Vérification de la conjecture de Proth-Gilbraith jusqu'à $\pi(10^{14})$ et méthodes pour anticiper et minorer les résultats de vérification au-delà

Simon Plouffe
6 octobre 2025

Résumé

Un calcul a été effectué pour vérifier la conjecture de Proth-Gilbraith pour tous les nombres premiers jusqu'à 10^{14}. Le calcul précédent avait été effectué par Andrew Odlyzko en 1993 jusqu'à 10^{13}.
Il s'agit de calculer les différences entre premiers consécutifs en valeur absolue et à recommencer. La conjecture affirme que toutes les lignes sauf la première commencent par 1. Pour l'établir il suffit de trouver une ligne commençant par 1 et suivie uniquement de 0 et de 2.

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, …
1, 2, 2, 4, 2, 4, 2, 4, 6, 2, …
1, 0, 2, 2, 2, 2, 2, 2, 4, …
1, 2, 0, 0, 0, 0, 0, 2, …
1, 2, 0, 0, 0, 0, 2, …
1, 2, 0, 0, 0, 2, …
1, 2, 0, 0, 2, …

La conjecture a été vérifiée par A. Odlyzko en partant de la ligne composée de tous les nombres premiers inférieurs ou égaux à 10^{13}. Une question suggérée par Jean-Paul Delahaye était de savoir si en 2025 on pouvait faire mieux et vérifier la conjecture pour la suite des nombres premiers jusqu'à 10^{14}. Dans un premier temps j'ai pu aller jusqu'à 2×10^{13}. J'ai ensuite pu aller jusqu'à 10^{14}. Le calcul s'est terminé le 7 octobre 2025 à 14h25. L'article contient diverses informations et statistiques sur le calcul et sur les méthodes qu'on peut utiliser pour anticiper le nombre de lignes avant d'arriver à celle comportant un 1 suivi de 0 et de 2.

Note : Un calcul similaire est en cours par Jean-François Colonna Chercheur, Centre de mathématiques appliquées, École polytechnique

Introduction

Estimation du travail à effectuer

Il y a 3204941750802 nombres premiers jusqu'à 10^{14} (3204 milliards) et on doit effectuer les différences entre ces nombres jusqu'à la ligne finale. Un test avec des blocs de 10 millions de nombres premiers successifs pris à différents endroits montre que le nombre moyen d'opérations de différence avant d'arriver à la ligne finale est de l'ordre de 400. Il est déraisonnable de faire le calcul sur l'ensemble de tous ces nombres premiers en un seul bloc. Une bonne idée est de couper par blocs de 10 millions de nombres premiers et d'y ajouter un chevauchement pour chaque bloc de telle sorte que les calculs peuvent être faits indépendamment. Sachant que le nombre maximal d'opérations par bloc est de quelques centaines en prenant un chevauchement de plusieurs milliers de valeurs (3000 à 4000) on est assuré de couvrir toute l'étendue entre 2 et 10^{14}. De cette façon les calculs peuvent être effectués de façon indépendante et ou en parallèle sur une machine. Avec ces informations on peut estimer que le nombre total d'opérations est de l'ordre de 1.2×10^{15}.

Une première tentative avec un programme Maple montre que c'est largement trop lent, il aurait fallu 271 jours de calcul sur 2 ordinateurs en continu. Suite à cet échec un programme a été créé utilisant des scripts Unix et une série de commandes natives à ce langage. Résultat : c'est presqu'aussi lent qu'en Maple (241 jours de calcul).

L'alternative qui a été trouvée est d'utiliser la librairie numpy de Python (3.12) et le programme < diff > qui permet d'effectuer l'opération à très grande vitesse. De plus on peut contrôler la taille des entiers qui interviennent dans le calcul. Ici des nombres de 16 bits, (max 32768) ont été utilisés ce qui fait gagner un temps appréciable.

Muni de ce programme spécial sur 2 ordinateurs rapides :

Intel I9-9900KF 8 cœurs/16 threads à 5 Ghz et 32 gigas de mémoire
Intel I7-6700K 4 cœurs / 8 threads à 4 Ghz et 32 gigas de mémoire.

