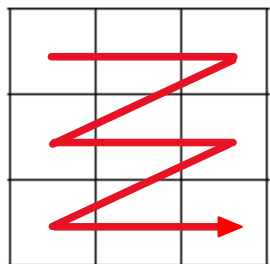


Quantum Information and Computing

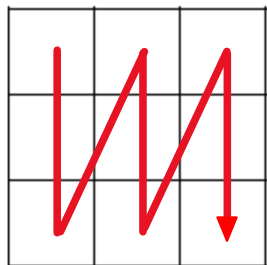
2021 - 2022

Saverio Monaco
21/11/2021
Exercise 3

Scaling of the matrix-matrix multiplication



Loop method



Loop2 method
(not cache
optimized)

Pseudocode for Loop function:

```
LOOP(A,B): # multiplies A and B
  C = matrix(A.ncols,B.nrows)
  for i from 1 to A.ncols:
    for j from 1 to B.nrows:
      sum = 0
      for k from 1 to A.nrows:
        sum = sum +  $A_{ik} \times B_{kj}$ 
      end for
    end for
  end for
```

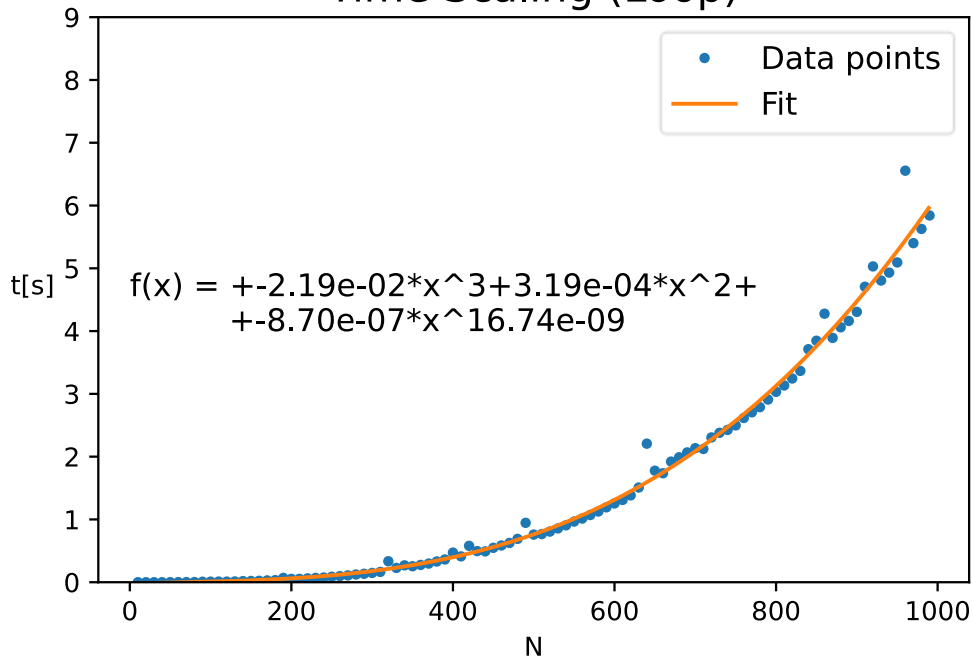
$$O(N^3)$$

Loop multiplication - matmul:

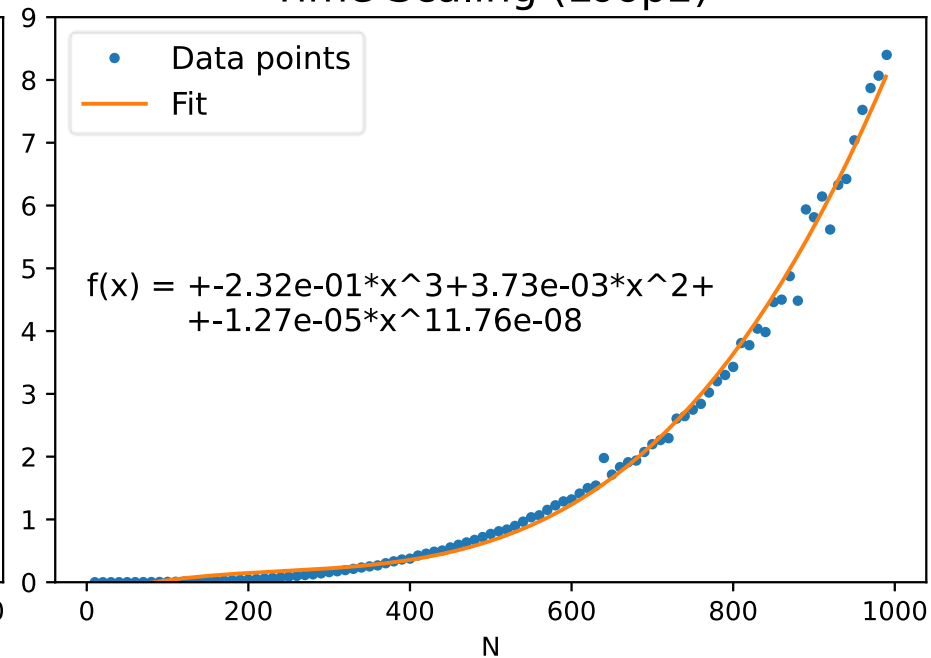
	0.00000000	-7.62939453E-06	0.00000000	
	0.00000000	0.00000000	0.00000000	
	3.81469727E-06	0.00000000	0.00000000	

