'entiers :



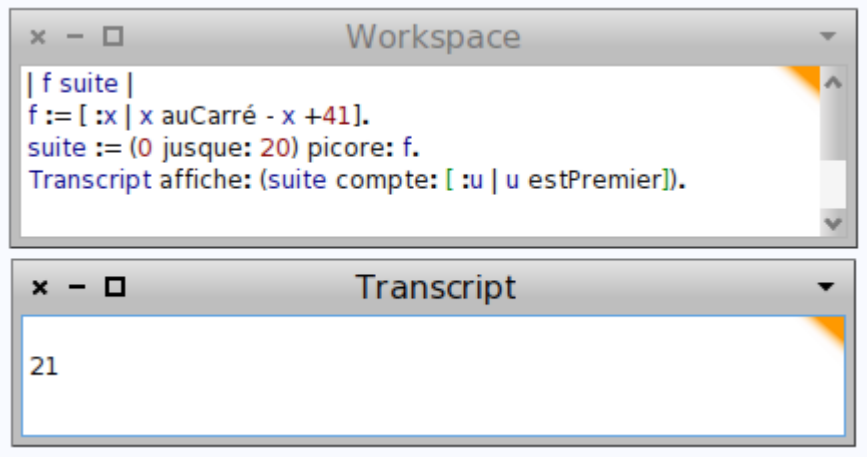
La suite obtenue, découverte par Leonhard Euler en 1772, est intéressante pour qui connaît un peu d'arithmétique, en particulier cette définition[4] du même Euler :

> *Les nombres premiers se distinguent des autres nombres, en ce qu'ils n'admettent d'autres diviseurs que l'unité et eux-mêmes. Ainsi 7 est un nombre premier, parce qu'il n'est divisible que par l'unité et soi-même. Les autres nombres qui ont, outre l'unité et eux-mêmes, encore d'autres diviseurs, sont nommés composés: comme par exemple le nombre 15, qui, outre l'unité et soi-même, est divisible par 3 et par 5. Donc en général, si le nombre p est premier, il ne sera divisible que par 1 et par p: mais si p est un nombre composé, il aura, outre 1 et p, encore d'autres diviseurs ;*

3 Le verbe « picorer » est le résultat d'une traduction maladroite de « collect » qui en réalité, représente la construction d'une collection de nombres. Le résultat obtenu est donc un tableau de nombres, noté par un « # » suivi des éléments entre parenthèses séparés par des espaces. On va beaucoup utiliser ces tableaux dans cet article.

4 Extraite de « *découverte d'une loi tout extraordinaire des nombres, par rapport à la somme de leurs diviseurs* », publié en 1751, et lisible ici : http://math.dartmouth.edu/~euler/docs/originals/E175.pdf ; cet article, portant sur une suite aujourd'hui notée $\sigma_n = \sum_{d|n} d$ (somme des diviseurs de l'indice) et notée alors $\int n$, est une « tout extraordinaire » mine d'information sur les suites entières et la manière de les étudier. Il jouera donc un rôle central dans le présent article.

En essayant de trouver les diviseurs des termes de cette suite, on constate que lesdits termes sont tous premiers. On peut vérifier en demandant, via MathsOntologie, à chacun des termes de la suite s'il est vraiment premier :



En faisant l'appel avec *compte*, on trouve 21 nombres premiers sur 21 termes de la suite, et on arrive à une conjecture qui avait amené Euler à s'intéresser à cette suite : Tous les termes de cette suite sont premiers. Ceci dit, si on répète l'expérience en allant jusqu'à 100 au lieu de 20, on trouve seulement 87 nombres premiers au lieu de 101, ce qui veut dire que la conjecture est fausse. D'ailleurs le terme d'indice 41 étant le carré de 41, ne peut être premier.

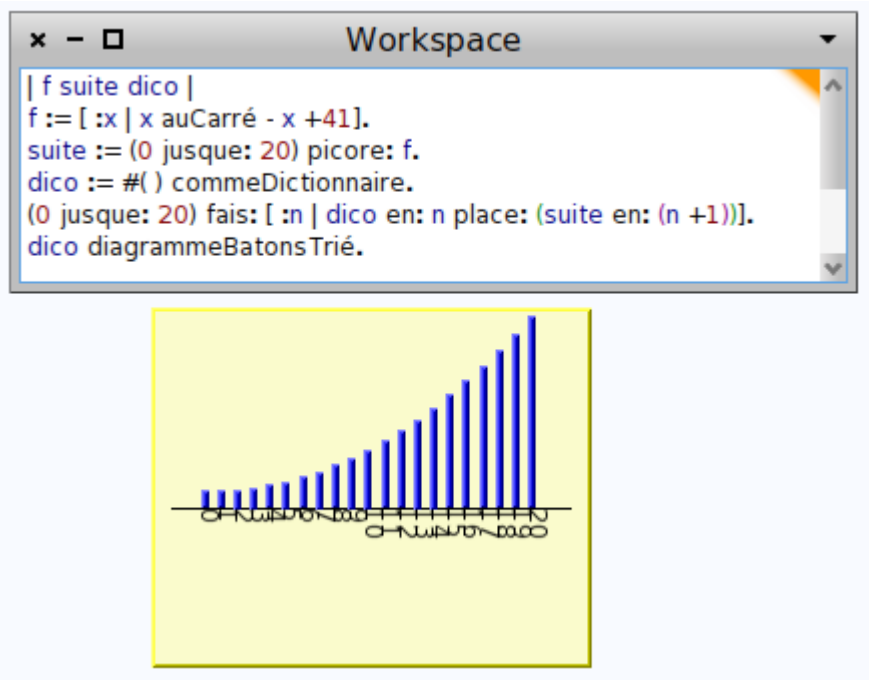
Cette histoire montre qu'une suite comporte une infinité de termes, et qu'il faut se méfier des conjectures hasardeuses...

La représentation d'une suite[5] dans MathsOntologie est donc basée sur un tableau de nombres, les termes successifs de la suite. Cela ne permet pas de représenter graphiquement la suite[6] ; pour cela on aura donc besoin d'un nouvel objet, qui associe chaque valeur de l'indice au terme correspondant de la suite. Un tel objet s'appelle un *dictionnaire*, et le meilleur moyen d'avoir un dictionnaire dans MathsOntologie est de créer un tableau vide puis le convertir en dictionnaire. Ensuite, pour chaque valeur de l'indice, on place dans le dictionnaire, en l'indice, le terme de la suite[7] :

5 En fait, on ne peut représenter en machine qu'une partie finie de la suite, puisque la mémoire de la machine est finie. L'infini des suites est donc un infini potentiel. Mais on verra plus bas que dans certains cas, on peut traiter l'infini comme un nombre avec MathsOntologie.

6 MathsOntologie ne possède pas d'outil de représentation graphique des suites ; toutefois il permet de dessiner des diagrammes en bâtons, et pour les suites positives, cet outil peut être détourné de sa fonction primitive pour représenter la suite. Cela s'applique donc au moins aux suites entières, mais aussi à la plupart des suites figurant dans les sujets du bac ES

7 MathsOntologie commençant à compter à partir de 1 et non de 0, on est obligé de lire le terme en n+1 pour avoir le terme d'indice n. En bref, MathsOntologie est plus adapté à l'étude des suites définies à partir de 1. Sauf si on passe directement par un dictionnaire, comme on le fera avec les suites récurrentes.

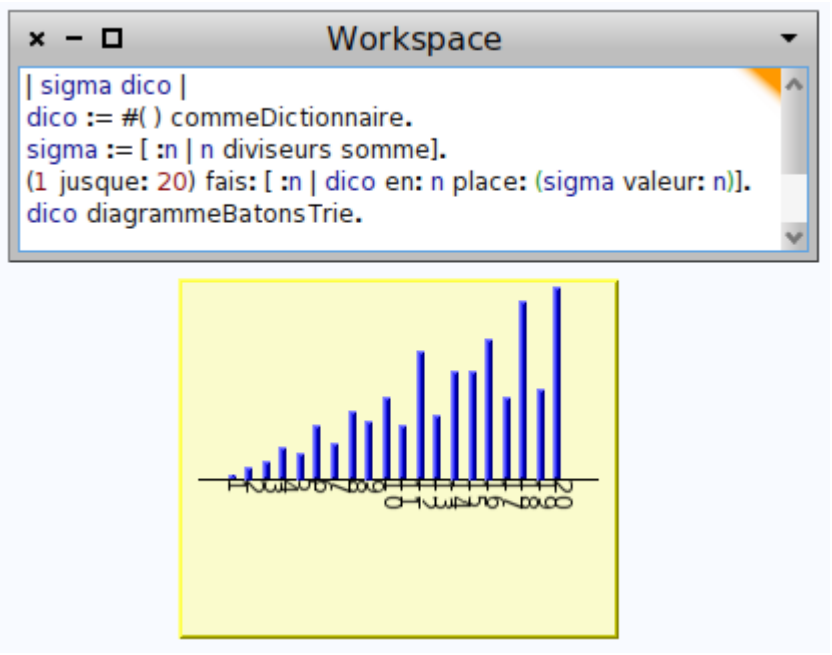


La représentation graphique suggère que la suite tend vers l'infini, mais ceci est une autre histoire qui sera abordée plus bas.

