BTS ABM 1
11/09/2017

Corrigé du TP n° 1

Sujet : *Le but du TP est de calculer le nombre e par un algorithme : En additionnant les termes d'une suite, la somme converge vers e.*

1. Première partie : Une suite auxiliaire

La variable p (comme « produit ») va successivement valoir le produit des entiers successifs jusqu'à n, avec l'algorithme suivant :

```
p ← 1
pour n∈[1..20]
    p ← p*n
    affiche p
```

On le programme avec alcoffeethmique, de la façon suivante :

```
p = 1
pour n dans [1..20]
    p = p*n
    affiche p
```

Le programme ci-dessus, lorsqu'on l'exécute, donne l'affichage suivant :

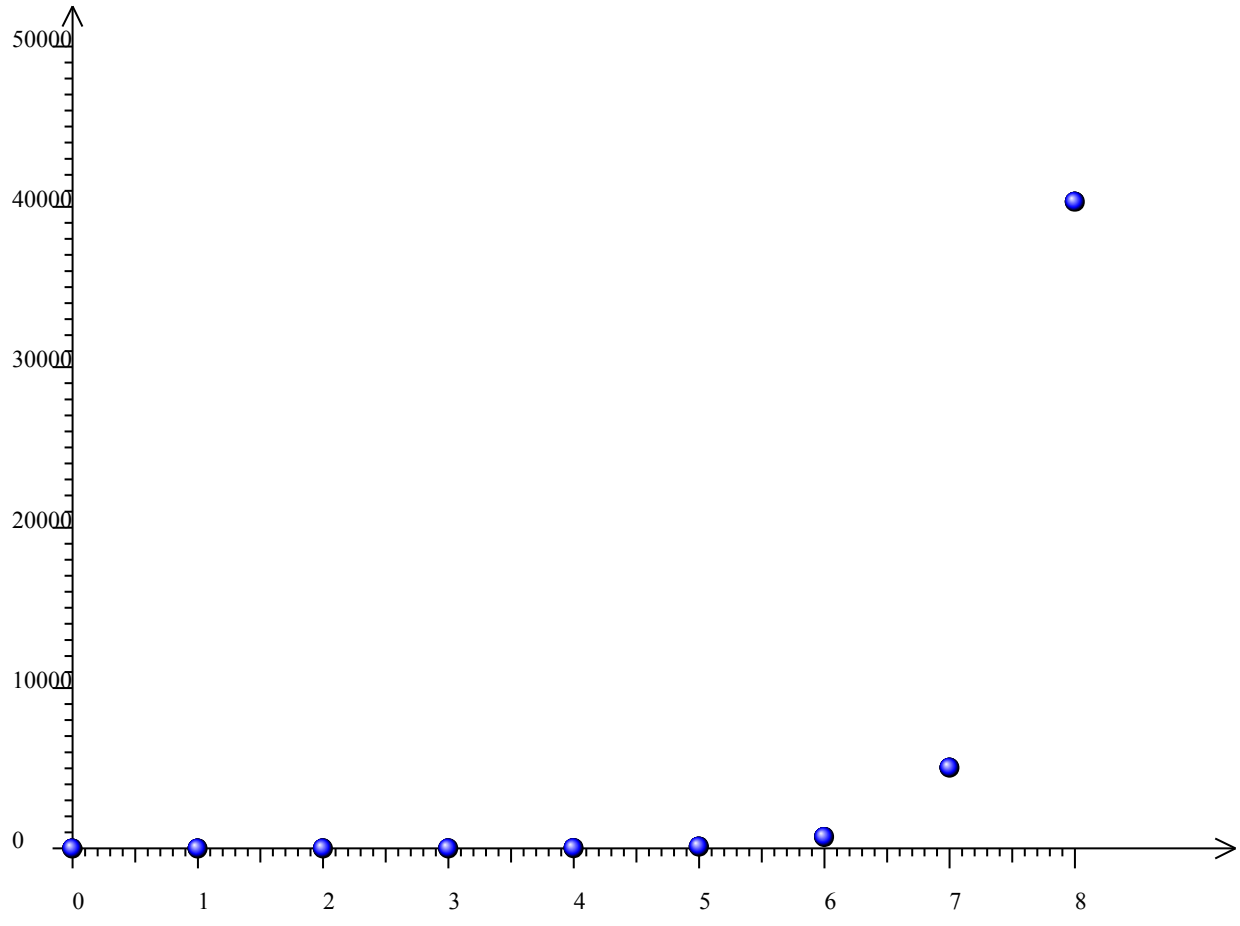
```
1
2
6
24
120
720
5040
40320
362880
3628800
39916800
479001600
6227020800
87178291200
1307674368000
20922789888000
355687428096000
6402373705728000
121645100408832000
2432902008176640000
```

