

# Nieme\_Nombre\_Premier

January 9, 2021

## 1 Le nième nombre premier

Marc Lorenzi - 5 juin 2018

```
[1]: import matplotlib.pyplot as plt
     from math import log, sqrt
```

```
[2]: plt.rcParams['figure.figsize'] = (12, 6)
```

Pour  $n \geq 1$ , notons  $p_n$  le  $n$ -ième nombre premier. On a  $p_1 = 2$ ,  $p_2 = 3$ ,  $p_3 = 5$ . Mais que vaut  $p_n$  pour  $n$  quelconque ? Peut-on avoir un équivalent de  $p_n$  ? Un développement asymptotique de  $p_n$  ?

### 1.1 1. Nombres premiers

Les fonctions ci-dessous ne sont pas efficaces mais elles suffiront pour faire quelques simulations.

La fonction `smallest_divisor` prend un entier  $n \geq 2$  en paramètre. Elle renvoie le plus petit diviseur premier de  $n$ . Sa complexité est  $O(\sqrt{n})$ , ce qui ne permet de l'utiliser que pour des entiers d'au plus une dizaine de chiffres.

```
[3]: def smallest_divisor(n):
     k = 2
     while n % k != 0 and k * k <= n: k += 1
     if k * k > n: return n
     else: return k
```

```
[4]: smallest_divisor(1234567)
```

```
[4]: 127
```

Un entier naturel  $n$  est premier si et seulement si  $n \geq 2$  et  $n$  n'admet que 1 et  $n$  comme diviseurs.

```
[5]: def is_prime(n):
     if n >= 2: return smallest_divisor(n) == n
     else: return False
```

```
[6]: print(is_prime(1234567891))
     print(is_prime(1234567893))
```

True  
False

```
[7]: print([n for n in range(100) if is_prime(n)])
```

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

La fonction `primes_from_to` prend deux entiers  $m$  et  $n$  en paramètres et renvoie la liste des nombres premiers dans l'intervalle  $[m, n]$ .

```
[8]: def primes_from_to(m, n):  
    s = []  
    for k in range(m, n + 1):  
        if is_prime(k):  
            s.append(k)  
    return s
```

```
[9]: primes = primes_from_to(1, 100)  
print(primes)
```

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

## 1.2 2. Le théorème des nombres premiers

Pour tout  $x \geq 1$ , notons  $\pi(x)$  le nombre de nombres premiers inférieurs ou égaux à  $x$ . On a  $\pi(2) = 1, \pi(3) = 2, \pi(4) = 2, \pi(5) = 3$ , etc.

**Théorème :** On a

$$\pi(x) \sim \frac{x}{\ln x}$$

Cet équivalent a tout d'abord été conjecturé par Adrien Marie Legendre à la fin du XVIIIe siècle. L'histoire de sa démonstration est plutôt longue, mais s'est achevée de façon heureuse : le théorème a été démontré indépendamment en 1896 par Jacques Hadamard et Charles Jean de la Vallée Poussin. Sa démonstration originale fait intervenir des résultats sur les fonctions de variables complexe. Une démonstration "élémentaire" (c'est à dire ne passant pas par le plan complexe) en a été faite en 1948 par, entre-autres, Paul Erdős.

Voyons un peu ce que donne concrètement tout cela. La fonction `pi` prend un entier  $n$  en paramètre et renvoie la liste des  $\pi(k)$  pour  $k = 2, \dots, n$ . Un compteur est mis au départ à 0. Chaque fois qu'un nombre premier est rencontré, on incrémente le compteur. Évidemment, cette fonction ne fonctionne que pour des petites valeurs de  $n$ .

```
[10]: def pi(n):  
    s = []  
    c = 0
```

```

for k in range(2, n + 1):
    if is_prime(k): c += 1
    s.append(c)
return s