Une autre suite entière étudiée par Euler est celle qui, à tout entier naturel non nul, associe la somme de ses diviseurs. Elle présente un aspect chaotique, à en croire Euler :

> *je viens de découvrir une loi fort bizarre parmi les sommes des diviseurs des nombres naturels, sommes qui, au premier coup d'œil, paraissent aussi irrégulières que la progression des nombres premiers, et qui semblent même envelopper celle-ci.*

La représentation graphique confirme cet aspect chaotique :



En fait, puisque

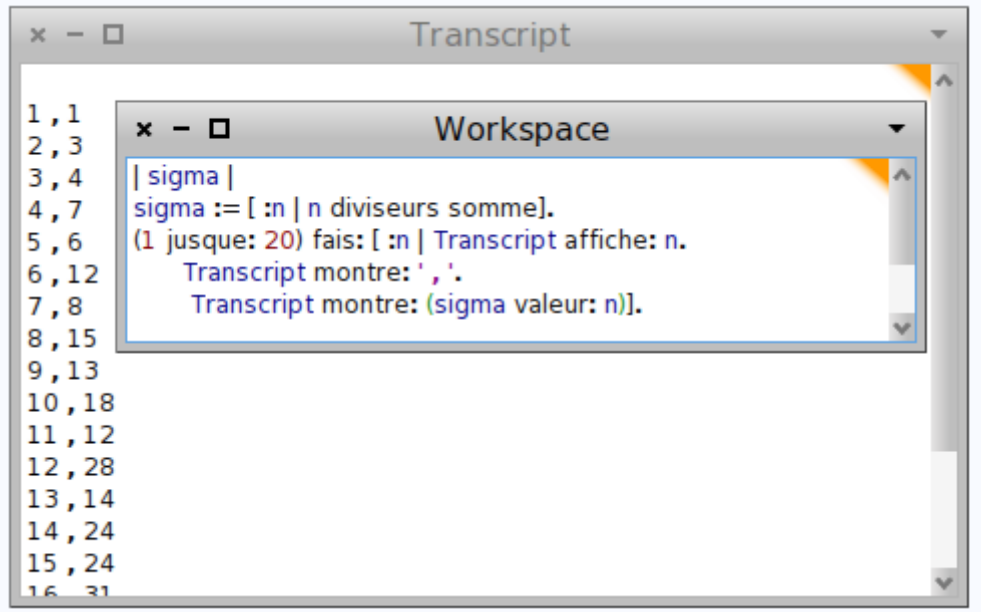
- n est un entier
- n diviseurs est une liste d'entiers (les diviseurs de n)
- n diviseurs somme est un entier (la somme des éléments de la liste ci-dessus)

la fonction n'est définie que sur les entiers (la liste des diviseurs de 3,5 n'existe pas). L'aspect chaotique est confirmé par le tableau des premières valeurs :

indice	somme des diviseurs
1	1
2	3
3	4
4	7
5	6
6	12
7	8
8	15
9	13
10	18
11	12
12	28
13	14
14	24
15	24
16	31
17	18
18	39
19	20
20	42

Voici comment a été fait ce tableau :

Tout d'abord, le script suivant a été écrit puis lancé dans MathsOntologie :

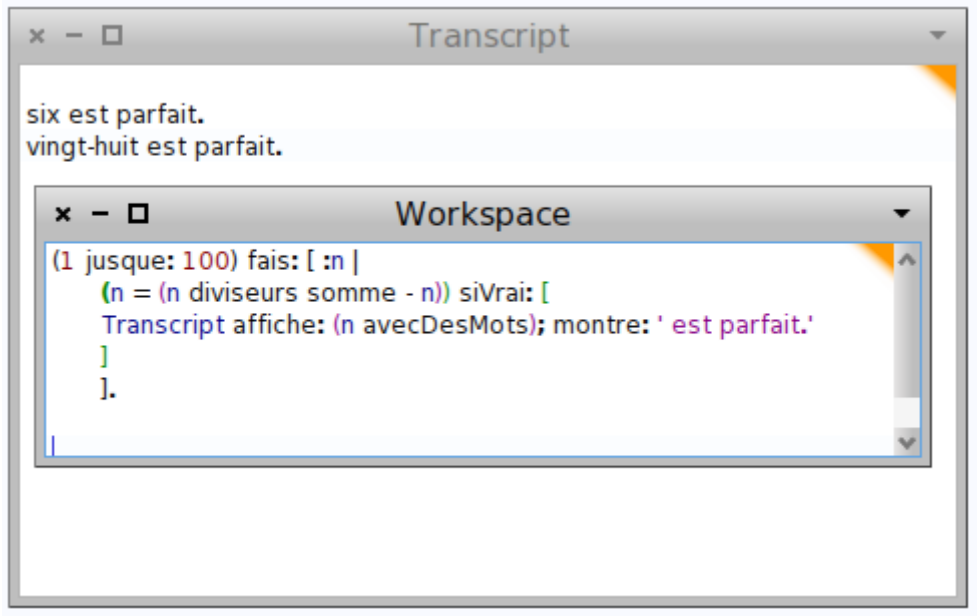


Ensuite, le contenu du transcript a été copié puis collé dans un éditeur de texte ; puis le résultat a été sauvegardé avec l'extension « csv ».

Enfin, le fichier csv obtenu a été ouvert avec Libre Office Calc (tableur) puis, après quelques modifications cosmétiques, sauvegardé avec l'extension « ods ». Ce qui a permis de l'incorporer dans ce document. Une variante permet de représenter graphiquement la suite avec le tableur (comme un nuage de points).

On y voit alors l'aspect globalement croissant mais localement chaotique de la suite, ce qui amène alors des définitions qui étaient classiques lorsqu'on enseignait l'arithmétique en Seconde :

Selon que $\sigma_n - n$ est plus grand, plus petit ou égal à n, celui-ci est dit *abondant*[8], *déficient*[9] ou *parfait*[10]. Les nombres parfaits sont assez rares, puisque par exemple jusqu'à 100 il n'y en a que deux :



Ceci dit, bien que la suite $(\sigma_n)_{n\geqslant 1}$ ait l'air chaotique, Euler a découvert, non pas une formule permettant de calculer son terme général à partir de l'indice, mais de calculer le terme général à partir des termes précédents :

> *Néanmoins j'ai remarqué, que cette progression suit une loi bien régulière et qu'elle est même comprise dans l'ordre des progressions que les géomètres appellent récurrentes, de sorte qu'on peut toujours former chacun des termes par quelques-uns des précédents, suivant une règle constante.*

Ce sont donc les suites récurrentes, où chaque terme dépend explicitement du (ou des) précédent(s), qui sont les plus intéressantes, à la fois comme suites, et comme objets algorithmiques. On les construit par des boucles du style (1 jusque: 20) ou [n<20] tantQueVrai ou [n>=20] jusqueVrai.

---

8 http://fr.wikipedia.org/wiki/Nombre_abondant
9 http://fr.wikipedia.org/wiki/Nombre_d%C3%A9ficient avec le cas extrême des nombres premiers
10 http://fr.wikipedia.org/wiki/Nombre_parfait

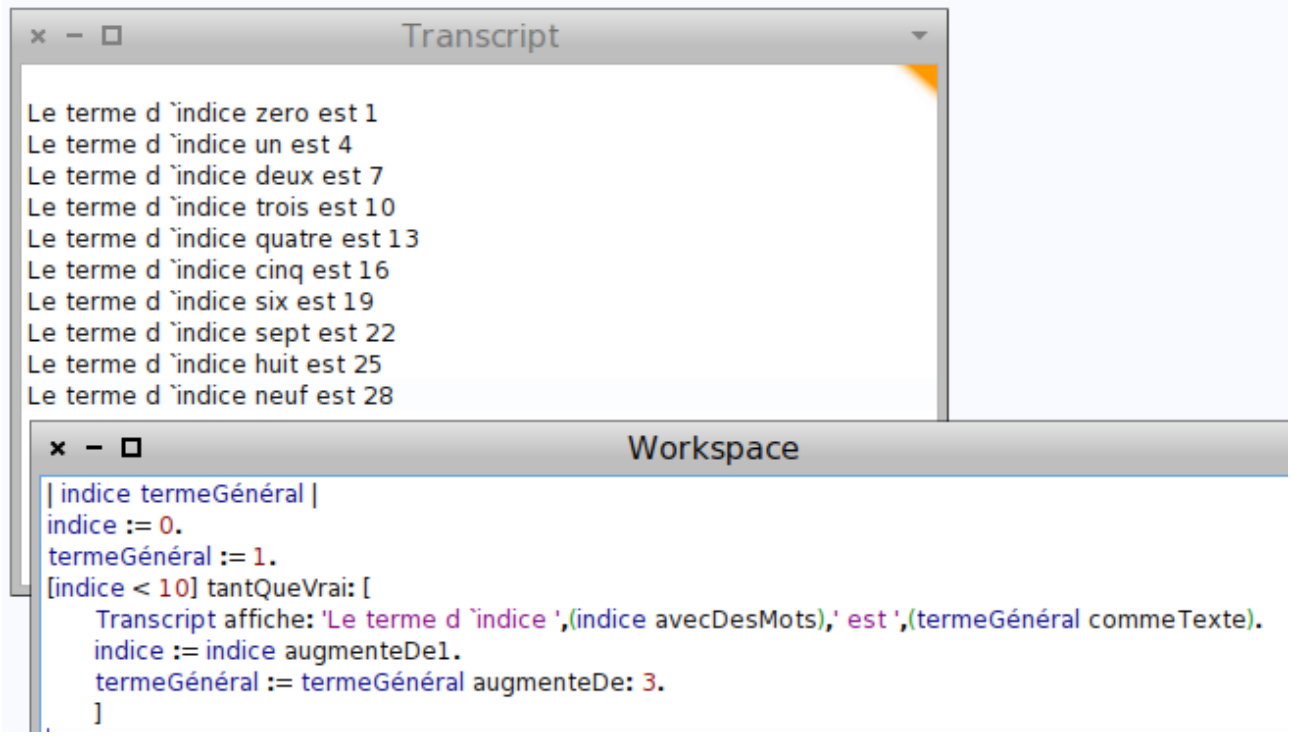
3) Suites arithmétiques

Un premier exemple de suites récurrentes est formé par les suites arithmétiques[11], que l'on peut définir comme cas particulier des suites précédentes (définies par fonction) où la fonction est affine. Alors

- ce qui s'appelait « ordonnée à l'origine » lorsqu'on parlait de fonction, devient « premier terme » de la suite ;
- ce qui s'appelait « coefficient directeur » dans le langage des fonctions devient « raison[12] de la suite ».