Cet affichage suggère que la suite p_n des valeurs successives de p tend vers l'infini. Cela se confirme sur la représentation graphique de la suite p_n, obtenue par ce script :

```
p = 1
u = [1]
pour n dans [1..10]
    p = p*n
    u.empile p

dessineSuite u, 8, 0, 50000
```

Voici la représentation graphique obtenue :



2. Seconde partie : La somme des termes

Puisque la suite p_n tend rapidement vers l'infini, il en résulte que la suite des inverses de p tend, elle, rapidement, vers 0. On regarde alors quelle est la limite de la somme S_n des inverses, définie par la formule récurrente $S_{n+1} = S_n + p_{n+1}$:

```
p ← 1
S ← 1
pour n∈[1..20]
    p ← p*n
    S ← S + 1/p
    affiche S
```

Le script que voici

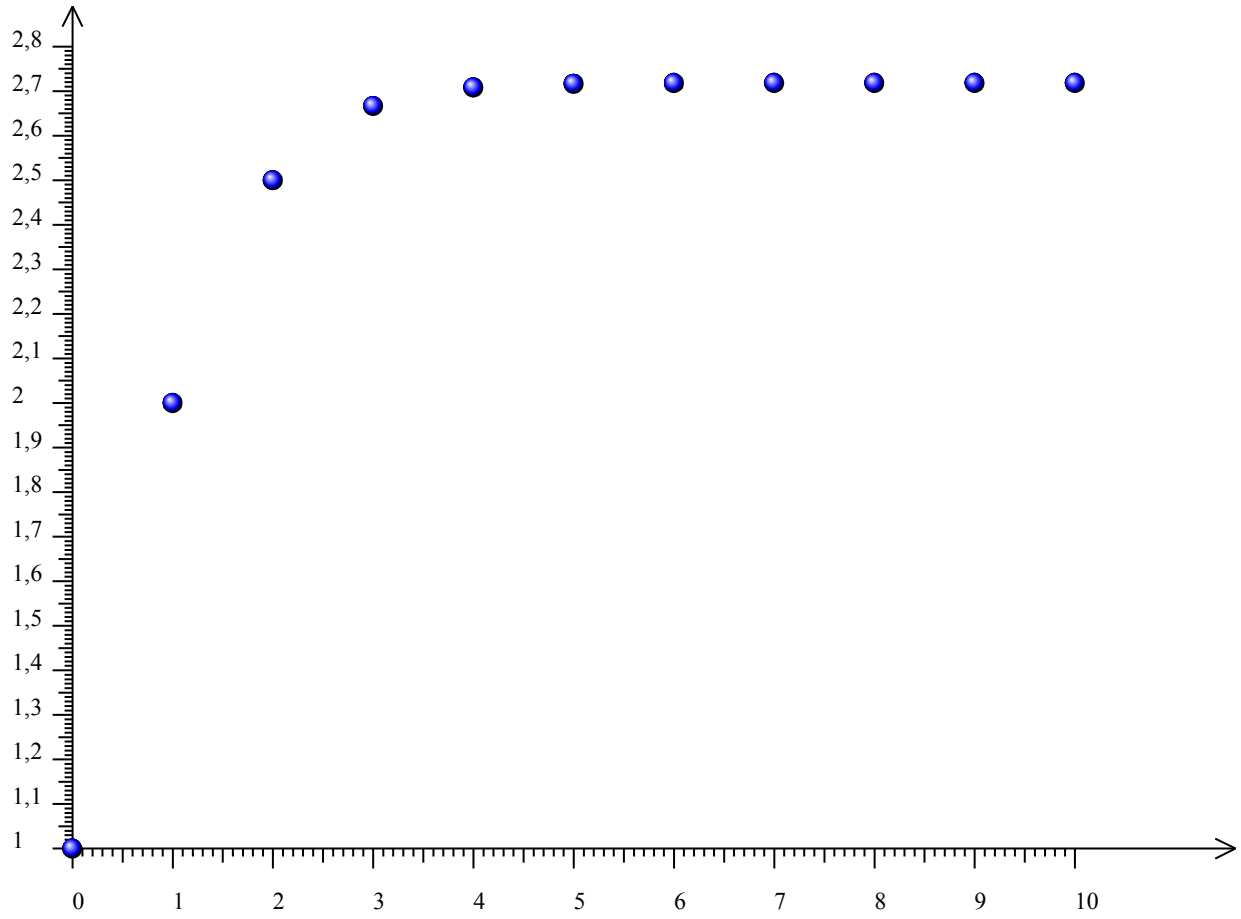
```
p = 1
S = 1
pour n dans [1..20]
    p = p*n
    S = S + 1/p
    affiche S
```