```

```
[11]: print(pi(100))
```

```

[1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, 6, 6, 7, 7, 8, 8, 8, 8, 9, 9, 9, 9, 9,
9, 10, 10, 11, 11, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 14, 14, 14, 14, 15,
15, 15, 15, 15, 16, 16, 16, 16, 16, 16, 17, 17, 18, 18, 18, 18, 18, 18, 19,
19, 19, 19, 20, 20, 21, 21, 21, 21, 21, 21, 22, 22, 22, 22, 23, 23, 23, 23, 23,
23, 24, 24, 24, 24, 24, 24, 24, 24, 24, 25, 25, 25, 25]

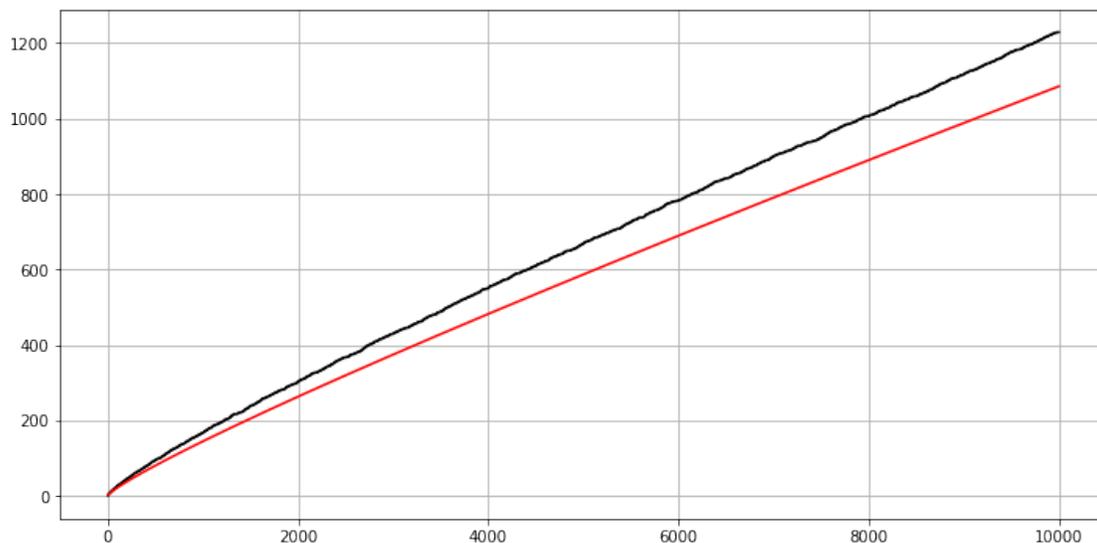
```

Traçons sur un même graphe  $\pi(x)$  et  $\frac{x}{\ln x}$ .

```

[12]: N = 10000
ks = range(2, N + 1)
plt.plot(ks, pi(N), 'k')
plt.plot(ks, [k / log(k) for k in ks], 'r')
plt.grid()

```



Ce n'est pas mal mais ce n'est pas si bien que cela. En fait, un « meilleur équivalent » de  $\pi(x)$  est

$$Li(x) = \int_2^x \frac{dt}{\ln t}$$

Mais c'est quoi un meilleur équivalent ? Après tout, tous les équivalents sont ... équivalents ? Cela signifie que

$$\pi(x) = Li(x) + o(Li(x))$$

où le petit o est un petit o du petit o qui apparaît dans

$$\pi(x) = \frac{x}{\ln x} + o\left(\frac{x}{\ln x}\right)$$

Ma phrase est rigolote mais incompréhensible. Pour être plus clairs :

$$\pi(x) - Li(x) = o\left(\pi(x) - \frac{x}{\ln x}\right)$$

Lowell Schoenfeld a démontré en 1976 que si une certaine conjecture appelée **l'hypothèse de Riemann** est vérifiée (voir plus loin !), alors, pour tout  $x \geq 2657$  on a

$$|\pi(x) - Li(x)| \leq \frac{\sqrt{x} \ln x}{8\pi}$$

Oui, bon, mais cette intégrale ne peut pas s'évaluer à l'aide des fonctions usuelles ? Eh bien non, mais il s'avère qu'on a un développement asymptotique de  $Li(x)$ .