Des exemples de suites entières de nature arithmétique reviendront plus tard dans cet article :

- la suite de premier terme 0 et de raison 1 est la suite des entiers naturels[13] ;
- la suite de premier terme 1 et de raison 1 est la suite des entiers naturels non nuls ;
- la suite de premier terme 1 et de raison 2 est la suite des nombres impairs ;
- la suite de premier terme 1 et de raison 3 est la suite des nombres congrus à 1 modulo 3 ; elle interviendra plus bas dans la suite (!) des aventures d'Euler au pays des sommes de diviseurs. Voici comment on peut la définir dans MathsOntologie :



Bref, une suite arithmétique est récurrente, et sa raison est le nombre qui figure après « augmenteDe » dans le script ci-dessus. Les suites arithmétiques se construisent toutes seules par des choses comme 1 jusque: 20 parPasDe: 3

---

11 Ainsi appelées parce qu'elles sont caractérisées par la propriété suivante : *Tout terme de la suite est la moyenne arithmétique entre le terme qui le précède et celui qui le suit*. Autrement dit, à chaque fois qu'on picore, on obtient la moyenne entre le picorement précédent et le prochain picorement...

12 Du latin « ratio », ou quotient, par analogie avec les suites géométriques ; il s'agit donc ici d'un abus de langage. Le mot « différence » serait plus adéquat. Et éviterait peut-être que des élèves, en lisant « quelle est sa raison ? » s'imaginent qu'il doivent expliquer *pourquoi* la suite est arithmétique...

13 Les entiers naturels forment donc une suite, d'habitude on ne le voit pas ainsi mais on peut la définir comme $u_n = n$; cette définition comme suite récurrente est à la base de l'axiomatisation de l'arithmétique par Guiseppe Peano.

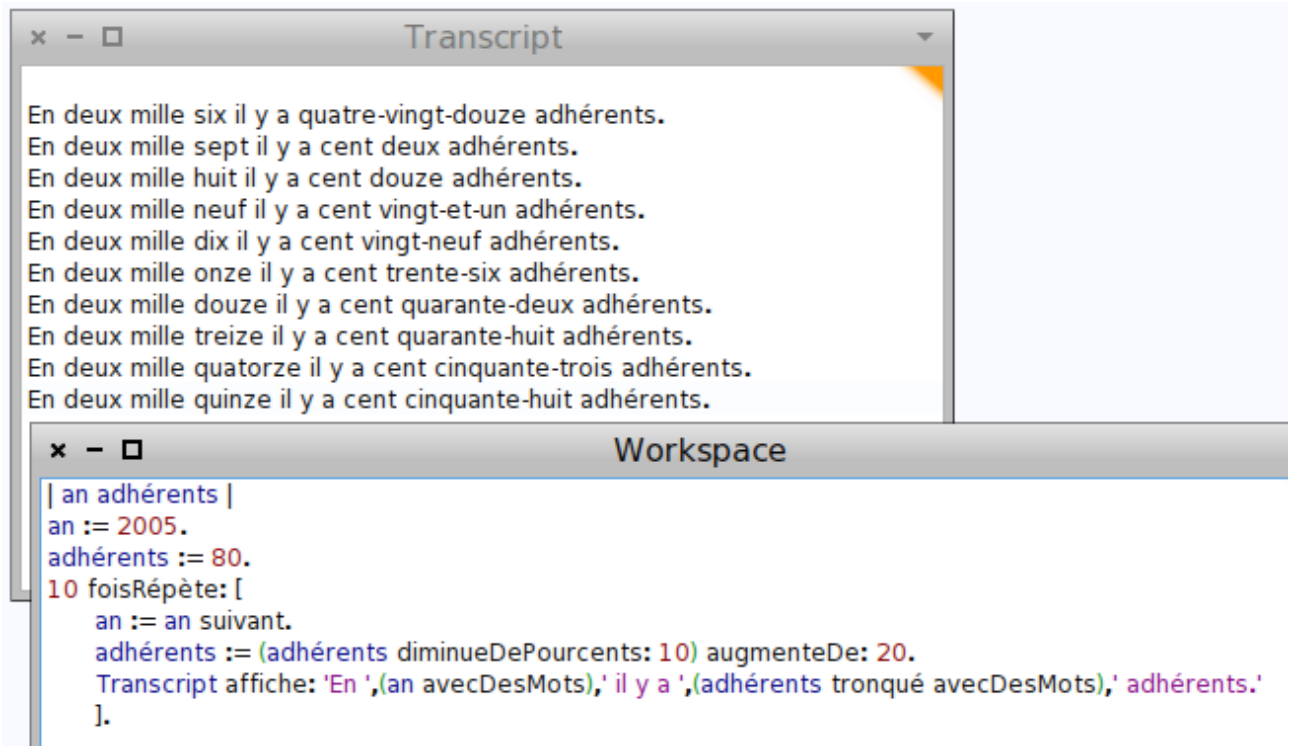
4) Suites géométriques

L'étape suivante avec les suites récurrentes, c'est le cas où, au lieu d'additionner toujours le même nombre comme ci-dessus, on multiplie toujours par le même nombre. On parle alors de suite géométrique[14], et cette fois-ci la raison de la suite est vraiment un quotient. Les suites géométriques apparaissent souvent lorsqu'il est question de pourcentages, et dans MathsOntologie, on remplace augmenteDe par augmenteDePourcents. Et ceci concerne également les suites dites arithmético-géométriques, où on fait un peu des deux, et qui sont à la mode au bac.

Voici un exemple extrait de sujet de bac (ES Antilles-Guyane septembre 2013) :

> En 2005, année de sa création, un club de randonnée pédestre comportait 80 adhérents. Chacune des années suivantes
> on a constaté que :
> — 10 % des participants ne renouvelaient pas leur adhésion au club ;
> — 20 nouvelles personnes s'inscrivaient au club.
> On suppose que cette évolution reste la même au fil des ans.

On peut donc étudier la suite des nombres d'adhérents, année après année, avec ce script :



Souvent, une suite définie de manière récurrente peut aussi être définie par une fonction. Mais alors que la fonction était facile à trouver pour les suites arithmétiques (elle est affine), la fonction est plus abstraite pour les suites géométriques[15] parce qu'elle fait appel à des puissances. On peut donc dire que la définition récurrente des suites géométriques est plus naturelle que la définition explicite, laquelle présente pourtant un intérêt dans les recherches de limites :

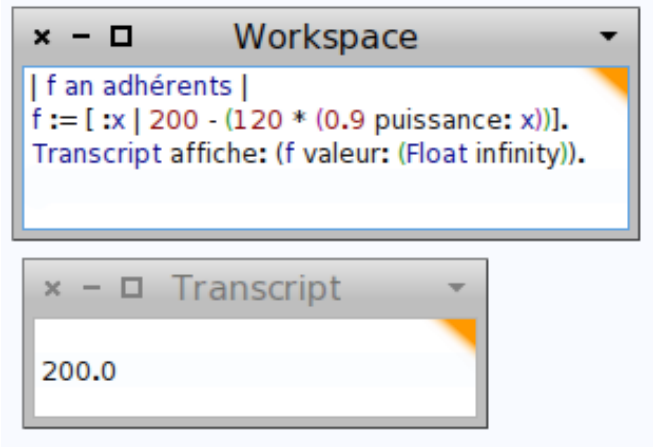
---

14 Parce que chaque terme est la moyenne géométrique entre le terme qui le précède et le terme suivant, du moins lorsque les termes sont tous positifs.

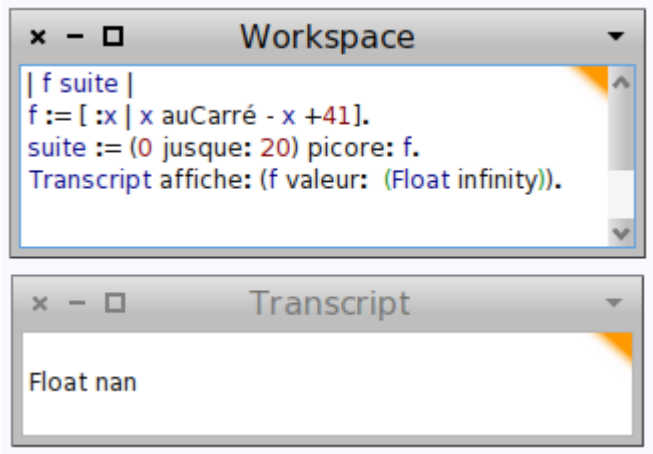
15 Pour les suites arithmético-géométriques, on peut montrer qu'elles sont sommes de suites constantes et de suites géométriques. Elles possèdent donc aussi une forme explicite.

5) L'infini

MathsOntologie permet, dans certains cas, de calculer la limite d'une fonction (et partant, d'une suite), en calculant l'image de l'infini par cette fonction. L'infini est un objet de type Float, appelé infinity. On le crée donc en écrivant Float infinity, et on demande simplement l'image de ce « nombre » par la fonction. Pour le sujet de bac précédent, à partir du moment où on a une expression explicite de la suite, on a rapidement sa limite, en admettant qu'elle est égale à celle de la fonction :

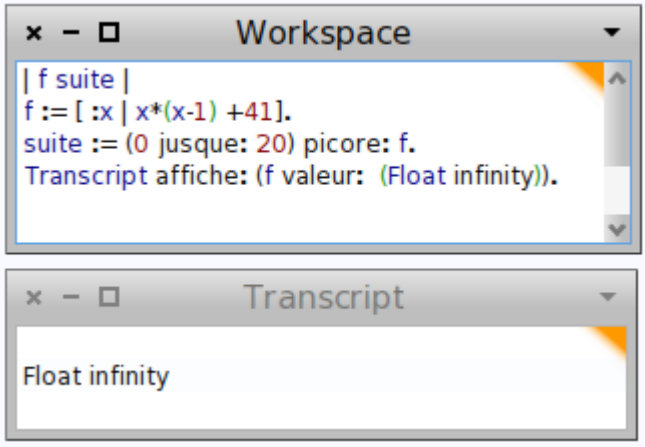


Cette technique (basée sur la gestion des dépassements de capacité par MathsOntologie) ne fonctionne pas toujours : D'une part, il faut une expression explicite de la suite, et d'autre part celle-ci ne doit pas comporter de forme indéterminée. Par exemple, la suite $u_n = n^2 - n + 41$ vue au début de cet article tend clairement vers l'infini, mais MathsOntologie ne le sait pas, à cause de la forme indéterminée $\infty - \infty$:



L'objet « nan » (« not a number » : Un nombre à la Magritte) est lui aussi de type Float. Et comme un Float est un nombre, on en déduit que « je ne suis pas un nombre » est un nombre... En tout cas, celui-ci signale ici que la gestion des dépassements de capacité ne permet pas de trouver la limite s'il y en a une.