Scaling of the matrix-matrix multiplication

Time Scaling (Loop)

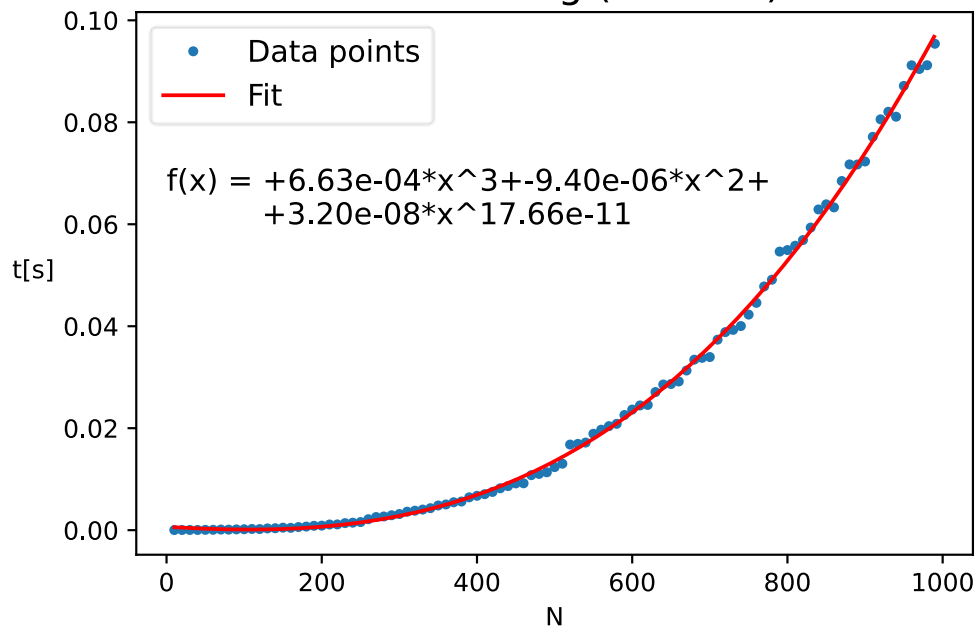


Time Scaling (Loop2)

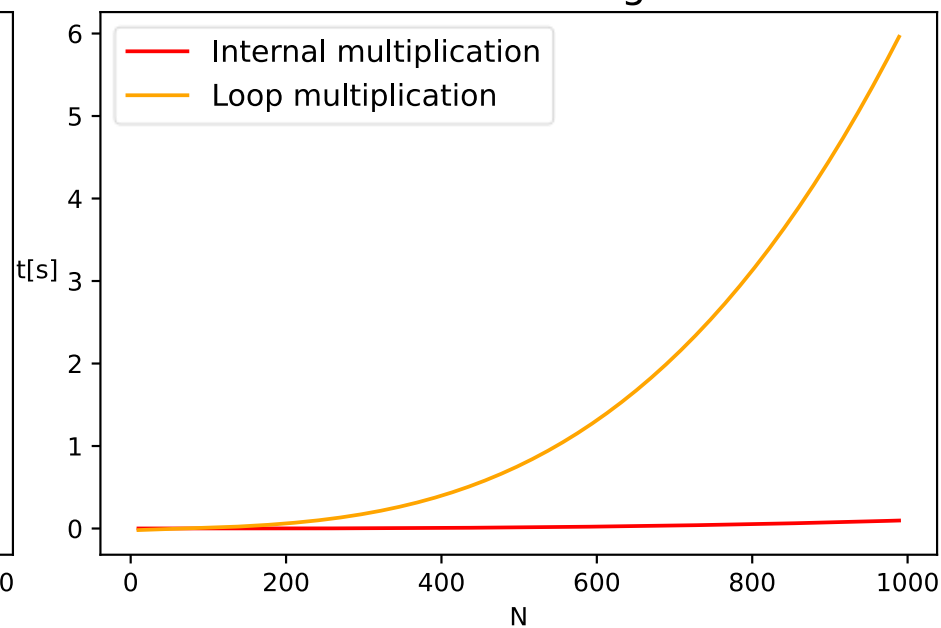


Scaling of the matrix-matrix multiplication

Time Scaling (Internal)