**Proposition :** On a

$$Li(x) = \sum_{k=1}^n (k-1)! \frac{x}{\ln^k x} + o\left(\frac{x}{\ln^n x}\right)$$

**Démonstration :** Intégrer par parties.

Remarquons que

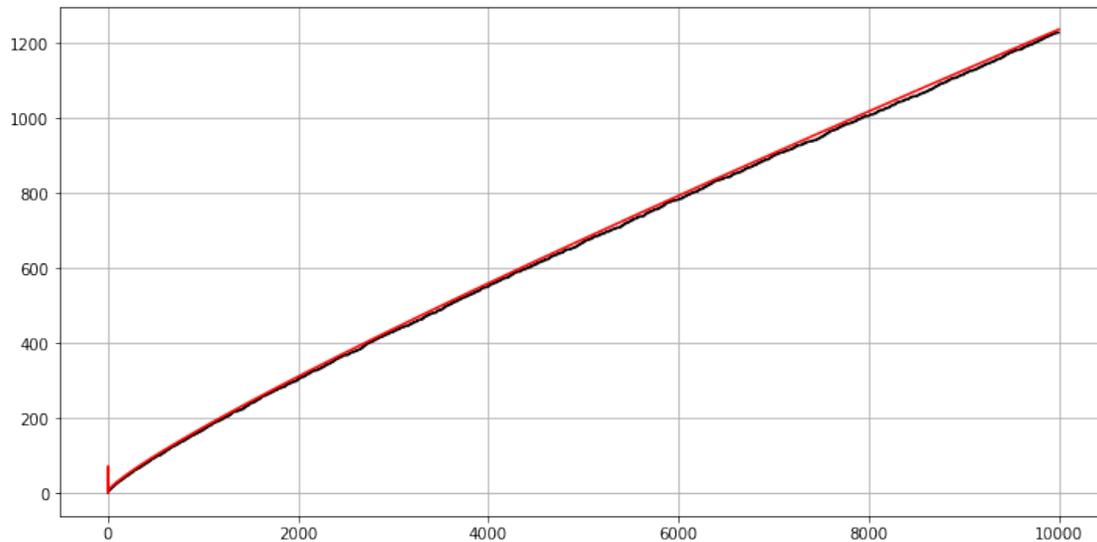
$$\frac{\sqrt{x} \ln x}{\frac{x}{\ln^n x}} = \frac{\ln^{n-1} x}{\sqrt{x}} \rightarrow 0$$

orsque  $x$  tend vers l'infini. En d'autres termes, le majorant de Schoenfeld est meilleur que toute approximation que nous pourrions obtenir à l'aide du développement asymptotique de  $Li(x)$ . Difficile de faire mieux ...

La fonction `approx_Li` prend un réel  $x$  et un entier  $n$  en paramètre et renvoie la somme des  $n$  premiers termes du développement asymptotique ci-dessus.

```
[13]: def approx_Li(x, n):
    s = 0
    l = log(x)
    p = 1
    for k in range(1, n + 1):
        s = s + p * x / l ** k
        p = p * k
    return s
```

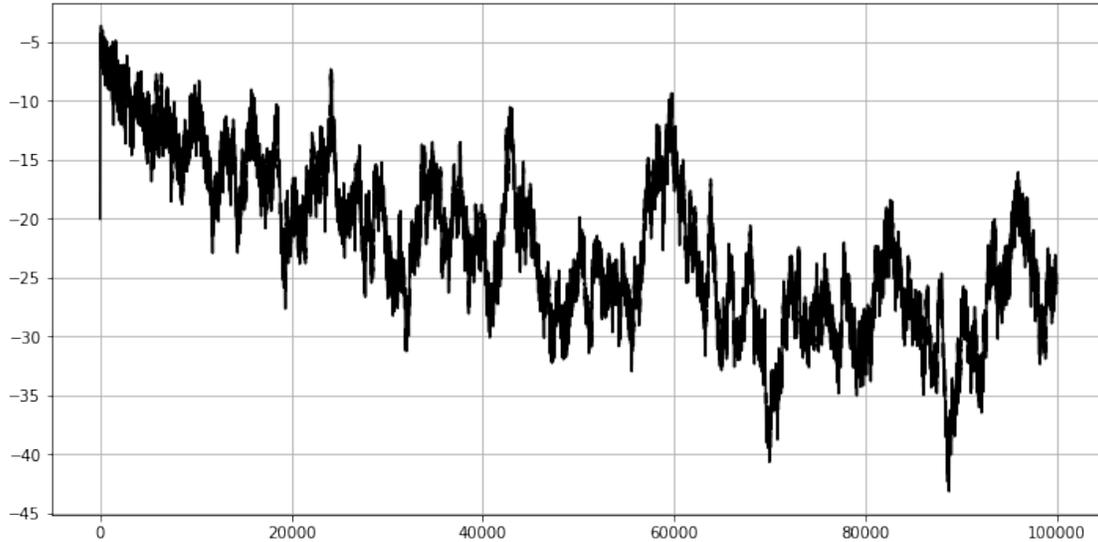
```
[14]: N = 10000
plt.plot(pi(N), 'k')
plt.plot([0,0] + [approx_Li(k, 4) for k in range(2, N)], 'r')
plt.grid()
```



