Quantum Information and Computing

2022 - 2023

Nguyen Xuan Tung

21/12/2022

Exercise #03

Physics of Data

matrix-matrix multiplication

Theory

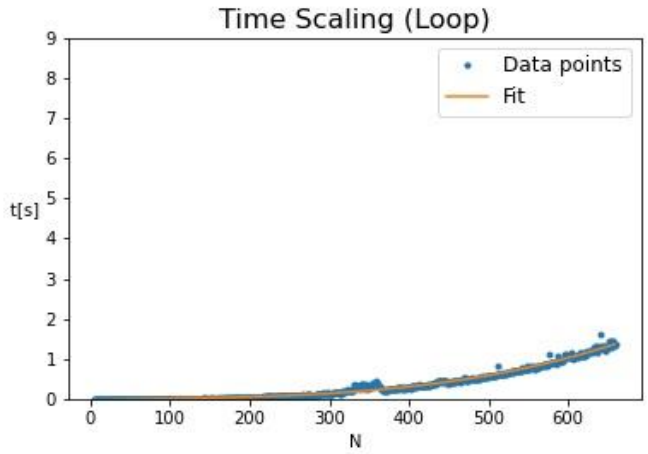
matrix multiplication is a binary operation that produces a matrix from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix.

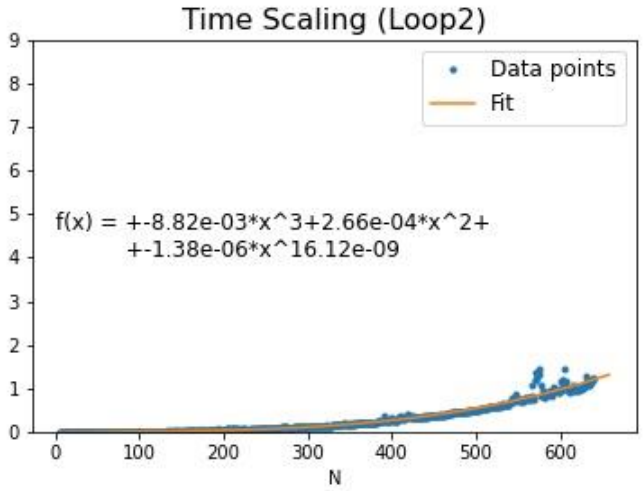
Code development

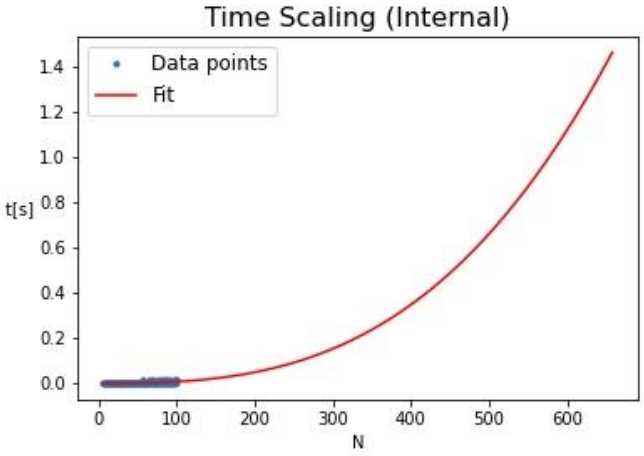
- We create a function perform matrix multiplication through a loop1, loop2 and matmul method.
- Store the matrix in csv file and run them in python code.

```
function matrix_multiplication_transposed (matrixA, matrixB) result(matr
    integer :: ii, jj, kk
    logical :: check
    real*4, dimension(:,:)   :: matrixA, matrixB
    real*4, dimension(size(matrixA,2),size(matrixB,1)) :: matrixC

! Check if multiplication is possible (shapes)
    if (size(matrixA,2) .eq. size(matrixB,1)) then
        check = .TRUE.
    else
        print*, "Input matrices cannot be multiplied"
        check = .FALSE.
    end if

! Matrix multiplication
    if (check .eqv. .TRUE.) then
        do kk = 1, size(matrixA,1), 1
            do jj = 1, size(matrixB,2), 1
                do ii = 1, size(matrixB,1), 1
                    if (kk == 1) then
                        matrixC(ii,jj) = 0
                    end if
                    matrixC(ii,jj) = matrixC(ii,jj) + matrixA(ii,kk)*mat
                enddo
            enddo
        enddo
    end if
end function matrix_multiplication_transposed
```

matrix-matrix multiplication







Result

- We can assume that the matrix multiplication operation scales line O(n^3).

