

Reprise du travail sur les fonctions sphéroïdales prolates, pour réussir à trouver par programme les résultats de Richard-Jung, Ramis, Thomann et Fauvet, Denise Vella-Chemla, 14 décembre 2024

Je cherche depuis longtemps à comprendre l'approche qu'a Alain Connes de l'hypothèse de Riemann et la vision qu'il a de l'ensemble des nombres premiers. Informaticienne de formation, je comprends en programmant ; en faisant cela, je suis le conseil du Professeur Donald Ervin Knuth, que j'admire entre tous, et qui dit qu'on ne comprend bien une notion que le jour où on arrive à l'enseigner à une machine. Qu'on ne se leurre pas, je suis loin de toute compréhension ici, vraiment, mais le fait d'avoir un programme qui commence à se rapprocher doucement de l'objectif fait le plus grand bien, surtout quand on a vraiment le sentiment d'avoir lutté pour l'écrire.

Donc voici : il s'agit de voir les zéros de la fonction ζ de Riemann comme les valeurs propres d'un opérateur. C'est-à-dire qu'on dispose d'un certain nombre de valeurs, les parties imaginaires des zéros non triviaux de la fonction ζ , et on cherche une fonction qui enverrait les entiers sur ces valeurs. Alain Connes a pensé à l'opérateur prolate en 1997 pour permettre d'atteindre les zéros de ζ parce que cet opérateur a notamment comme propriété intéressante de permettre de limiter et une fonction et sa transformée de Fourier, c'est ce que j'ai entendu à plusieurs reprises : il permet de limiter la bande passante à la fois de la coordonnée temporelle et de la coordonnée fréquentielle d'un signal ; habituellement, cela n'est pas possible.

Les références souvent citées concernant les fonctions prolates sont les articles de Slepian, Landau et Pollack. Une femme a travaillé avec eux, il s'agit d'Estelle Sonnenblick. Non seulement, elle a fourni toute une série de tableaux de référence pour les valeurs propres λ_n et les χ_n correspondantes, et Slepian écrit qu'il a une dette à son égard d'avoir effectué tous ces calculs sur ordinateur, mais elle fournit également des fonctions d'approximation de ces valeurs.

Ci-après un graphique que l'on voit fréquemment associé aux fonctions sphéroïdales prolates.

On teste les fonctions d'approximation de l'article de Sonnenblick et Slepian [4] par le programme fourni en annexe 1, c'est à dire qu'on vérifie en les comparant aux valeurs des tableaux qui sont fournis dans ce même article que les approximations sont assez correctes. Alors on utilise les fonctions d'approximation pour calculer les valeurs dont Alain Connes, qui les a fournies récemment dans une conférence ¹, a expliqué qu'elles provenaient d'un article de Richard-Jung, Ramis, Thomann, et Fauvet : Richard-Jung et al. ont trouvé des polynômes pour calculer les fonctions prolates plus simplement car ces fonctions sont très complexes à calculer.

Voici les valeurs fournies des 31-ièmes valeurs propres obtenues par les fonctions de Richard-Jung, Ramis, Thomann, Fauvet [5], à lire par ligne, elles sont croissantes.

¹Il s'agit d'une conférence donnée le 20 novembre 2024, pendant le Séminaire Nord-Atlantique de Géométrie non-commutative, qui a été organisé par les chercheurs de l'IMPAN à Varsovie - et qui a été l'occasion de fêter l'anniversaire de Piotr Hajac.

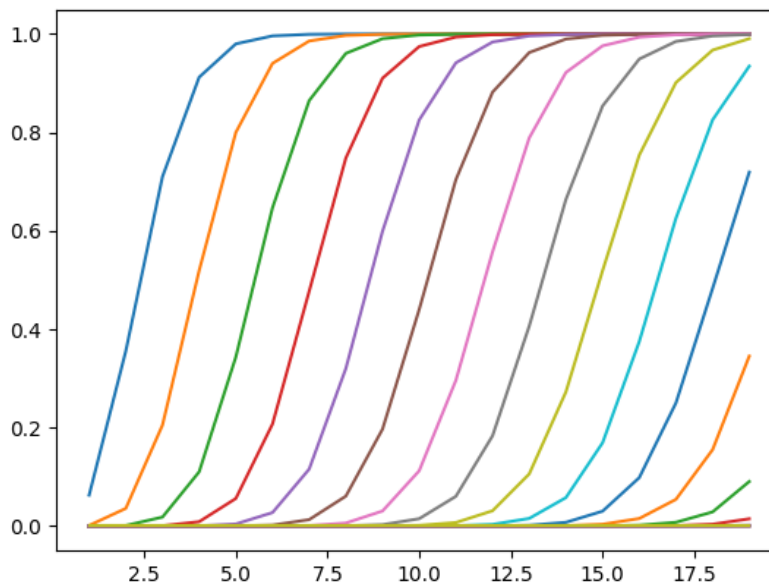
Les 14 premières 31-ièmes valeurs propres :