affiche les nombres suivants :

```
2
2.5
2.6666666666666665
2.708333333333333
2.7166666666666663
2.7180555555555554
2.7182539682539684
2.71827876984127
2.7182815255731922
2.7182818011463845
2.718281826198493
2.7182818282861687
2.7182818284467594
2.71828182845823
2.718281828458995
2.718281828459043
2.7182818284590455
2.7182818284590455
2.7182818284590455
2.7182818284590455
```

Une vérification avec la touche e de la calculatrice permet de voir que c'est bien le nombre e qui est calculé ainsi.

La rapidité de la convergence saute aux yeux sur la représentation graphique :



3. En grande précision

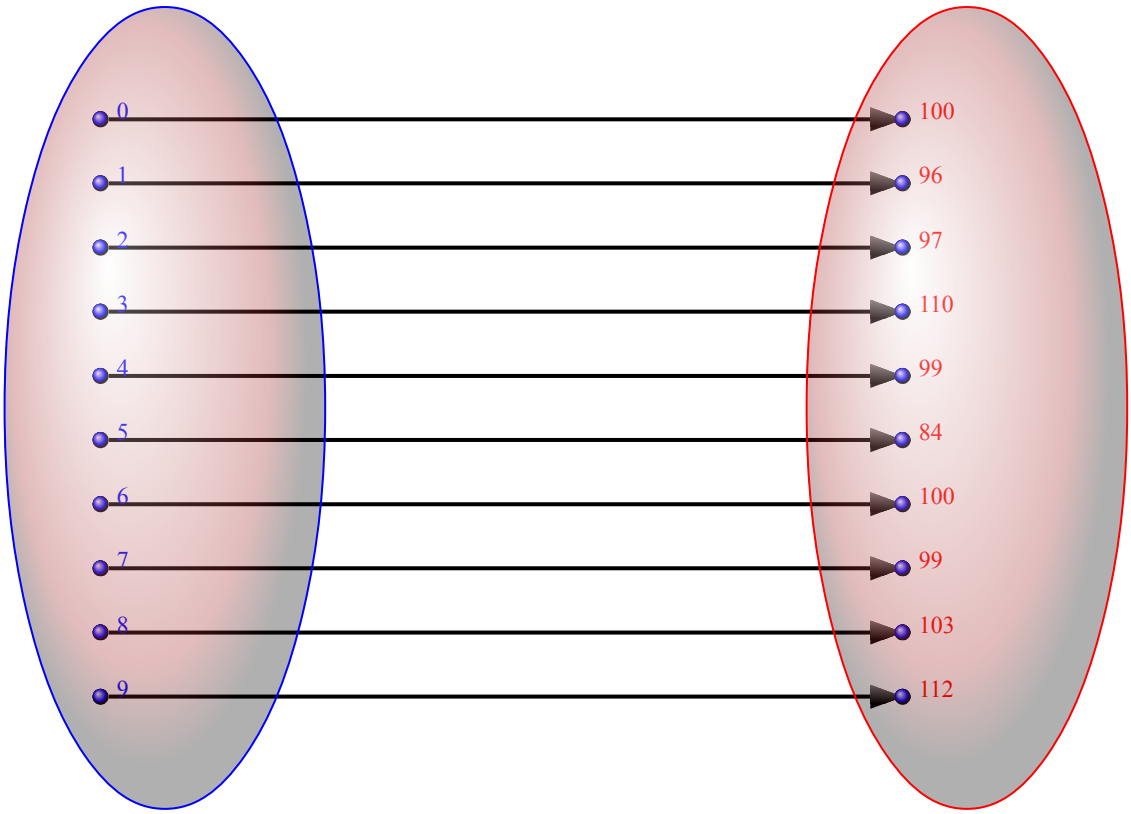
Avec alcoffeethmique on peut décider du nombre de chiffres sur lesquels calcule Big (utilitaire de calcul en grande précision). Par exemple, ce script permet de calculer e avec 1000 décimales :

```
Big.DP = 1000
p = Big "1"
S = Big "1"
pour n dans [1..500]
    p = p.times n
    S = S.plus p.pow(-1)
affiche S
```