Random matrix theory

Theory

Random Hermitian matrices, drawn from one-cut regular unitary invariant ensembles, converge in law to Gaussian multiplicative chaos measures. We prove this in the so-called L2L2-phase of multiplicative chaos

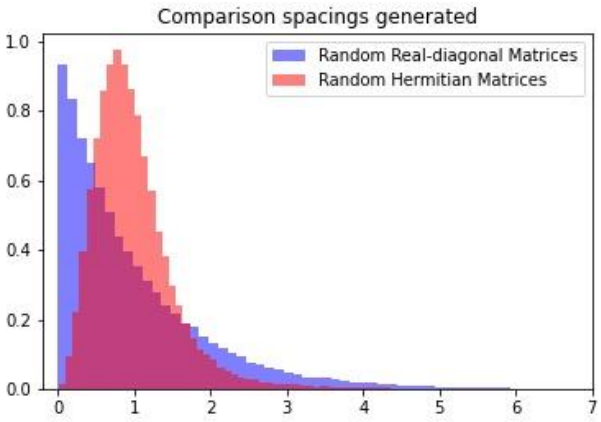
Code Development

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

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

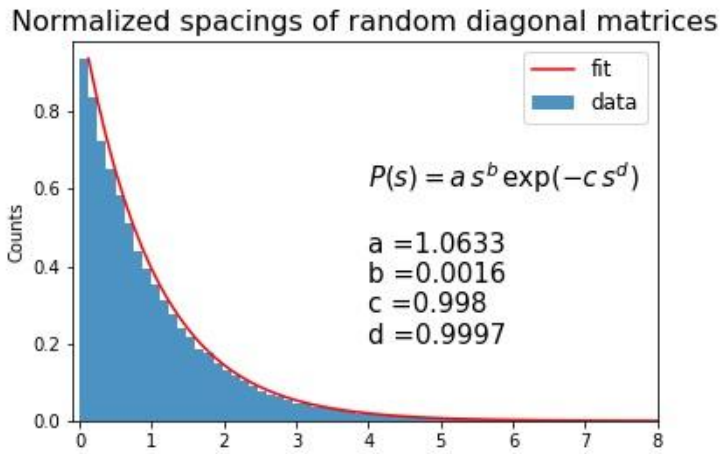
- We can exploit the ‘BLAS’ and ‘LAPACK’ -libraries which contains ‘ZHEEV’.
- In python, we will fit the function P(s) using `curve_fit` from `scipy.optimize`

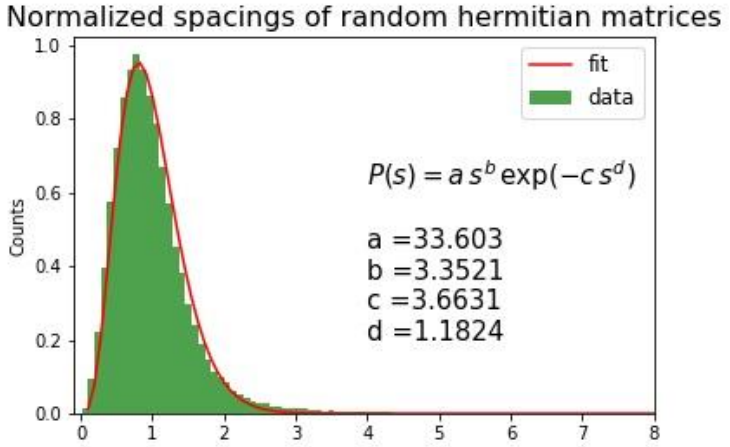
Random matrix theory

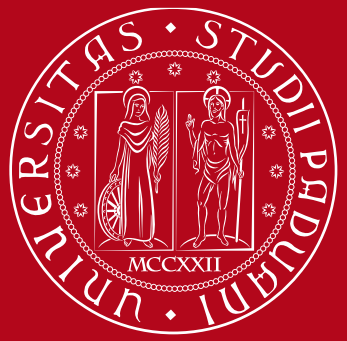


Result

- We perform the fit of normalize spacing of random diagonal and Hermitian matrices.







UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Thanks for the attention