Ceci dit, on peut souvent lever des indéterminations en factorisant au moins partiellement la suite :



On voit là que la limite de $u_n = n^2 - n + 41$ est infinie.

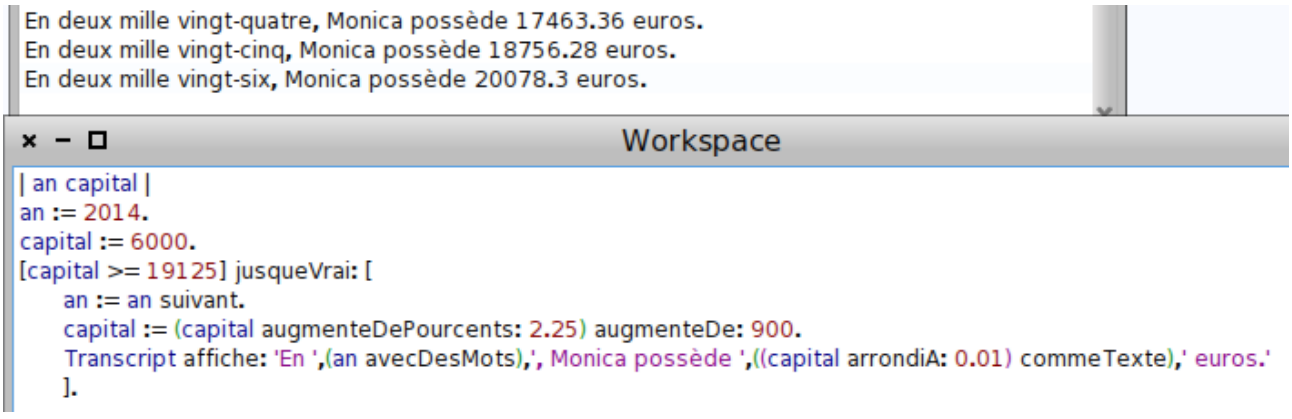
II/ Quelques applications

1) Recherche de seuils

Encore un sujet de bac, encore le bac ES, encore l'année 2013, avec cet extrait du bac ES Nouvelle-Calédonie de novembre 2013 :

> Le premier janvier 2014, Monica ouvre un livret d'épargne sur lequel elle dépose 6 000 euros.
> Elle décide de verser 900 euros sur ce livret chaque premier janvier à partir de 2015 jusqu'à atteindre le plafond autorisé de 19 125 euros.
> On suppose dans tout cet exercice que le taux de rémunération du livret reste fixé à 2,25 % par an et que les intérêts sont versés sur le livret le premier janvier de chaque année.

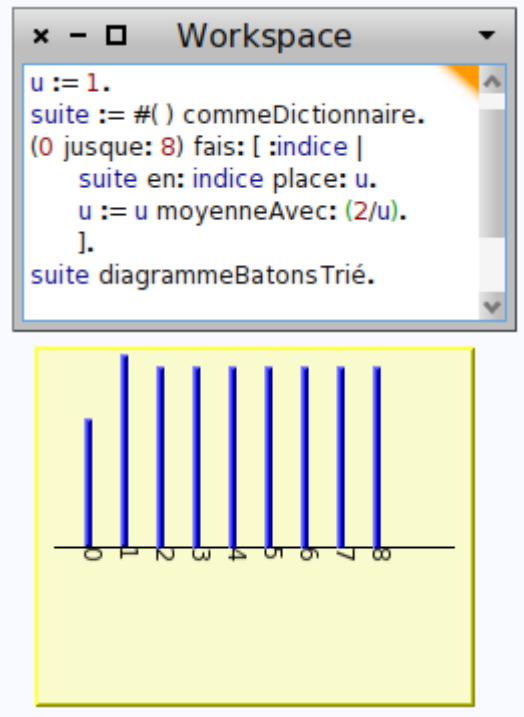
La description de l'algorithme avec MathsOntologie est assez proche de celle de l'énoncé :



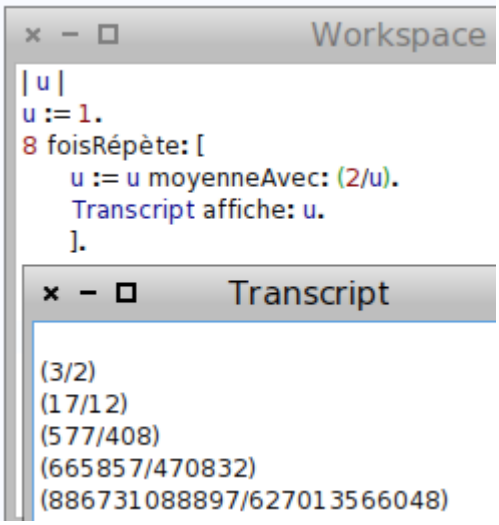
2) Algorithme de Heron

L'un des plus vieux algorithmes connus est celui de Heron pour calculer une racine carrée : En effet il était apparemment connu des babyloniens qui l'avaient sans doute utilisé pour calculer une valeur approchée de $\sqrt{2}$. Comme beaucoup d'algorithmes classiques, celui-ci consiste à construire une suite convergeant vers $\sqrt{2}$, et à calculer un terme de la suite d'indice suffisamment élevé pour que le terme soit proche de la limite : Ce terme est alors une valeur approchée de $\sqrt{2}$.

Or la suite est récurrente, donnée par $u_{n+1}=\frac{u_n+\frac{2}{u_n}}{2}$. On peut la mettre en œuvre ainsi :

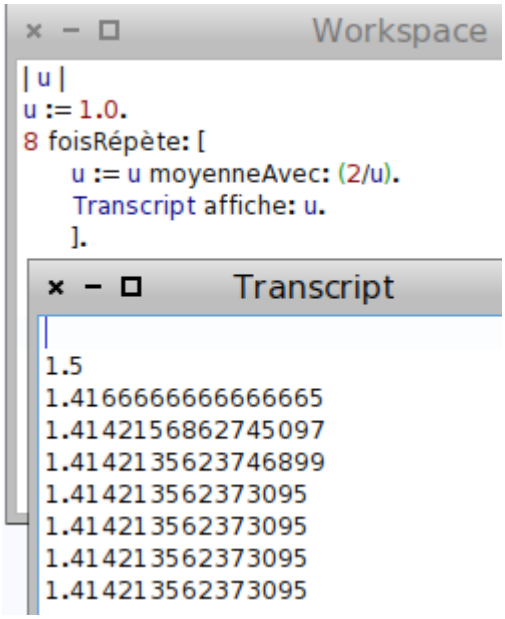


La représentation graphique montre la rapide convergence. Pour conjecturer la limite, il faut passer par l'affichage numérique des termes de la suite :



Au secours ! MathsOntologie fait ses calculs, par défaut, en valeurs exactes, et cet affichage a au moins le mérite de rappeler que la suite des approximations de $\sqrt{2}$ est rationnelle, pour peu que son premier terme le soit.

Pour mieux voir les termes de la suite, on peut juste prendre 1.0 (réel) à la place de 1 (entier) comme premier terme :

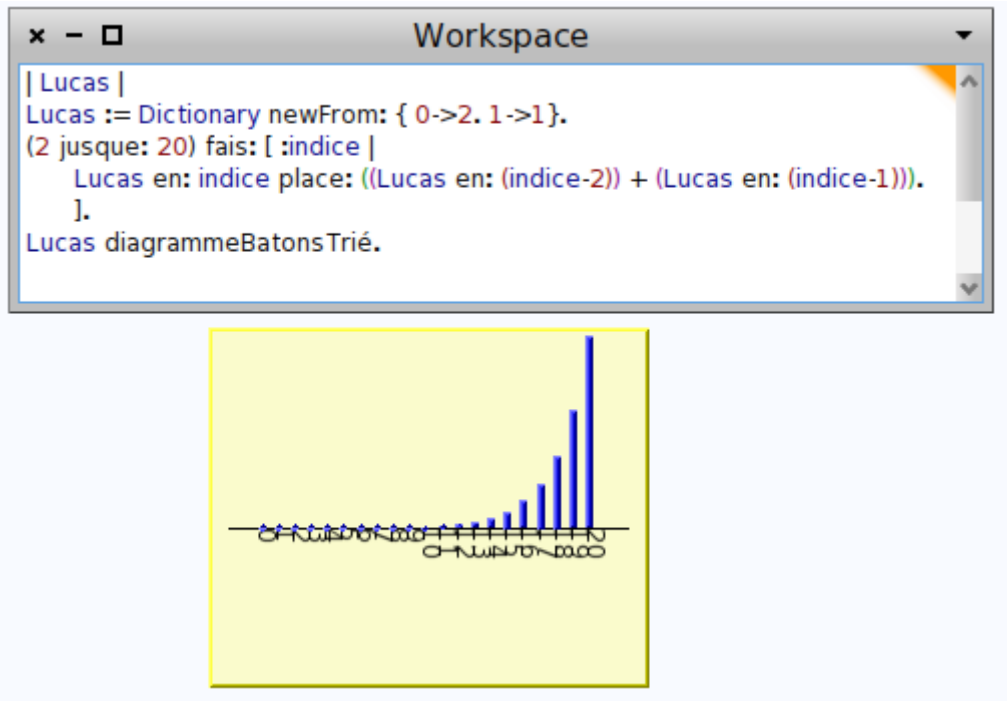


3) Suites de Lucas

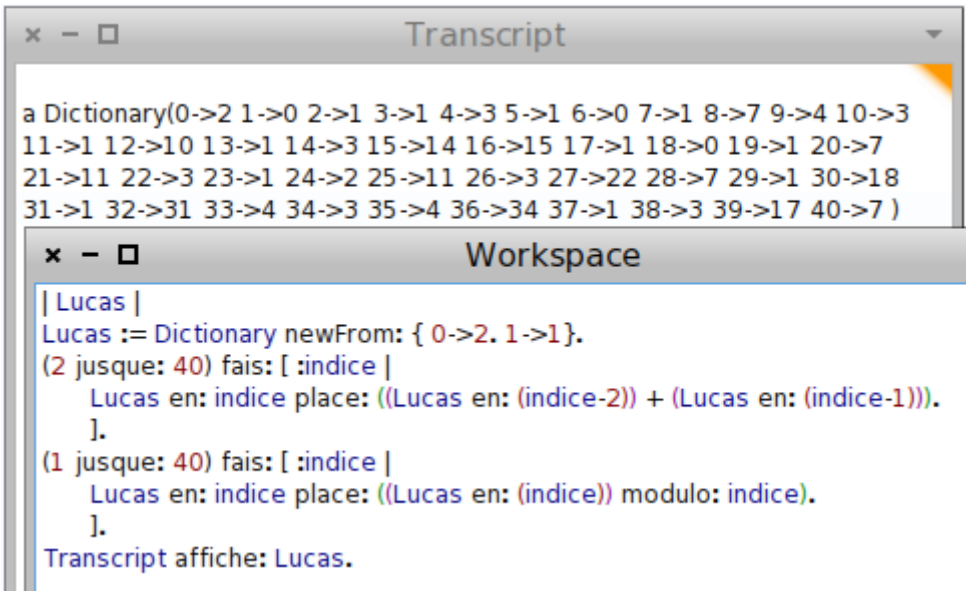
Édouard Lucas, passionné d'arithmétique, a inventé au moins deux suites récurrentes utiles pour tester la primalité des entiers :