Time Scaling



Random Matrix Theory

Study of the $P(s)$ distribution, where s_i are the normalized spacings between eigenvalues:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{13} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \xrightarrow{\text{ZHEEV}() \text{ (Lapack)}} \lambda_1, \lambda_2, \lambda_3 \quad \lambda_1 < \lambda_2 < \lambda_3 \xrightarrow{\quad} s_i = \Delta\lambda_i / \bar{\Delta}\lambda$$

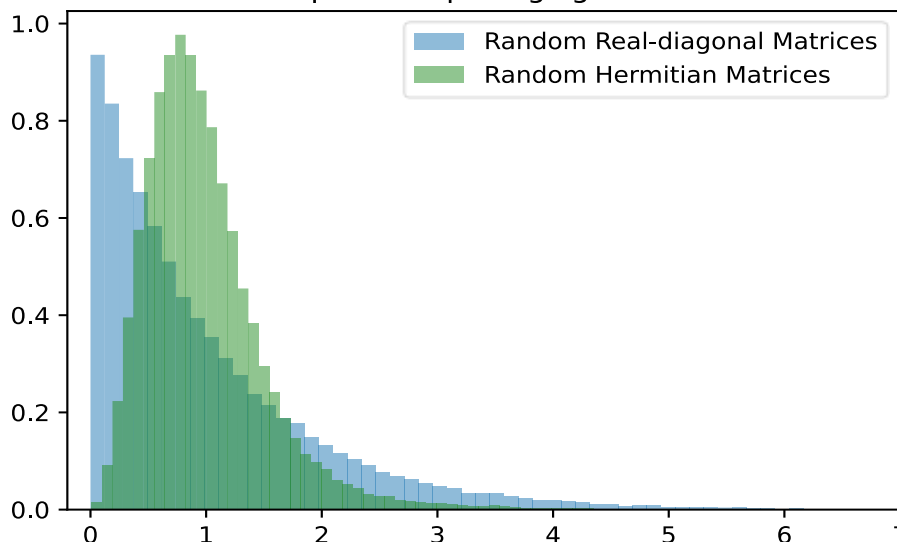
Fit $P(s)$ using `curve_fit` from `scipy.optimize` in Python with the function:

$$P(s) = a s^b \exp(-c s^d)$$

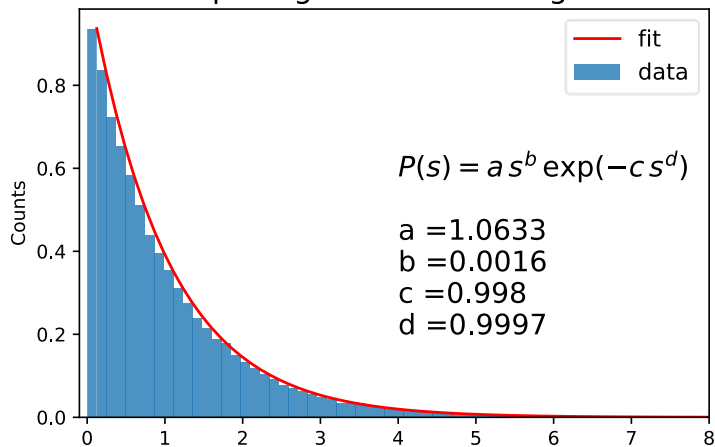
For random Hermitian matrices and random (real) diagonal matrices

Random Matrix Theory

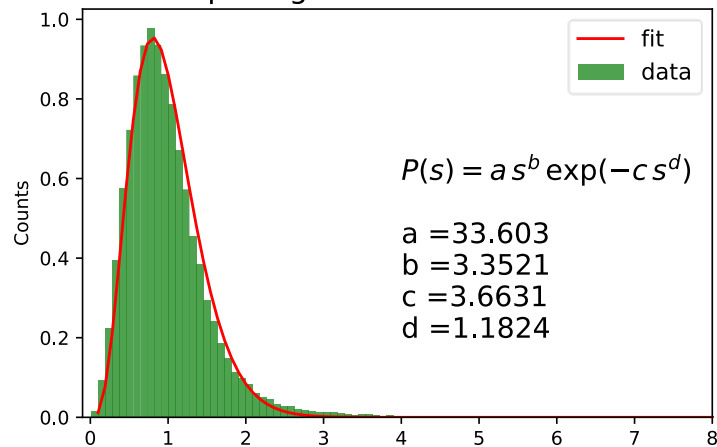
Comparison spacings generated



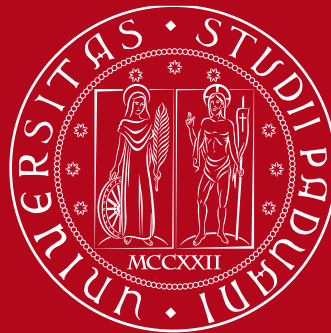
Normalized spacings of random diagonal matrices



Normalized spacings of random hermitian matrices



1222 • 2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Thanks for the attention