L'opération de différence sur les gros tableaux de nombres s'effectue à la vitesse de 2 milliards de calculs par seconde ce qui est 24 fois plus rapide que Maple ou les scripts Unix. Au final le calcul complet a pris 11 jours seulement.

Les premiers ont été générés avec le programme *primesieve* qui est le plus rapide connu pour générer des nombres premiers jusqu'à $2^{64} - 1$. Pour économiser de l'espace, le calcul est effectué en flux tendu (expliqué plus loin), chaque fichier généré est immédiatement traité, seul le résultat du nombre d'étapes du calcul pour arriver à la ligne finale est conservé. Pour garder 3204 milliards de nombres premiers il faut 25 téraoctets. Certains calculs ont été effectués par Maple où avec un script pour vérifier ponctuellement la validité du calcul.

Plan de calcul

Comme l'a suggéré Jean-Paul Delahaye on peut couper en tranches le travail en s'assurant de la validité de l'opération grâce au chevauchement des listes de nombres premiers. Tout d'abord on choisit quelle est la largeur d'une tranche : ici 10 millions de premiers, on choisit également quel sera la taille du chevauchement, entre 3000 et 4000 nombres premiers. En calculant le triangle de Proth-Gilbraith (opérations successives des différences absolues) de 2 tranches successives on arrive à la fin du calcul quand la ligne qui reste ne comporte que des 0 ou des 2 : seule la première tranche commence par 1. Si les tranches finales une fois le calcul terminé se chevauchent avec suffisamment de valeurs alors le calcul est validé. De façon empirique : le nombre d'étapes que Odlyzko a obtenu pour son calcul a été 635, c'est la valeur de la fonction G : donc pour $G(\pi(10^{13})) = 635$. Ce qui veut dire qu'on devrait s'attendre à quelques centaines d'étapes au mieux. La valeur maximale rencontrée pour $G(\pi(10^{14}))$ a été de 693. Donc étant donné que le chevauchement est constitué de plus de 3000 nombres premiers on est sûr qu'il n'y a pas de trous tant que la valeur de G ne dépasse pas 1000 ou 2000.

Si les tranches sont de 10 millions de premiers auquel on ajoute 3000 ou 4000 premiers pour le chevauchement on a donc 320495 tranches de premiers à traiter. Un programme dynamique a été mis au point pour traiter à la volée le tableau des différences avec au final une trace de chaque tranche une fois terminée. En ordre on a pour chaque tranche de 10 millions de premiers + chevauchement.

1- Création de la liste des premiers
2- Programme *awk* pour effectuer la différence entre 2 premiers à la volée ou en flux tendu, le programme compare 2 lignes successives à mesure que le fichier des premiers est rempli. Le fichier est d'abord généré et envoyé directement au fil de l'eau par la commande awk mais précédé du < | >.
3- Lancement du programme Python avec numpy.diff (np.diff en plus court).
4- On garde la trace du calcul pour valider le nombre d'étapes nécessaires pour chaque tranche (320495 fichiers).

Nomenclature :

G(tranche de nombres premiers) = nombre d'étapes pour se rendre à une ligne ne contentant que des 0 et 2, sauf la première tranche qui commence par 1.

G'(tranche de nombres premiers) = nombre d'étapes pour se rendre à une ligne ne contentant que des 0 et 2. Le calcul est effectué sur la plus grande valeur connue M. Des tables des records d'écarts entre les premiers existent déjà. De cette façon la valeur de 693 a déjà été détectée le 24 septembre comme étant la très possible valeur finale de $G(\pi(10^{14}))$ parce que la valeur de M à un endroit déjà connu est de 766. Plusieurs sources confirment ce maximum M après le nombre premier : 19581334192423 et qui a été découvert en 1989.