a) Suite de Lucas

La suite de Lucas est très similaire à la suite de Fibonacci, à ceci près que son premier terme est 2 au lieu de 1. On la calcule donc par récurrence :



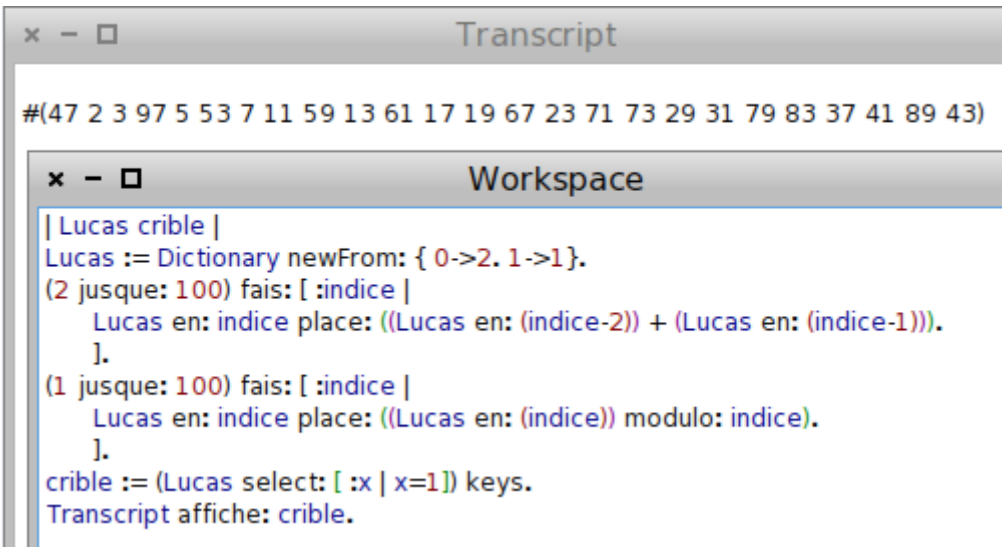
Comme sa grande sœur fibonaccienne, la suite de Lucas tend exponentiellement vers l'infini, comme on peut le conjecturer sur la représentation graphique. D'ailleurs les deux suites sont asymptotiquement équivalentes à l'infini, étant toutes les deux asymptotiquement géométriques de raison le nombre d'or. C'est en considérant, non pas le terme général de la suite de Lucas, mais le terme général modulo l'indice, qu'il se passe des choses intéressantes :



Cela ne saute peut-être pas aux yeux, la suite a maintenant l'air plutôt chaotique ; mais en regardant les valeurs de l'indice pour lesquelles la suite modulo l'indice vaut 1, on reconnaît des nombres familiers. Alors on peut commencer par fabriquer un crible en ne gardant que ces valeurs de l'indice pour lesquelles le terme général de la suite de Lucas est congru à 1, en construisant une image réciproque[16] par la fonction ainsi définie[17] :

16 L'image réciproque de 1 est l'ensemble des antécédents de 1 ; chercher une image réciproque, c'est donc résoudre une équation.

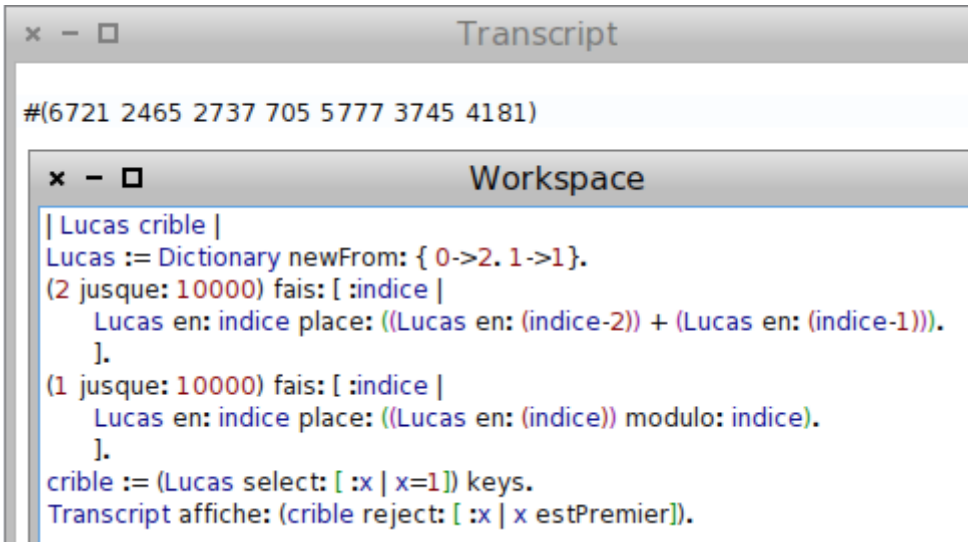
17 La suite de Lucas modulo l'indice est une fonction de $\mathbb{N}$ dans $\mathbb{N}$



Une fois de plus, des nombres premiers... En fait, en rejetant (« reject ») les nombres premiers de cette liste de nombres, on ne récupère qu'un tableau vide.

> **Définition** : On appelle ***nombre pseudo-premier de Lucas***, un nombre *n* tel que le n-ième terme de la suite de Lucas est congru à 1 modulo *n*, mais qui n'est pas premier.

Le script ci-dessus montre donc qu'il n'y a pas de nombre pseudo-premier de Lucas qui soit inférieur à 100. Donc s'il y en a, ils sont rares et grands[18]. Vérification, en listant les nombres pseudo-premiers de Lucas inférieurs à 10000 :

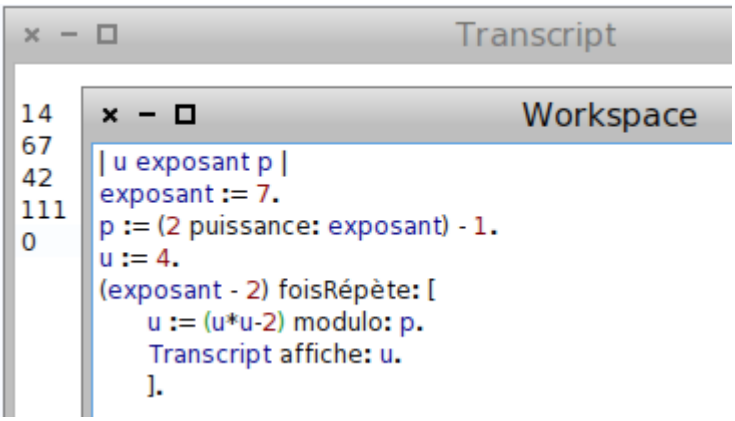


Il a fallu quelques secondes pour trouver les 7 nombres pseudo-premiers de Lucas inférieurs à 10000. Le plus petit est 705 qui est divisible par 5.

18 Et s'il n'y en a pas, cela veut dire qu'on peut très facilement tester la primalité d'un nombre, puisque le calcul des termes de la suite de Lucas peut se faire modulo n. Mais ça se saurait...

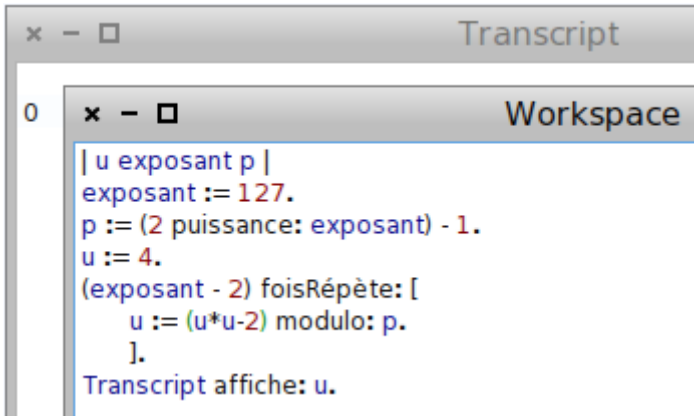
b) Suite de Lucas-Lehmer

Pour tester la primalité d'un nombre de Mersenne[19], on utilise un test de Lucas amélioré par Lehmer, et qui lui aussi est basé sur une suite récurrente[20]. On calcule donc les termes successifs de la suite récurrente $u_{n+1}=u_n^2-2$ (modulo le nombre *p* à tester) jusqu'à ce qu'on en ait calculé autant que l'exposant du nombre *p*. Si l'antépénultième[21] est nul, p est premier, et réciproquement. On vérifie avec $127=2^7-1$ (on sait que 7 est premier). On doit donc calculer 7-2=5 termes de la suite, modulo 127 :



Comme le dernier terme calculé est 0, on a la confirmation de la primalité de 127, et ceci de façon presque instantanée.