Le résultat obtenu suggère bien que e n'est pas un nombre décimal :

```
2.718281828459045235360287471352662497757247093699959574966967627724076630353547594571382178525166427427466391932003059921817413596629043572900334295260595630738132328627943490763233829880753195251019011573834187930702154089149934884167509244761460668082264800168477411853742345442437107539077744992069551702761838606261331384583000752044933826560297606737113200709328709127443747047230696977209310141692836819025515108657463772111252389784425056953696770785449969967946864454905987931636889230098793127736178215424999229576351482208269895193668033182528869398496465105820939239829488793320362509443117301238197068416140397019837679320683282376464804295311802328782509819455815301756717361332069811250996181881593041690351598888519345807273866738589422879228499892086805825749279610484198444363463244968487560233624827041978623209002160990235304369941849146314093431738143640546253152096183690888707016768396424378140592714563549061303107208510383750510115747704171898610687396965521267154688957035036 3
```

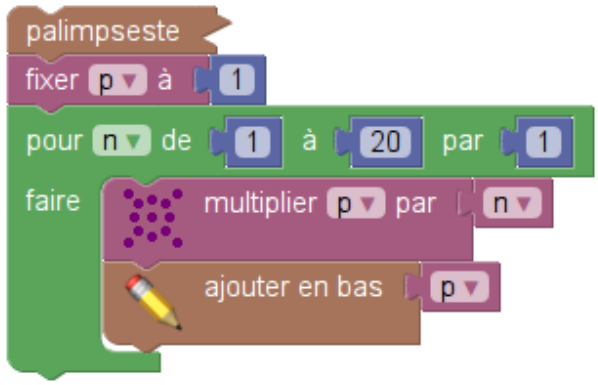
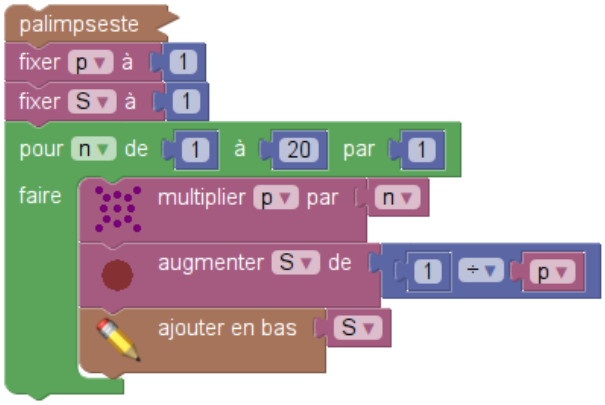
Avec autant de décimales on peut faire des statistiques sur la répartition des chiffres dans l'écriture décimale de e :



On constate que chaque chiffre apparaît environ 100 fois : Les chiffres de e semblent équirépartis (mais au hasard).

4. Conclusion

Cet algorithme, publié en 1748 par Leonhard Euler, se programmait avantageusement avec les variations en place de Sofus :

Calcul de p (produits)	Calcul de e (sommes)
	

Mais on n'avait même pas besoin de programmer les suites p_n et S_n : Il était tout-à-fait possible de les calculer avec un tableur.