

## Prolate pour zéros de zêta, Denise Vella-Chemla, décembre 2025.

On revient au programme python pour trouver les zéros non triviaux de la fonction zêta de Riemann en utilisant les valeurs propres d'un opérateur prolate. On n'arrive pas à faire mieux que le programme ci-dessous, qui trouve à peine quelques zéros.

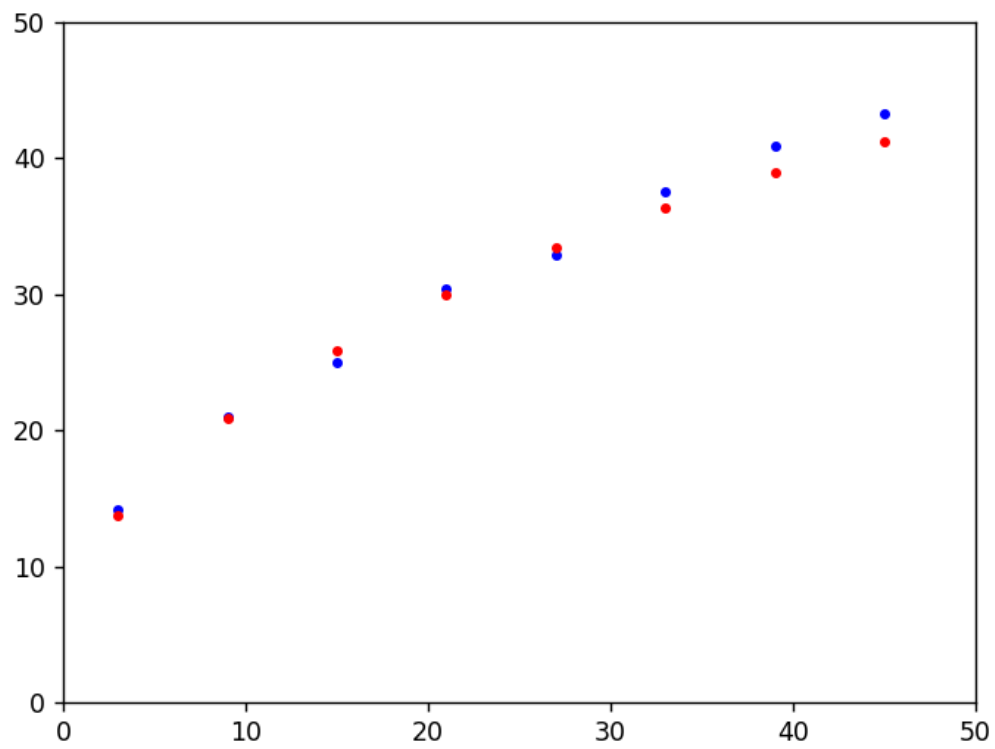
```
from scipy.special import pro_cv, pro_cv_seq, pro_ang1, pro_rad1, pro_rad2
import math
from math import pi
import matplotlib.pyplot as plt
import numpy as np

with open("zeros50", 'r') as f:
    lines = f.readlines()
    zeros = [float(line) for line in lines]
for k in range(1,10,2): # ne marche pas bien au dela de 14
    res = pro_cv(0,k,32)
    lezero = (2*res)**0.5
    indice = (k-1)//2
    print('indice = ',indice)
    print(f'{k} r = pro_cv(0,k,32) (python) {pro_cv(0,k,32):7.2f} sqrt(2r)
          {lezero:7.2f} zeta non trivial {zeros[indice]:7.2f}')
    plt.scatter(3*k,zeros[indice],color='blue',s=9) # on met 3*k pour mieux voir
    plt.scatter(3*k,lezero,color='red',s=9)
plt.xlim(0,50)
plt.ylim(0,50)
plt.show()
```

Les sorties numériques du programme sont :

```
indice = 0
1 r = pro_cv(0,k,32) (python) 94.22 sqrt(2r) 13.73 zeta non trivial 14.13
indice = 1
3 r = pro_cv(0,k,32) (python) 217.02 sqrt(2r) 20.83 zeta non trivial 21.02
indice = 2
5 r = pro_cv(0,k,32) (python) 335.44 sqrt(2r) 25.90 zeta non trivial 25.01
indice = 3
7 r = pro_cv(0,k,32) (python) 449.18 sqrt(2r) 29.97 zeta non trivial 30.42
indice = 4
9 r = pro_cv(0,k,32) (python) 557.88 sqrt(2r) 33.40 zeta non trivial 32.94
```

et la visualisation graphique, pour 8 zéros, est (où l'on commence à voir la divergence entre les deux ensembles de points) :



Si on tente d'augmenter le nombre de zéros, on obtient, par exemple, pour 100 zéros :