-39.383216574467	-94.280833772513	-152.569476972276	-214.006491323911
-278.401004621174	-324.597681671966	-415.466848582142	-487.898087149855
-562.795870371439	-640.076470907495	-719.665701171603	-801.497217687773
-885.511220599887	-971.653437429483		

Les 31-ièmes valeurs propres, de la 74^{ième} à la 78^{ième} :

-9104.33317141284	-9281.09333576484	-9459.04366446659	9638.178978345020
-9818.49419143022			

Ces valeurs propres avaient préalablement été calculées à la main par Alain Connes et Henri Moscovici avec l'outil Mathematica, du moins leur partie entière approximative, car il s'agissait des valeurs sur lesquelles deux fonctions (à retrouver) étaient en phase ou en opposition de phase.



Voici les valeurs correspondantes (les 31-ièmes valeurs propres) calculées par les formules d'approximation fournies dans l'article d'Estelle Sonnenblick.

Les 14 premières 31-ièmes valeurs propres, approximées par Estelle Sonnenblick :

30.243951612903224	91.21975806451613	151.15927419354838	210.0383064516129
267.83266129032256	324.5181451612903	380.070564516129	434.4657258064516
487.679435483871	539.6875	590.4657258064516	639.9899193548387
688.2358870967741	735.179435483871		

Les 31-ièmes valeurs propres, de la 74^{ième} à la 78^{ième}, approximées par Estelle Sonnenblick :

991.6260080645161	991.6260080645161	991.6260080645161	991.6260080645161
991.6260080645161			

Le passage de ces valeurs propres aux valeurs des zéros de ζ s'effectue en leur appliquant la fonction :

$$f(x) : x \mapsto \begin{cases} \mathfrak{I}(2\sqrt{-x}) & \text{si } x > 0 \\ \mathfrak{I}(2\sqrt{x}) & \text{si } x < 0 \end{cases}$$

On reporte les valeurs calculées par l'approximation d'Estelle Sonnenblick et celles calculées par Richard-Yung, Ramis, Thomann et Fauvet dans un tableau avec calcul de la fonction f pour les comparer entre elles et les comparer aux parties imaginaires des zéros non triviaux de ζ .

Les 14 premières 31-ième valeurs propres par les deux formules :

n	$a = \chi_n$ (Sonnenblick et al.)	$f(a)$	$b = \chi_n$ (Richard et al.)	$f(b)$	$\zeta(x)$
1	30.243951612903224	10.998900238278956	-39.383216574467	12.551209754357068	14.134725142
2	91.21975806451613	19.101807041692798	-94.280833772513	19.419663619384657	21.022039639
3	151.15927419354838	24.589369588791687	-152.569476972276	24.703803510575128	25.01085758
4	210.0383064516129	28.98539676813915	-214.006491323911	29.257921411057964	30.424876126
5	267.83266129032256	32.731187652776825	-278.401004621174	33.370705993201526	32.935061588
6	324.5181451612903	36.028774342810514	-342.597681671966	37.01878883334602	37.586178159
7	380.070564516129	38.99079709450059	-415.466848582142	40.76600782917758	40.918719012
8	434.4657258064516	41.68768287187243	-487.898087149855	44.176830449902354	43.327073281
9	487.679435483871	44.16693041106076	-562.795870371439	47.44663825273352	48.005150881
10	539.6875	46.46235034950342	-640.076470907495	50.59946525043501	49.773832478
11	590.4657258064516	48.599001051727456	-719.665701171603	53.65317143176545	52.970321478
12	639.9899193548387	50.59604408863755	-801.497217687773	56.62145238998283	56.446247697
13	688.2358870967741	52.468500534959986	-885.511220599887	59.515081134108755	59.347044003
14	735.179435483871	54.22838502053592	-971.653437429483	62.34271208182984	60.831778525