C'est remarquable. Reprenez les valeurs dans la cellule ci-dessus et faites vos propres tests. C'est tout l'intérêt des notebooks ...

Un graphique encore plus parlant est celui de l'erreur commise en approchant  $\pi(x)$  par  $Li(x)$ . Le voici.

```
[15]: N = 100000
s1 = pi(N)
s2 = [approx_Li(k, 5) for k in range(2, N + 1)]
s = [s1[k] - s2[k] for k in range(3, len(s1))]
plt.plot(s, 'k')
plt.grid()
```



Remarquez la petitesse de l'erreur : au plus une quarantaine. La majoration de Schoenfeld donne quant à elle :

```
[16]: sqrt(1e5) * log(1e5) / (8 * 3.1415926536)
```

```
[16]: 144.85911691693855
```

Je vous laisse essayer avec  $N = 10^6$ . Remarquons que l'erreur commise est plus ou moins croissante. C'est normal : regardez le petit  $o$  dans le développement asymptotique, c'est quelque chose de négligeable devant un truc qui tend vers l'infini ! Si vous espériez obtenir une somme "convergente", vous serez déçus. Les nombres premiers ne se laissent pas attraper aussi facilement.

### 1.3 3. Le $n$ ième nombre premier

#### 1.3.1 3.1 Un équivalent de $p_n$

Nous y voilà. Que vaut  $p_n$ , le  $n$ -ième nombre premier ?

**Théorème :** On a

$$p_n \sim n \ln n$$

**Démonstration :** on a

$$\pi(p_n) = n \sim \frac{p_n}{\ln p_n}$$

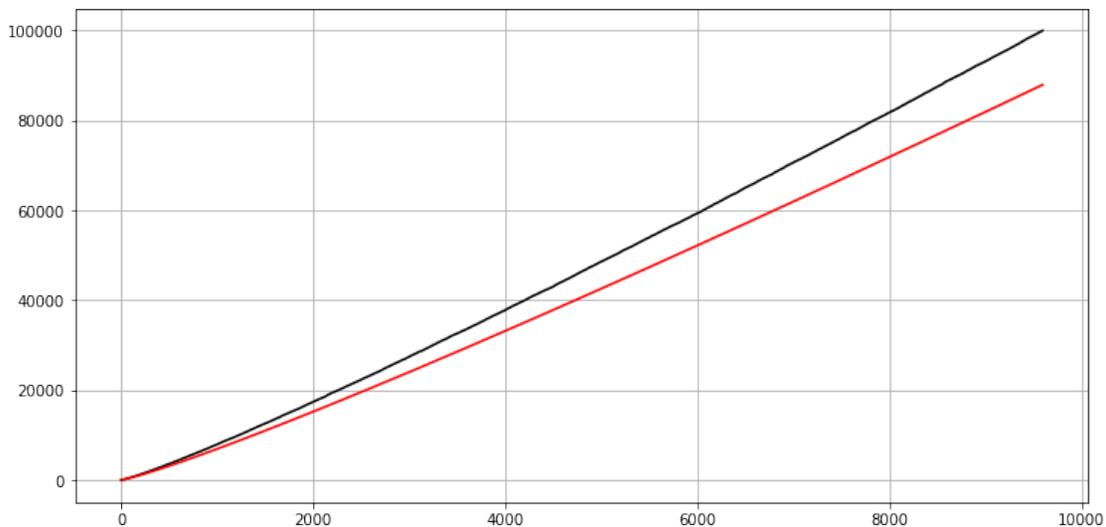
Ainsi,  $p_n \sim n \ln p_n$ . Mais on peut prendre des logarithmes dans des équivalents qui tendent vers l'infini. Donc

$$\ln n \sim \ln \left( \frac{p_n}{\ln p_n} \right) = \ln p_n - \ln \ln p_n \sim \ln p_n$$

d'où le résultat.