M(tranche de nombre premiers) = maximum obtenu suite à l'opération différence effectuée 1 fois, La valeur de M est le maximum entre tous les premiers de la tranche considérée. Ici le maximum rencontré a été de 766 sur toute l'étendue entre 2 et 10^{14}.

R(tranche de nombres premiers) = ratio entre G(tranche)/M(tranche).

Par exemple G(tranche # 25127) = 635 est la valeur trouvée en 1993 par Odlyzko et qui correspond à l'écart après le nombre premier 7177162511713. La valeur de 693 correspond à l'écart de 766 après le nombre premier 19581334192423. On a G(tranche 66233) = 693 qui est la valeur maximale sur toute l'étendue : G($\pi(10^{14})$).

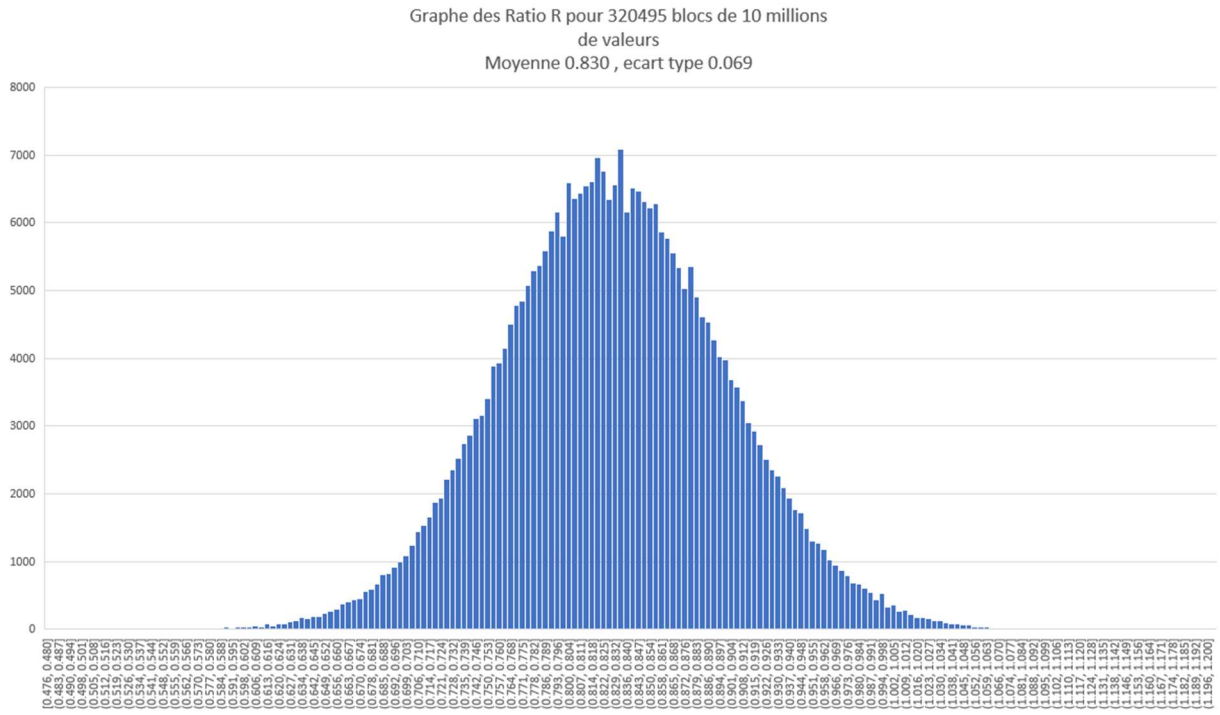
Résultats et statistiques

On se rend compte assez vite qu'il y a un lien direct entre la valeur de G et celle de M, la valeur R est en moyenne 0.830 et très peu de valeurs de R dépassent 1 : 99.2 % des valeurs sont ≤ 1.