Pour vérifier avec un nombre plus grand, on choisit l'exposant 127 dont on vient de montrer qu'il est premier, et on doit donc boucler 125 fois ; on n'affiche que le dernier terme pour regarder s'il est nul :



Bingo ! Il est nul, donc $2^{127}-1=170141183460469231731687303715884105727$ est premier. Impressionnant...

---

19 Un nombre de la forme 2^e-1 où e est premier

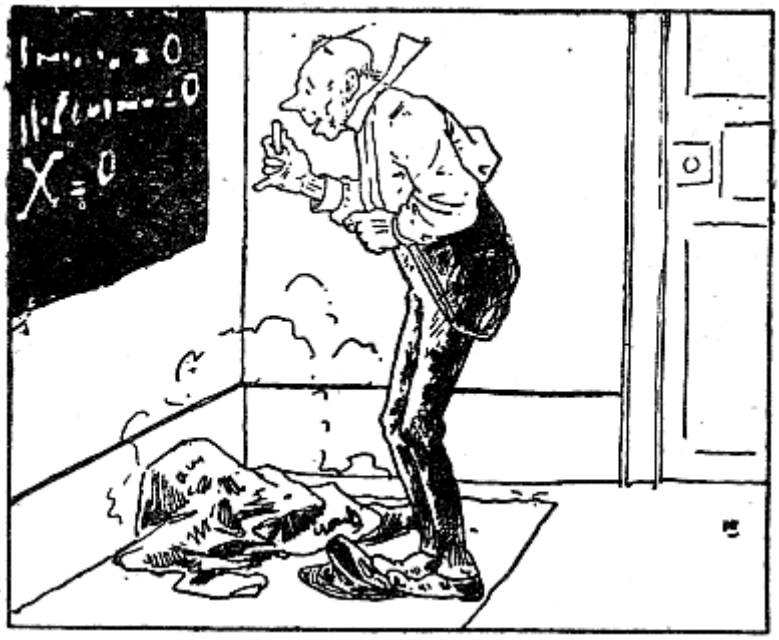
20 En fait c'est une suite « à la Julia », obtenue en itérant z^2-2 à partir de 4. L'ensemble de Julia correspondant est un segment.

21 On n'a donc pas besoin de calculer les deux derniers termes

On remarque que, une fois de plus, on a fait plein de calculs pour aboutir à un nombre aussi simple que 0 ; ce fait, lui-même récurrent, avait été signalé par le bédéaste Christophe[22] dès 1893 :

> *À trois heures et demi, le docteur découvre la valeur de x, l'inconnue cherchée ; ce qui lui cause une joie sans mélange.*
> *– Nous prions les esprits superficiels de s'abstenir de toute réflexion sur la valeur de x, et de ne point prétendre que Zéphyrin a beaucoup travaillé pour peu de chose.*

Avec l'illustration :



III/ Séries

1) Somme de termes

Des calculs de géométrie comme la longueur d'une courbe ou l'aire délimitée par une courbe (calcul intégral) sont basés, non sur une suite explicite, mais sur la somme de ses termes. Celle-ci, menée de 0 jusqu'à n, est elle-même une suite, dont la forme explicite et la limite présentent souvent un intérêt mathématique. On constate qu'une série est une suite récurrente avec $S_{n+1}=S_n+u_{n+1}$ (en posant $S_n=\sum_{k=0}^{n} u_k$). Le simple développement décimal d'un réel est une série...

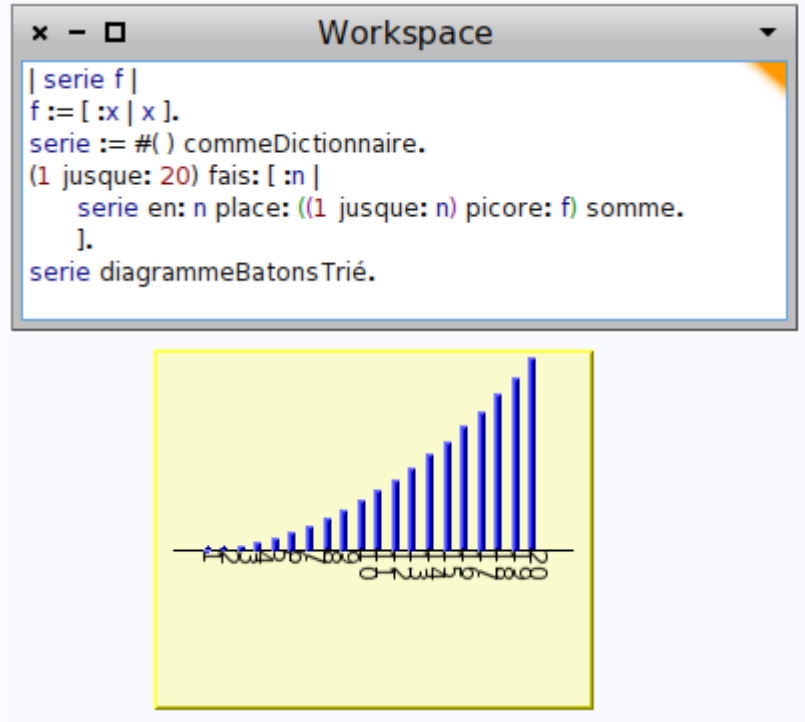
Mais les algorithmes de statistique présents dans MathsOntologie permettent de déléguer à celui-ci les additions, plutôt que d'avoir à rédiger l'algorithme $S_{n+1}=S_n+u_{n+1}$. Il suffit pour cela de s'adresser au tableau construit comme début de la suite, et de lui demander la somme de ses éléments, avec le message somme.

22 Christophe, « l'idée fixe du savant Cosinus », lisible en ligne ici : http://aulas.pierre.free.fr/chr_cos_01.html

2) séries arithmétiques

a) Nombres triangulaires

La somme des premiers entiers est un nombre triangulaire[23]. Pour la calculer, on additionne les entiers :



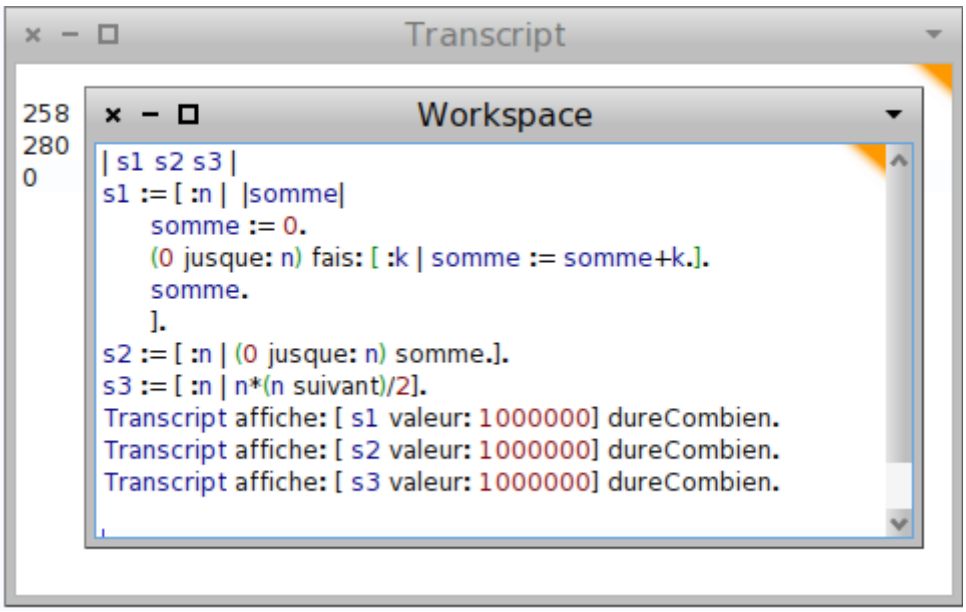
En fait, on démontre par récurrence qu'il y a une formule explicite pour les nombres triangulaires : $\sum_{k=0}^{n} k = \frac{n(n+1)}{2}$, ce qui explique l'aspect de parabole qu'on voit sur la représentation graphique.

En dehors du fait que cette formule explicite permet de calculer la limite de la série, le calcul est plus rapide qu'en effectuant beaucoup d'additions. On peut le vérifier par une mesure du temps pris à exécuter certains algorithmes :

- Le calcul de la somme à l'aide de la formule récurrente $\sum_{k=0}^{n+1} k = n+1+\sum_{k=0}^{n} k$ (s1 ci-dessous) ;
- La création de la liste des entiers puis le calcul de la somme de ses termes (s2 ci-dessous) ;
- La formule explicite (s3 ci-dessous).

La seconde méthode est légèrement plus lente que la première, probablement parce qu'il faut de la place dans la mémoire pour la longue liste de nombres, mais comme attendu, la troisième méthode est largement plus rapide que les deux autres (le temps sont donnés en millisecondes) :

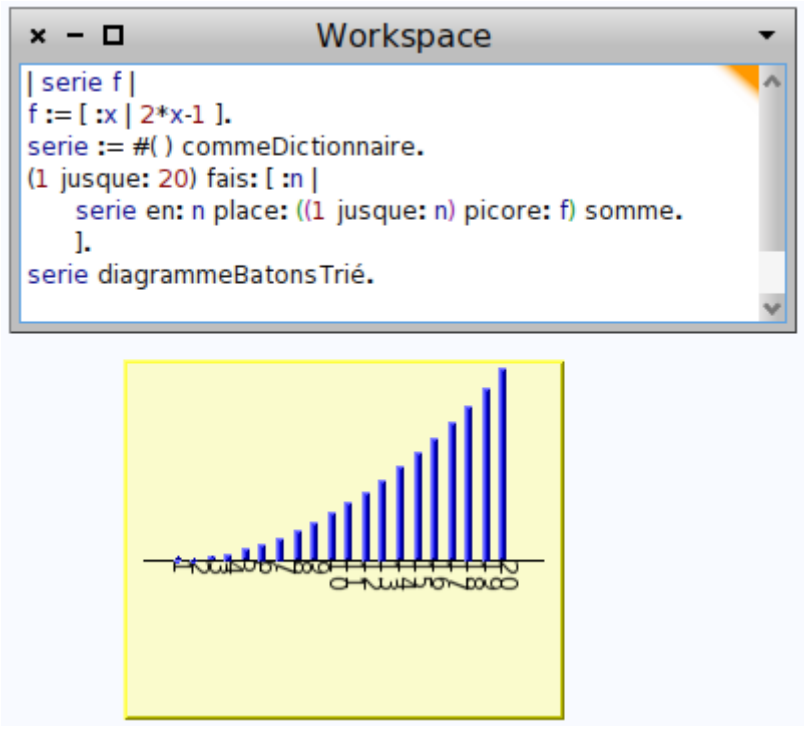
23 http://fr.wikipedia.org/wiki/Nombre_triangulaire