Une petite simulation ? En noir, le  $n$ ème nombre premier. En rouge,  $n \ln n$ .

```
[17]: N = 100000
primes = primes_from_to(2, N)
s = [t * log(t) for t in range(2, len(primes))]
plt.plot(primes, 'k')
plt.plot([0,0] + s, 'r')
plt.grid()
```



Oui, bon, c'est un peu comme  $\pi(n)$ . Pas mal mais peut mieux faire. Vu que  $\pi(p_n) = n \sim Li(p_n)$  est une meilleure approximation, on se doute que si on calcule la réciproque de la fonction  $Li$  on sera sur la bonne voie. Cette fonction est appelée *ali* par les experts. On a donc  $p_n \sim ali(n)$ . Mais comment calculer la réciproque d'une fonction que l'on ne sait pas elle-même calculer ?

### 1.3.2 3.2 Quelques résultats

Voici sans preuve (on s'en doute, ce n'est pas trivial), un développement asymptotique de  $p_n$ .

**Théorème :** On a

$$p_n = n \ln n \left( 1 + \sum_{k=1}^N \frac{P_{k-1}(\ln \ln n)}{\ln^k n} + O \left( n \left( \frac{\ln \ln n}{\ln n} \right)^N \right) \right)$$

où les  $P_k$  sont des polynômes, de degré  $\leq k$  si  $k \geq 1$ . On a en particulier :

- $P_0 = X - 1$
- $P_1 = X - 2$
- $P_2 = \frac{1}{2}(X^2 - 6X + 11)$

Testons.

```
[18]: def f(n):
      if n <= 2: return 0
      else:
          u = log(n)
          x = log(u)
          P0 = x - 1
          P1 = x - 2
          P2 = -(x ** 2 - 6 * x + 11) / 2
          return n * (u + P0 + P1 / u + P2 / (2 * u ** 2))
```

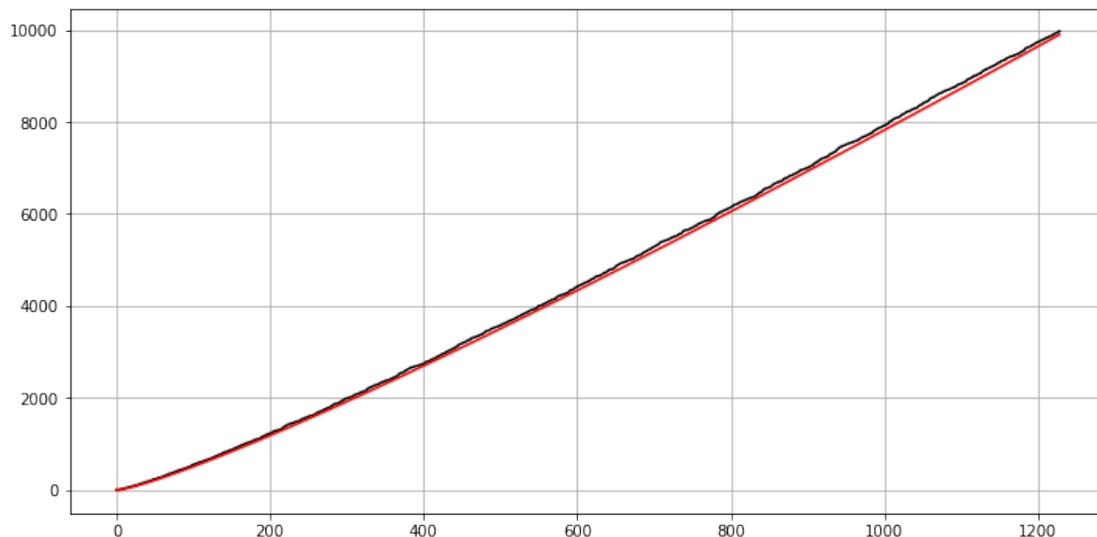
Par exemple, le 2  $10^{17}$ ème nombre premier est (paraît-il :-)) 8512677386048191063.

```
[19]: f(2e17)
```

```
[19]: 8.512759102518627e+18
```