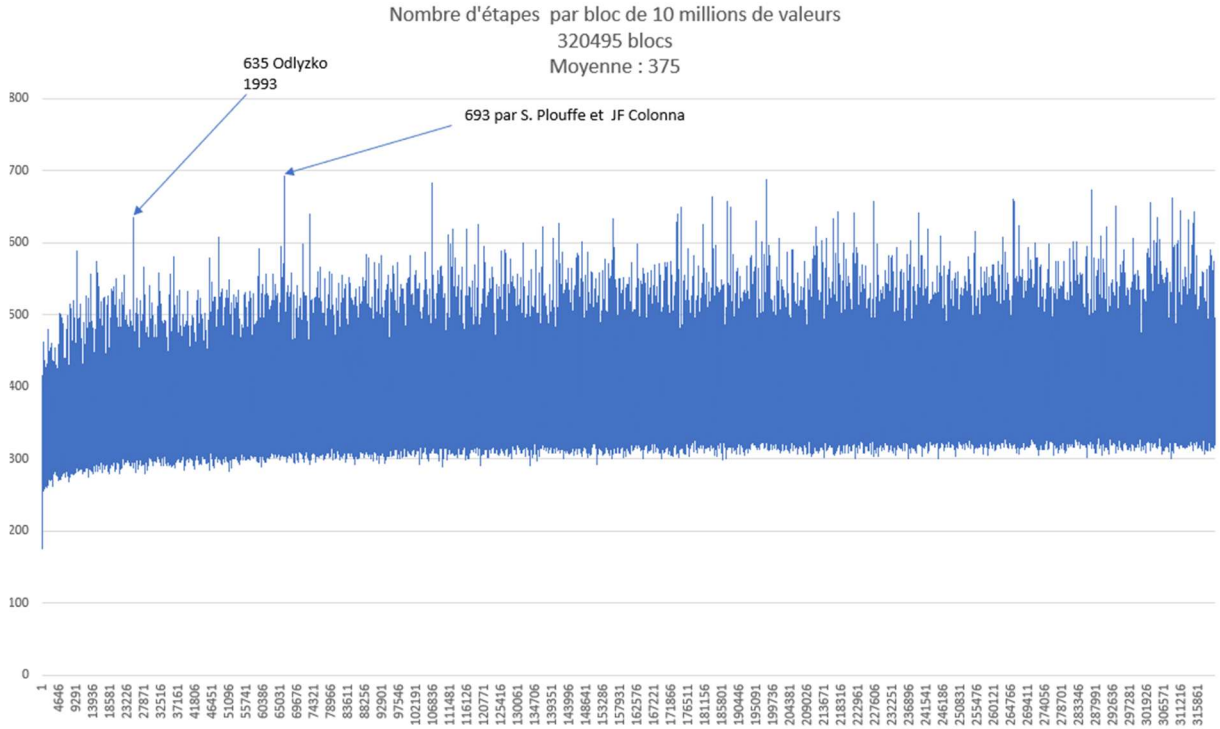
La valeur maximale trouvée de 693 a comme valeur de M qui est 766, soit le maximum de la plage une fois faite la première différence entre les premiers. Le ratio R est donc 693/766 = .9046. Cette valeur a été récemment validée par le calcul complet de JF Colonna et l'auteur de cet article.

Une autre conséquence de cette moyenne est qu'on peut prédire la valeur de G selon la valeur de M. Effectivement dans la table [7] on trouve l'écart 906 après le nombre 218209405436543 trouvé par T. Nicely en 1996. La valeur de G est 773. Ici le ratio est 0.8532. On peut aussi consulter la liste des écarts connus record sur wikipedia (Prime gap) et aussi sur d'autres sources.

Voici un histogramme des valeurs de R calculées sur 320495 valeurs de G et M. En abscisse la valeur de R et en ordonnée le nombre de cas avec cette valeur. On voit rapidement que la grande majorité des valeurs sont ≤ 1, plus de 99.2 %.



Graphe des valeurs de G, on reconnait les 2 points maximums connus, le premier à 635 tel que calculé par Odlyzko en 1993 (revérifié ensuite) et la valeur de 693 détectée maintenant. La valeur de G moyenne est de 375 sur cet échantillonnage.