Les 31-ièmes valeurs propres de la 74^{ième} à la 78^{ième} par les deux formules :

n	$a = \chi_n$ (Sonnenblick et al.)	$f(a)$	$b = \chi_n$ (Richard et al.)	$f(b)$	$\zeta(x)$
74	991.6260080645161	62.98018761688524	-9104.33317141284	190.83325885613166	189.416158656
75	991.6260080645161	62.98018761688524	-9281.09333576484	192.6768625005591	192.026656361
76	991.6260080645161	62.98018761688524	-9459.04366446659	194.51522988667585	193.079726604
77	991.6260080645161	62.98018761688524	-9638.178978345020	196.34845533739266	195.265396680
78	991.6260080645161	62.98018761688524	-9818.49419143022	198.17663022092407	196.876481841

On obtient des valeurs qui semblent au début similaires à celles de Richard-Jung et al. avec le programme de résolution simple de Sturm-Liouville ² et le calcul des valeurs propres fourni ci-dessous ; les valeurs semblent s'éloigner très vite de celles de Richard et al.

²Le calcul utilise une matrice proposé par Li-Lian Wang dans [7] (définie en (2.19)) ainsi que les formules (2.20) et (2.21) pour le calcul des fonctions prolate. Je remercie Jacques Chemla d'avoir programmé cet opérateur.

```

import matplotlib.pyplot as plt
import numpy as np
import math
from math import pi

def solve_SturmLiouville(c, n):
    N = int(1.1*c+n+20)
    A = np.zeros((N, N))
    for k in range(N):
        A[k, k] = k*(k+1) + (2*k*(k+1) - 1) / ((2*k+3)*(2*k-1)) * c**2
        if k+2 in range(N):
            A[k, k+2] = A[k+2, k] = (k+1)*(k+2) / ((2*k+3) * np.sqrt((2*k+1)*(2*k+5))) *
c**2
    chi, beta = np.linalg.eigh(A)
    return chi, beta, A

def prolate(c, n):
    chi, beta, A = solve_SturmLiouville(c, n)
    coef = np.diag([np.sqrt(k + 1/2) for k in range(len(beta))]) @ beta
    psi = [np.polynomial.Legendre(coef[:, k]) for k in range(n+1)]
    lmb = [(-1)**(k//2) * np.sqrt(2) * (
        beta[0, k] / psi[k](0) if k % 2 == 0 else
        beta[1, k] / psi[k].deriv()(0) * c / np.sqrt(3)
    ) for k in range(n+1)]
    return lmb, psi, chi, beta, A

with open("zeros200", 'r') as f:
    lines = f.readlines()
    zeros = [float(line) for line in lines]

c = 32
n = 64
lmb, psi, chi, beta, A = prolate(c, n)
for k in range(len(chi)):
    res1 = chi[k]
    res2 = 2*(res1**0.5)
    print(' ', res1, ' ', res2)
    plt.scatter(res1, res2, s=4, color='blue')
    plt.scatter(res1, zeros[k], s=4, color='red')
plt.show()

```

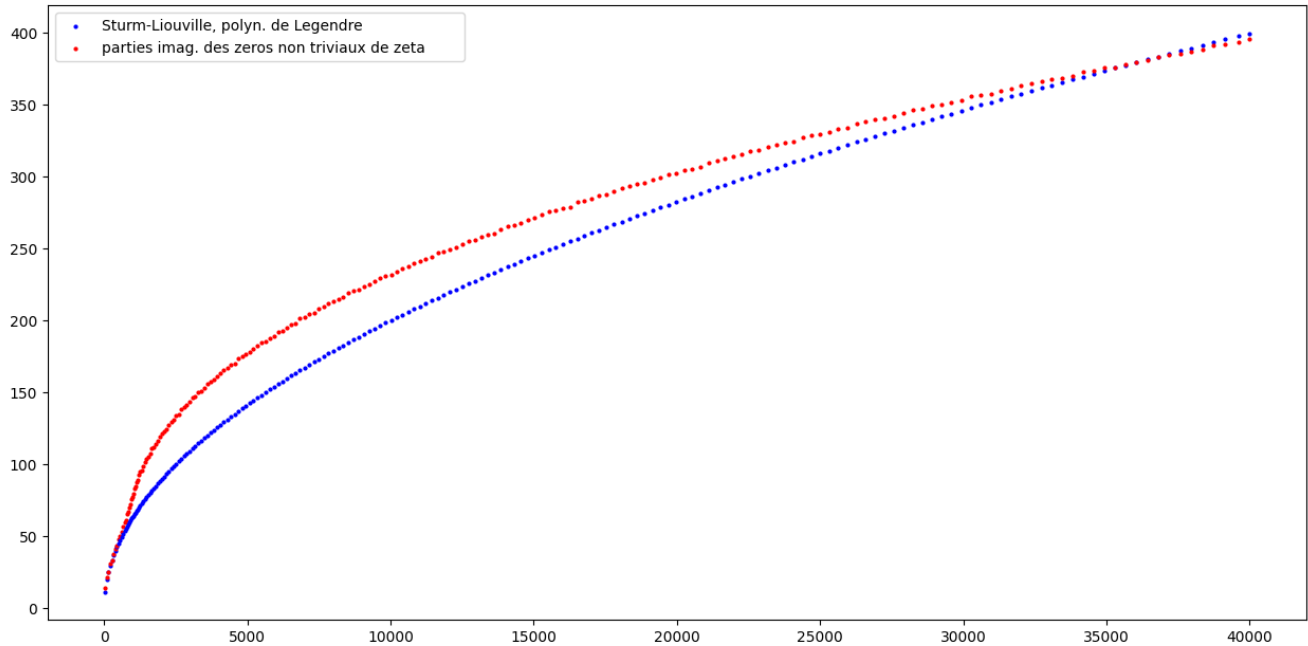
Les premières valeurs propres fournies sont les suivantes :