Cette formule se généralise à la somme des termes de toute suite arithmétique puisque $\sum_{k=0}^{n} ak+b = a\times\sum_{k=0}^{n} k+\sum_{k=0}^{n} b = a\frac{n(n+1)}{2}+b(n+1)$. Mais dans le cas de la suite des impairs on a un résultat intéressant :

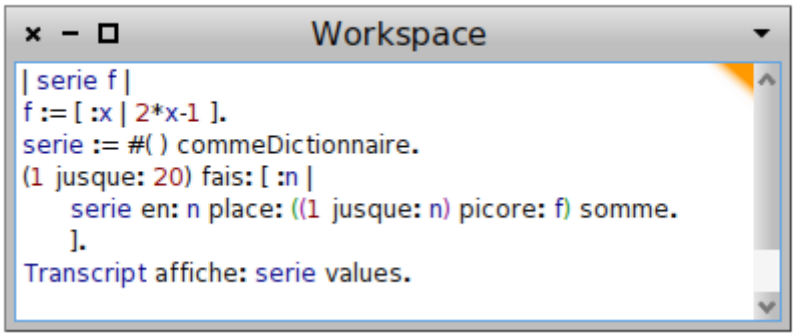
b) Nombres carrés

La somme des termes de la suite de premier terme 1 et de raison 2 (les nombres impairs) est également du second degré :



Mais en regardant les valeurs numériques plutôt que la représentation graphique, on a une surprise[24] :

```
#(1 4 9 16 25 36 49 64 81 100 121 144 169 196 225 256 289 324 361 400)
```

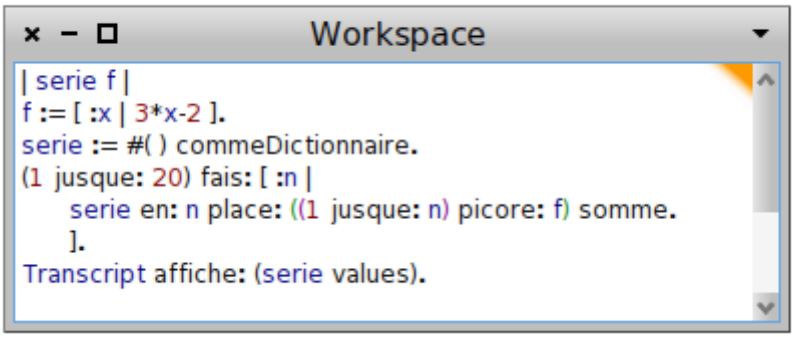


Autrement dit, $\sum_{k=0}^{n}(2k+1)=n^2$.

c) Nombres pentagonaux

Par conséquent, il est naturel d'appeler ***nombres pentagonaux***[25], les termes de la somme des nombres congrus à 1 modulo 3. Les voici donc calculés :

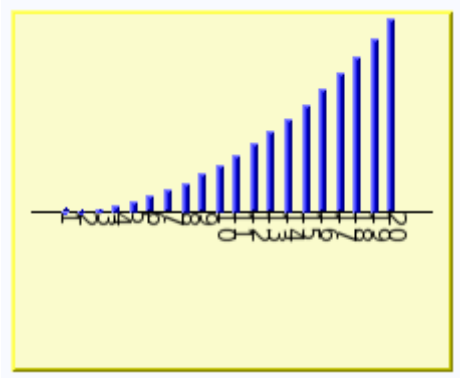
```
#(1 5 12 22 35 51 70 92 117 145 176 210 247 287 330 376 425 477 532 590)
```



24 On n'affiche pas le dictionnaire entier, seulement son ensemble d'arrivée appelé values (ce sont les valeurs que prend la fonction réalisée par la dictionnaire)

25 Et hexagonaux, les termes de la somme des nombres congrus à 1 modulo 4, etc. La généralisation des nombres triangulaires, carrés, pentagonaux etc. est appelée « ***nombres figurés*** » par Jean Le Rond d'Alembert dans *l'Encyclopédie ou Dictionnaire Raisonné des arts, des sciences et des métiers*, de 1751 : On remarque que *l'Encyclopédie* a été publiée la même année que l'article d'Euler qui sert de fil conducteur à celui-ci.

et représentés :



Euler s'intéresse aussi à la somme des nombres congrus à 2 modulo 3 :

```
#(2 7 15 26 40 57 77 100 126 155 187 222 260 301 345 392 442 495 551 610)
```

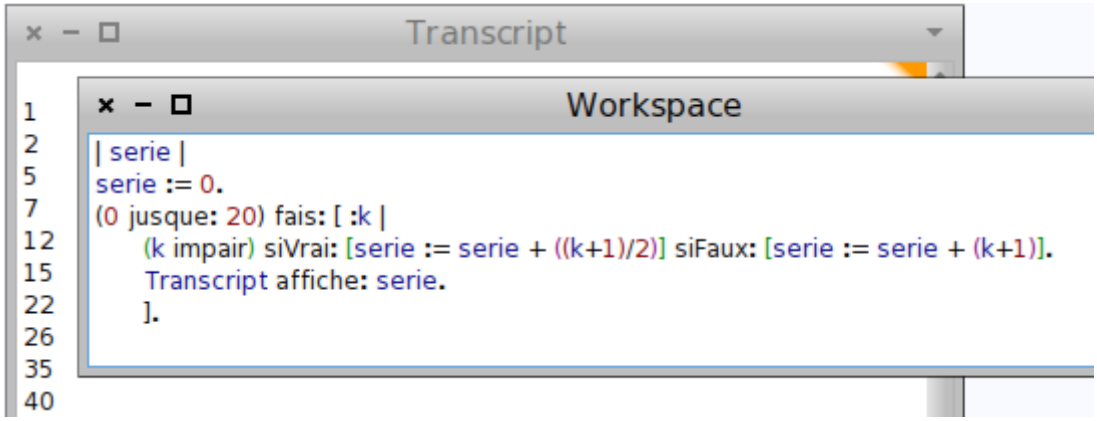
Workspace

```
| serie f |
f := [ :x | 3*x-1 ].
serie := #( ) commeDictionnaire.
(1 jusque: 20) fais: [ :n |
    serie en: n place: ((1 jusque: n) picore: f) somme.
    ].
Transcript affiche: (serie values).
```

En fait, il mélange les deux suites pour avoir les ***nombres pentagonaux généralisés***. Ceux-ci interviennent dans le calcul des sommes de diviseurs, et Euler les décrit ainsi :

> *alternativement on aura tous les nombres naturels 1, 2, 3, 4, 5, 6, etc. et les nombres impairs 3, 5, 7, 9, 11, etc. Par ce moyen on pourra continuer la suite de ces nombres aussi loin que l'on voudra.*

Avec MathsOntologie cela donne



Si l'indice k est impair, on ajoute à la somme de la série, le terme (k+1)/2 qui parcourt successivement les entiers à partir de 1. Sinon, c'est que k est pair, et dans ce cas on ajoute à la somme de la série, k+1 qui parcourt les nombres impairs. La série parcourt bien les nombres pentagonaux généralisés d'Euler. On va voir plus bas ce qu'il fait avec ces nombres.

3) séries géométriques

Pour connaître la somme des termes d'une suite géométrique, on a besoin de la somme des puissances d'un nombre. Or celle-ci intervient comme valeur de $\sigma(a^n)$ où a est premier. Voici ce qu'en dit Euler :

> *il est clair que, si p marque un nombre premier, la valeur de $\sigma(p)$ sera $=1+p$; excepté le cas où $p=1$, car alors nous avons $\sigma(1)=1$, et non pas $\sigma(1)=1+1$; d'où l'on voit qu'il faut exclure l'unité de la suite des nombres premiers; étant le commencement des nombres entiers, elle n'est ni premier ni composé.*

Les nombres premiers sont en quelque sorte « super-déficients » : Ils dépassent largement la somme de leurs diviseurs. Mais pour les puissances de nombres premiers, Euler écrit :

> *Pour les puissances des nombres premiers, on a besoin de règles particulières, comme:*
>
> $$\sigma(a^2)=1+a+a^2=\frac{a^3-1}{a-1}$$
> $$\sigma(a^3)=1+a+a^2+a^3=\frac{a^4-1}{a-1}$$
> *et généralement*
> $$\sigma(a^n)=\frac{a^{n+1}-1}{a-1}$$