Il est assez évident que la valeur de G pour une tranche donnée ou pour tout l'intervalle jusqu'à $\pi(10^{14})$ dépend directement de la plus haute valeur M de cet intervalle. En effet sur la plage de tous les premiers jusqu'à 10^{14}, le maximum connu est de 766 qui est la plus grande différence (Prime gap) connue. Conséquemment, le calcul de G autour de cette valeur donne effectivement la valeur de G maximale pour tout l'intervalle. Donc, connaissant ces valeurs M : liste des écarts entre nombres premiers record, il est assez facile de calculer directement la valeur G'(intervalle) qui sera probablement la valeur finale pour tout l'intervalle. Ceci pour une raison simple : Pour une valeur donnée de M la quantité G est en moyenne (calculé sur 320495 valeurs) est 0.830 et que la valeur maximale dépasse 1 dans seulement 0.8 % des cas. Quand la valeur dépasse 1 elle est au plus 1.20 donc si on a la valeur M maximale connue sur un nombre premier il y a de fortes chances que le calcul de G'(intervalle) donne la valeur finale pour G. La liste en [13] a été passée au complet pour calculer toutes les valeurs de G'($\pi(10^k)$), pour k = 12 à 20. On a constaté également que les valeurs connues de M maximales dans la table [13] correspondent exactement aux valeurs trouvées par Odlyzko en 1993 [3].

Ici la table des valeurs de G'($\pi(10^k)$).

K	Valeur de M	Valeur de G'	Ratio	Premier
15	880	788	0.8954	277900416000927
16	1132	970	0.8560	1693182318746371
17	1184	1199	1.012	43841547845541059
18	1358	1274	0.9381	523255220614645319
19	1446	1433	0.9910	9656919634106230133
20	1402	1462	1.042	10103695526434940251

On pourrait pousser le calcul plus loin mais pour ce faire : on doit avoir une liste d'écarts entre les premiers qui soient le plus petit possible, la liste qui apparaît sur plusieurs sources permet d'aller jusqu'à G'($\pi(10^{20})$), après cette marque c'est plus spéculatif. On a des écarts connus jusqu'à des premiers de 87 chiffres mais on ne sait pas si le nombre premier trouvé est le plus petit. En conséquence on ne peut donner qu'une indication sur le minimum de G'($\pi(10^m)$) où m > 20.

Références :

[1] François Proth, « Sur la série des nombres premiers », *Nouv. Corresp. Math*, vol. 4, 1878, p. 236-240.
[2] Chris Cladwell, « Gilbreath's conjecture [archive] », sur Prime Pages.
[3] Andrew Odlyzko, « Iterated absolute values of differences of consecutive primes », *Math. Comp.*, vol. 61, 1993, p. 373-380.
[4] Différence entre les nombres premiers sur Wikipedia :
https://fr.wikipedia.org/wiki/%C3%89cart_entre_nombres_premiers
[5] Prime Gap (plus complet): https://en.wikipedia.org/wiki/Prime_gap
[6] Thomas Nicely, Tables of prime gaps
https://faculty.lynchburg.edu/~nicely/index.html#TPG
[7] Thomas Nicely: https://faculty.lynchburg.edu/~nicely/gaps/gaplist.html
[8] Terrence Tao: Small and large gaps between primes:
https://terrytao.wordpress.com/wp-content/uploads/2015/07/lat.pdf
[9] Primesieve, primecount, primesum de Kim Walisch:
https://github.com/kimwalisch/primesieve
[10] Numpy et le sous-programme diff en python :
https://numpy.org/devdocs/reference/generated/numpy.diff.html
[11] Jean-Paul Delahaye, communication personnelle.
[12] Prime Gap records: https://primegap-list-project.github.io/lists/
[13] Prime gaps first occurences: https://primegap-list-project.github.io/lists/prime-gaps-first-occurrences/
[14] https://www.pzktupel.de/RecordGaps/risinggap.php : liste d'écarts entre premiers allant jusqu'à au moins 10^{87}.