31.243897105576774	94.21896027377812	156.1552259401503
217.02479690312134	276.7971243661913	335.4385620950815
392.91180818935726	449.17519504826606	504.18176959991
557.8780759336684	610.2025019024709	661.0829609105459
710.4335062292904	758.149104360807	804.0968939761898
848.0998392527229	889.9026608776077	929.1041044387448
965.1026164111247	997.4723147032043	1027.3206204218318

On fournit les valeurs propres calculées par résolution de l'équation de Sturm-Liouville (en utilisant la matrice tridiagonale fournie en (19) dans [7] et les fonctions (20) et (21) de l'article en question.) en annexe 2 jusqu'à la 200^{ième} en les mettant en regard des parties imaginaires des zéros non triviaux

de ζ et on constate que si les valeurs diffèrent assez vite au début, elles ont tendance à se rapprocher petit à petit sur la durée.

Voici le comportement relatif des valeurs propres calculées par Sturm-Liouville et des parties imaginaires des zéros non triviaux de ζ .



On note ici qu'on a trouvé dans un article de França, LeClair [8]³ une autre manière d'approximer les parties imaginaires des zéros non triviaux de ζ

$$g(t) = \frac{2\pi \left(t - \frac{11}{8}\right)}{W\left(\frac{t - \frac{11}{8}}{e}\right)}$$

avec W la fonction de Lambert, i.e. la réciproque de la fonction $W^{\leftarrow}(z) = ze^z$, avec z complexe. On a également trouvé dans cet article la fonction de Backlund

$$\frac{1}{\pi} \arg \Gamma\left(\frac{1}{2} + \frac{t}{2}i\right) + 1 + \frac{1}{\pi} \arg \zeta\left(\frac{1}{2} + it\right)$$

qui permet de rapidement calculer le nombre de zéros inférieurs à une grandeur donnée, mais qui ne semble pas découler de concepts théoriques puissants, comme ceux de la géométrie non-commutative d'Alain Connes ou ceux de la théorie quantique (on a omis de dire que la vision "opérateur issu de la physique" pour les zéros de ζ (ou les nombres premiers) consiste en fait à transposer le problème dans la théorie quantique et à compter des états quantiques dont l'énergie est bornée par une certaine grandeur).

³Giuseppe Mussardo a également travaillé sur ce sujet avec eux.