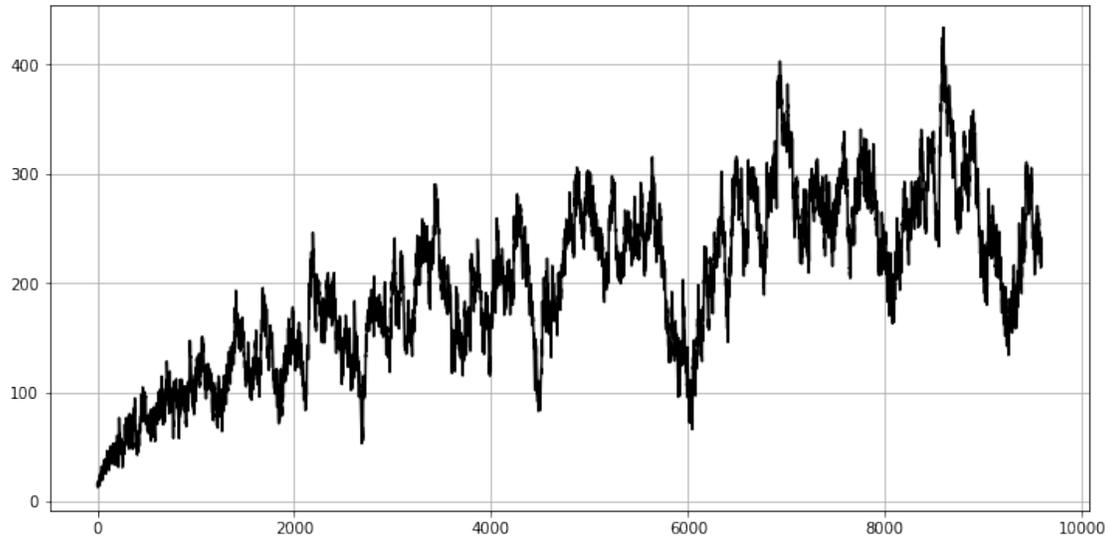
Une erreur relative de l'ordre de  $10^{-4}$  ...

```
[20]: N = 10000
      primes = primes_from_to(2, N)
      s = [f(k + 1) for k in range(len(primes))]
      plt.plot(primes, 'k')
      plt.plot(s, 'r')
      plt.grid()
```



Les deux courbes se superposent quasiment. Enfin, traçon le graphe de l'erreur commise.

```
[21]: N = 100000
primes = primes_from_to(2, N)
s = [f(t + 1) for t in range(len(primes))]
s1 = [abs(primes[k] - s[k]) for k in range(2, len(primes))]
plt.plot(s1, 'k')
plt.grid()
```



L'erreur ne dépasse pas ici quelques centaines.

### 1.3.3 3.3 Une majoration théorique de l'erreur

L'**hypothèse de Riemann** affirme que les zéros complexes non triviaux d'une certaine fonction (la fonction zeta) sont tous situés sur une certaine droite. Cette conjecture a été énoncée par B. Riemann en 1859, elle est à ce jour non démontrée. Cela dit, beaucoup d'énoncés d'arithmétique commencent par "Si l'hypothèse de Riemann est vérifiée alors" ... Conclusion :

- Si quelqu'un démontre un jour que l'hypothèse de Riemann est vraie nous aurons plein de théorèmes démontrés d'un coup.
- Si quelqu'un démontre un jour que l'hypothèse de Riemann est fausse nous aurons plein de théorèmes à mettre à la poubelle, ou, plutôt, plein de théorèmes qu'il nous faudra essayer de prouver autrement.

**Théorème** : Si l'Hypothèse de Riemann est vérifiée, alors pour tout  $n \geq 11$

$$|p_n - ali(n)| \leq \frac{1}{\pi} \sqrt{n} (\ln n)^{\frac{5}{2}}$$

## 1.4 4. Conclusion

Un notebook sur la fonction  $\zeta$  semble tout indiqué.