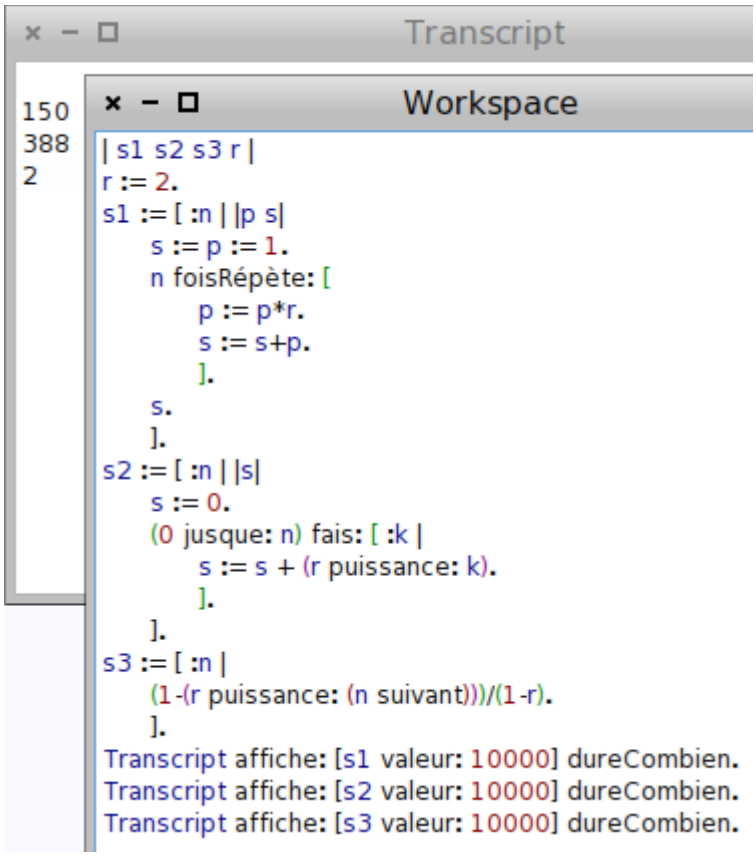
Comme la somme des diviseurs d'un produit est le produit des sommes de diviseurs, Euler peut alors utiliser la décomposition en facteurs premiers d'un entier pour calculer rapidement la somme de ses diviseurs. Par exemple,

$$\sigma(360)=\sigma(8\times9\times5)=\sigma(8)\times\sigma(9)\times\sigma(5)=\frac{2^4-1}{2-1}\times\frac{3^3-1}{3-1}\times(5+1)=15\times13\times6=1170\ .$$

Comme pour les séries arithmétiques, on imagine qu'il est plus rapide de calculer $\frac{2^{1001}-1}{2-1}$ que d'additionner 1001 puissances de 2. On peut le vérifier en mesurant le temps pris par trois algorithmes différents mais qui font la même chose :

- La somme des puissances de 2 calculées elles-mêmes de façon récurrente (s1 ci-dessous) ;
- La somme des puissances de 2 calculées avec « puissance » (s2 ci-dessous) ;
- La formule (s3 ci-dessous)

La fonction « puissance » est nettement plus lente que la double boucle, mais la formule explicite est nettement plus rapide :



La formule permet également de calculer la limite de la série, comme on a vu plus haut avec les suites explicites.

4) Sommes de diviseurs

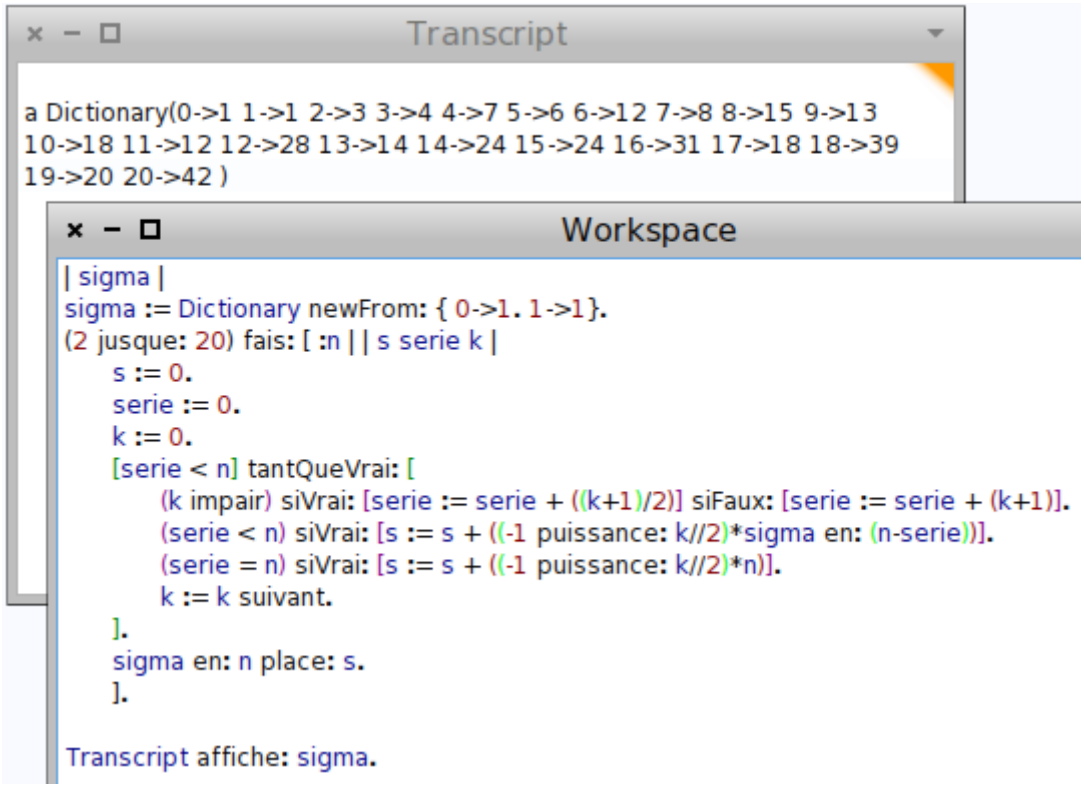
La « *découverte d'une loi tout extraordinaire des nombres, par rapport à la somme de leurs diviseurs* » d'Euler est la relation de récurrence « à la Fibonacci » suivante :

$$\sigma_n = \sigma_{n-1} + \sigma_{n-2} - \sigma_{n-5} - \sigma_{n-7} + \sigma_{n-12} + \sigma_{n-15} - \sigma_{n-22} - etc.$$ où

- les termes sont additionnés deux à la suite, puis soustraits deux à la suite, etc en alternance ;
- les indices sont calculés en soustrayant les nombres pentagonaux généralisés à l'indice courant, tant que la soustraction ne mène pas à un résultat négatif ;
- la somme des diviseurs de 0 est indéterminée, alors il faut la remplacer par n :

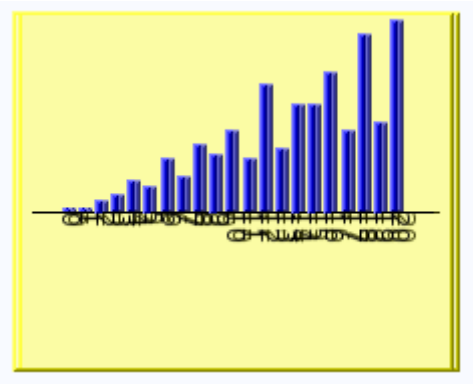
> *S'il arrive que le terme* σ_0 *se rencontre dans cette formule, comme sa valeur est indéterminée en elle-même, il faut, dans chaque cas, au lieu de* σ_0 *, mettre le nombre proposé même.*

L'algorithme, en MathsOntologie, donne ceci :

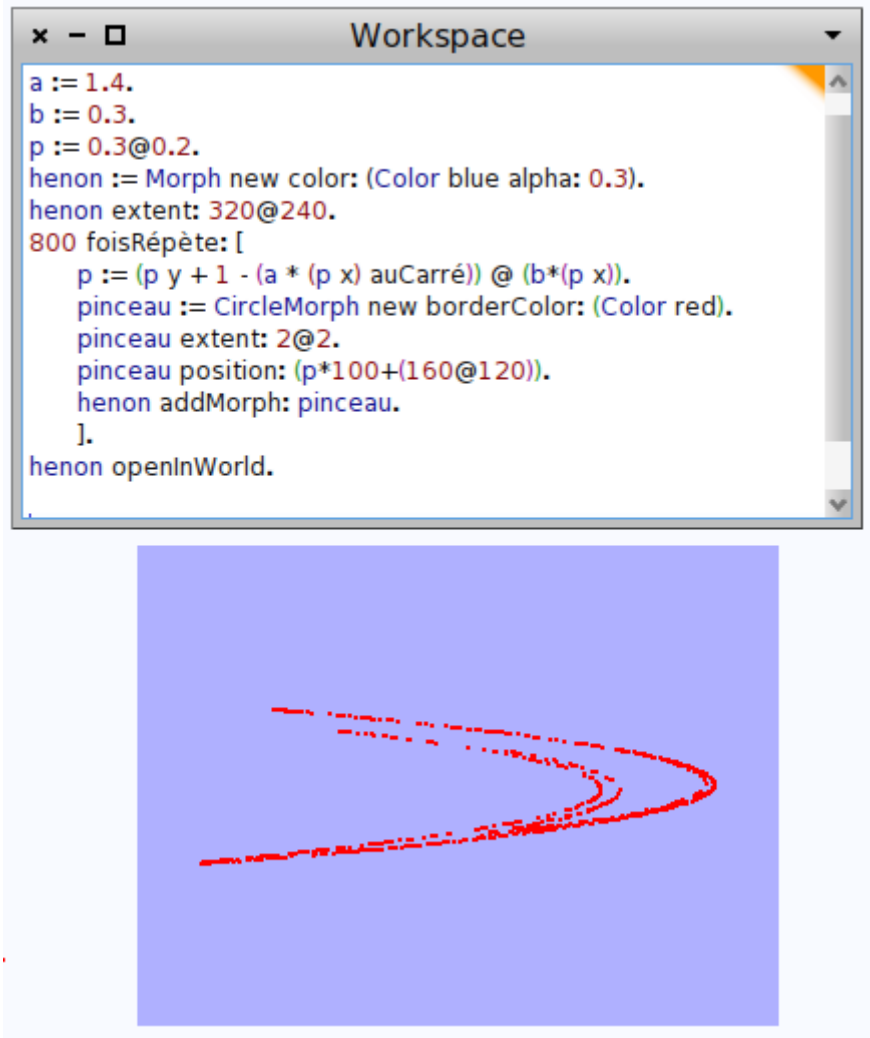


L'expression -1 puissance: k//2 prend successivement les valeurs 1, 1, -1, -1, 1, 1, -1, -1, etc. Le « double slash » représente le quotient euclidien, qui est pair si l'indice modulo 4 vaut 0 ou 1, et impair dans les autres cas. En élevant -1 à cette puissance, on a donc bien la suite souhaitée.

On peut comparer avec l'autre manière d'additionner les diviseurs, en superposant les deux représentations graphiques (elles sont translucides) :



Pour finir ce document, un exemple pour montrer que MathsOntologie sait aussi « calculer » des suites de points : L'attracteur de Hénon :



Alain Busser
LPO Roland-Garros
Le Tampon