Références

- [1] Alain Connes : Support de la conférence du 20 novembre 2024 : *From classfield theory to zeta spectral triples*, 20 novembre 2024, Séminaire Nord-Atlantique de Géométrie non-commutative 20241120-slides-Alain-Connes-anniv-Piotr.pdf (travail conjoint avec Caterina Consani et Henri Moscovici).
- [2] Alain Connes, Matilde Marcolli : traduction en français du § 3.3 du livre *Noncommutative geometry, quantum field and motives*, American Mathematical Society Colloquium Publications, American Mathematical Society, 2008, vol. 55, p. <https://denisevellachemla.eu/ACMM-zeta-quantique.pdf>.
- [3] Alain Connes, Henri Moscovici, *The UV prolate spectrum matches the zeros of zeta*, PNAS 2022, Vol. 119, n° 22, <https://www.pnas.org/doi/10.1073/pnas.2123174119>.
- [4] David Slepian, Estelle Sonnenblick, *Eigenvalues Associated with Prolate Spheroidal Wave Functions of Zero Order*, The Bell System Technical Journal, octobre 1965, p. 1745-1759 <https://denisevellachemla.eu/David-Slepian-Estelle-Sonnenblick-original.pdf>.
- [5] Françoise Richard-Jung, Jean-Pierre Ramis, Jean Thomann, and Frédéric Fauvet, *New Characterizations for the Eigenvalues of the Prolate Spheroidal Wave Equation*, Studies in applied mathematics, p. 1–40, 2016, Wiley Periodicals. <https://denisevellachemla.eu/Richard-Jung-Ramis-Thomann-Fauvet-2016.pdf>.
- [6] Denise. Vella-Chemla, *Décalages*, 2024, <https://denisevellachemla.eu/decalages.pdf>.
- [7] Li. Lian. Wang, *Analysis of spectral approximations using prolate spheroidal wave functions*, Mathematics of computation, Volume 79, Number 270, April 2010, p. 807–827.
- [8] Guilherme França, André LeClair, *Transcendental equations satisfied by the individual zeros of Riemann ζ , Dirichlet and modular L-functions* <https://arxiv.org/pdf/1502.06003>.

Annexe 1 : extrait du programme fournissant l'approximation d'Estelle Sonnenblick

```
import numpy as np
import math ; from math import log,pi,exp
import matplotlib.pyplot as plt

def bchapo(n,c):
    res = ((n*pi/2) - c +(pi/4))/((0.5772156649/2)+(2*log(2))+(0.5*log(c)))
    return(res)

def approximelambda(n,c):
    res = 1/(1+exp(pi*bchapo(n,c)))
    return(res)
```

```

def approximechi(n,c):
    # je ne sais pas ce que signifie n grand et c grand dans l'article de
    # Estelle Sonnenblick (je prends n > 20 ? et c > 50 ?)
    if n <= 30 and c <= 20:
        res = n*(n+1)+0.5*(1+(1/((2*n-1)*(2*n+3))))*c*c
    else:
        if n <= 30 and c > 20:
            res = (2*n+1)*c-((2*n*n+2*n+3)/4)-(((2*n+1)*(n*n+n+3))/(16*c))
        else:
            res = c*c+2*b*c+((b*b-1)/2)-((b*b-b)/(8*c))
    return(res)

for n in range(6):
    print('n = ',n)
    for c in range(1,6):
        print(approximechi(n,c),' ',end='')
    print('')

print('formule d approximation des lambda de l article d Estelle Sonnenblick ')
for x in range(1,20):
    lesy = []
    for y in range(1,20):
        print('lambda[',x,',',y,'] = ',l1lambda[x,y],' approxime par ',approximelambda(x,y))
        lesy.append(l1lambda[x,y])
    plt.plot(range(1,20),lesy)
plt.show()

print('formule d approximation des chi de l article d Estelle Sonnenblick ')
for x in range(1,20):
    for y in range(1,20):
        print('chi[',x,',',y,'] = ',chi[x,y],' approxime par ',approximechi(x,y))

print('la trentieme ligne')
for y in range(1,20):
    print('chi[30,',',y,'] = ',chi[30,y],' approxime par ',approximechi(30,y))

print('la trente-et-une-ieme colonne')
for y in range(0,20):
    print('chi[',y,',31] = ',chi[y,31],' approxime par ',approximechi(y,31))
for y in range(74,79):
    print('chi[',y,',31] = ',chi[y,31],' approxime par ',approximechi(y,31))

```

Le résultat de ce programme est stocké ici

<https://denisevellachemla.eu/res-approxime-Slepian-Sonnenblick.pdf>.

Annexe 2 : Calcul des 200 premiers zéros par résolution de l'équation différentielle de Sturm-Liouville

31^{èmes} valeurs propres de la 1^{ère} à la 50^{ème}

$vp[31](x)$ Sturm – Liouville	$2\sqrt{vp[31](x)}$	$\zeta(x)$
31.24389710557676	11.17924811524939	14.134725142
94.21896027377296	19.413290321197277	21.022039639
156.15522594015044	24.992416925151552	25.010857580
217.02479690312396	29.46352300069521	30.424876126
276.79712436619167	33.27444210598829	32.935061588
335.4385620950778	36.62996380533717	37.586178159
392.91180818935743	39.64400626522791	40.918719012
449.1751950482668	42.387507359988355	43.327073281
504.18176959991007	44.90798457289795	48.005150881
557.8780759336698	47.23888550479022	49.773832478
610.2025019024709	49.40455452293729	52.970321478
661.0829609105472	51.423067232927565	56.446247697
710.4335062292903	53.30791709415367	59.3470444003
758.1491043608085	55.0690150397048	60.831778525
804.0968939761894	56.71320459914744	65.112544048
848.0998392527224	58.244307507351216	67.079810529
889.9026608776078	59.66247265669125	69.546401711
929.1041044387441	60.96241807667226	72.067157674
965.1026164111245	62.1322015193772	75.704690699
997.4723147032058	63.1655702009633	77.144840069
1027.3206204218318	64.10368539863623	79.337375020
1057.7944588950675	65.04750445313233	82.910380854
1091.6948174573263	66.08161067823109	84.735492981
1129.675352045166	67.22128686793093	87.425274613
1171.2542652317939	68.44718446311123	88.809111208
1215.938430693059	69.74061745333373	92.491899271
1263.4224950736593	71.08930988759587	94.651344041
1313.5197100023847	72.4850249362552	95.870634228
1366.1058648582025	73.92173874735909	98.831194218
1421.092116378637	75.39475091486507	101.317851006
1478.4117719166898	76.9002411417985	103.725538040
1538.013113139856	78.43502057473705	105.446623052
1599.8550795064298	79.99637690561816	107.168611184
1663.90447608521	81.5819704612535	111.029535543
1730.1340732493504	83.18976074612429	111.874659177
1798.5212638337655	84.81795243540758	114.320220915
1869.0470859867453	86.46495442632802	116.226680321
1941.6954950439238	88.12934800720868	118.790782866

31^{èmes} valeurs propres de la 150^{ème} à la 200^{ème}

$vp[31](x)$ Sturm – Liouville	$2\sqrt{vp[31](x)}$	$\zeta(x)$
22863.472066686627	302.4134392958529	321.160134309
23163.452565570347	304.39088399996706	322.144558672
23465.43344945751	306.36862404272085	323.466969558
23769.414708278193	308.3466536758795	324.862866052
24075.396332289667	310.3249672990536	327.443901262
24383.378312063665	312.3035594549871	329.033071680
24693.360638474278	314.2824248250244	329.953239728
25005.343302686408	316.26155822474794	331.474467583
25319.32629614474	318.24095459977957	333.645378525
25635.309610563105	320.2206090217374	334.211354833
25953.293237914426	322.2005166843432	336.841850428
26273.277170421094	324.1806728996724	338.339992851
26595.261400545613	326.1610730945409	339.858216725
26919.245920981863	328.141712807024	341.042261111
27245.23072464659	330.12258768310045	342.054877510
27573.21580467125	332.1036934734165	344.661702940
27903.20115439433	334.08502603016694	346.347870566
28235.18676735384	336.0665813040853	347.272677584
28569.172637280182	338.0483553415409	349.316260871
28905.1587580894	340.0303442817385	350.408419349
29243.145123876453	342.01254435401313	351.878649025
29583.131728909146	343.9949518752224	353.488900489
29925.118567621936	345.97756324722525	356.017574977
30269.105634610183	347.960374954449	357.151302252
30615.092924624663	349.94338356153935	357.952685102
30963.080432566177	351.92658571108933	359.743754953
31313.06815348044	353.90997812144514	361.289361696
31665.056082553197	355.89355758458566	363.331330579
32019.044215105438	357.87732096407245	364.736024114
32375.032546588904	359.861265193068	366.212710288
32733.02107258179	361.8453872724194	367.993575482
33093.00978878441	363.82968426880404	368.968438096
33454.99869101528	365.8141533129372	370.050919212
33818.98777520718	367.79879159783644	373.061928372
34184.977037403485	369.78359637714317	373.864873911
34552.96647375451	371.7685649634972	375.825912767
34922.9560805142	373.7536947269643	376.324092231
35294.94585403642	375.7389830935109	378.436680250
35668.93579078962	377.724427543624	379.872975347
36044.925887283185	379.7100256105081	381.484468617
36422.916144412615	381.69577490149203	383.443529450