# Random Matrices

**Abstract**

The aim of this assignment is to learn how to create random matrices, in particular Hermitian and diagonal with real entries , and diagonalize the Hermitian ones using `LAPACK`, a package written in `FORTRAN90`. Then, the normalized spacings between their eigenvalues, which are stored in crescent order, are calculated. The distributions of these results are plotted using `GNUPLOT`. At the end fits on the aforementioned graphs, using a proper function, verify if the distribution matches with the theory.

## Theory

A random matrix is a matrix in which the entries are random numbers. In our case the random numbers are drawn from a normal distribution and are generated through the `BOX-MULLER` algorithm. The latter allows to generate two independent random numbers, $y_1$ and $y_2$, sampled from the normal distribution $\mathcal{N}(0,1)$, by drawing two random numbers from a uniform distribution $\mathcal{U}([0,1])$, $u_1$ and $u_2$. According to the rule, $y_1$ and $y_2$ are defined in this way

$$y_1 = \sqrt{-2\ ln(u_1)}\ cos(2\pi u_2) \tag{1}$$

$$y_2 = \sqrt{-2\ ln(u_1)}\ sin(2\pi u_2) \tag{2}$$

The eigenvalues, both for the Hermitian and the real diagonal matrix are found and sort in increasing order. At this point, the exercise requires to calculate the spacing between them,

$$\Delta\lambda_i = \lambda_{i+1} - \lambda_i \tag{3}$$

and normalize it with their average $< \Delta\lambda >$. Then one can fit the normalized spacings $s_i = \frac{\Delta\lambda_i}{<\Delta\lambda>}$ distribution with the function

$$P(x) = ax^\alpha exp(-bx^\beta) \tag{4}$$

For the Hermitian matrices, the expected values of the parameters are

| a | $\alpha$ | b | $\beta$ |
|---|---|---|---|
| $32/\pi^2$ | 2 | $4/\pi$ | 2 |

On the other hand, for diagonal matrices with real entries, no theoretical values are avaible.

# Code development

## Hermitian matrices

Firstly, in the code called `Ex5-Quaglia-CODE1.f90`, a `MODULE`, named `H_MATRIX`, is written. As the Listing 1 shows, the `MODULE` contains different `SUBROUTINE`s:

- The `'normal_rand'` `SUBROUTINE`, which generates two gaussian random numbers through the `BOX MULLER` algorithm, as described in the **Theory** section.
- The `'DIMENSION'` `SUBROUTINE` that asks the user to decide the size $N$ of the square matrix. If the user writes a negative value, a warning message is printed, and through a `DO WHILE` loop, the request is repeated until a positive value is inserted.

```fortran
MODULE H_MATRIX

  IMPLICIT NONE
  INTEGER*8 :: N  ! dimension of the matrix (square matrix)

CONTAINS
  SUBROUTINE normal_rand(x, y)
    ! this subroutine genetares two random numbers, x & y
    ! sampled from the normal distribution, through the BOX-MULLER algorithm
    REAL*8 :: u1, u2, r, theta
    REAL*8, INTENT(OUT) :: x, y

    CALL RANDOM_NUMBER(u1) ! u1 and u2 are random numbers drawn from ~ U([0,1])
    CALL RANDOM_NUMBER(u2)

    r = SQRT(2*(-LOG(u1)))
    ! pi = 2*arcsin(1)
    theta = 2*(2*ASIN(1.)) * u2

    x = r*COS(theta)
    y = r*SIN(theta)

  END SUBROUTINE normal_rand

  SUBROUTINE DIMENSION(N)
    ! This subroutine asks the user the dimension of the matrix
    INTEGER*8 :: N

    N=0      ! all the dimensions are 'inizialized' to zero

    DO WHILE (N <= 0)
       PRINT*, "The dimension of the matrix is:"
       READ*, N
       IF (N <= 0) PRINT*, "Try with a positive value"
    ENDDO
  END SUBROUTINE DIMENSION
```

```
END MODULE H_MATRIX
```

Listing 1: The MODULE 'H_MATRIX'

In the main program, named Ex5, after the definition of all the structures needed, the matrix $A$, filled with the normal random numbers, is asked to be Hermitian (2).

```
! Making the matrix hermitian
  DO ii=1,N
     DO jj=1,ii
        CALL normal_rand(re,im)

        IF (ii.EQ.jj) THEN
           A(ii,jj) = CMPLX(re, 0e0)
        ELSE
           A(ii,jj) = CMPLX(re, im)
           A(jj,ii) = CMPLX(re, -im)
        END IF
     END DO
  END DO
```

Listing 2: Hermitian matrix

The eigenvalues of $A$ are calculated using LAPACK'S 'ZHEEV' SUBROUTINE. The informations on the latter can be found at the following link [1]
The normalized spacings between eigenvalues, $s_i$, are calculated as described in the **Theory** section, and printed in the file called 's_i.dat'. This file is then used by the GNUPLOT's script, hist.gnu, to make a normalized histogram of the spacings $s_i$ and fit it with the equation (4). A check is performed on the variable 'info', and if the eigenvalues are in ascending order as we want, a message is printed on screen.

```
  IF (info == 0) THEN
     print*,"Eigenvalues are stored in ascending order" ! check if
    info == 0
  END IF
```

Listing 3: Check 'info'

## Diagonal matrices with real entries

In the second code, named Ex5-Quaglia-CODE2.f90, the MODULE 'H_MATRIX' of the first code is replaced by 'H_MATRIX2', which contains an adding SUBROUTINE, shown in the Listing 4, that is used to sort the eigenvalues in crescent order. In fact, in this case, the eigenvalues of the matrix are simply the elements of the diagonal, so LAPACK is not used.

```
  SUBROUTINE SORT(array)
 ! This subroutine orders an array given in input in crescent order
    REAL*8, DIMENSION(:), INTENT(INOUT) :: array
    INTEGER*8 :: i1, i2
```

[1] http://www.netlib.org/lapack/explore-html/df/d9a/group__complex16_h_eeigen_gaf23fb5b3ae38072ef4890ba43d5cfea2.html

```
    REAL*8 :: temp_num
    DO i2=1,SIZE(array)
       DO i1=1,(SIZE(array)-1)
          IF (array(i1).GE.array(i1+1)) THEN
             temp_num = array(i1)
             array(i1) = array(i1+1)
             array(i1+1) = temp_num
          END IF
       END DO
    END DO
    RETURN
  END SUBROUTINE SORT
```

Listing 4: `'SORT' SUBROUTINE`

The normalized spacings $s\_i$ are written in a file called `'s_iREAL.dat'`, used by the script `hist2.gnu`, similar to the one before.

# Results

## N = 1000

In this subsection the size $N$ of the square matrix is fixed to 1000. Three runs are made. The plot performed through the script `hist.gnu` is showed in figure 1. The parameters of the fitting function, reported in the `.log` file are in the following table. The errors are given by `GNUPLOT`.

| a | $\sigma(a)$ | $\alpha$ | $\sigma(\alpha)$ | b | $\sigma$(b) | $\beta$ | $\sigma(\beta)$ |
|---|---|---|---|---|---|---|---|
| 2.8 | 1.9 | 2.8 | 0.4 | 3.2 | 0.7 | 1.22 | 0.15 |

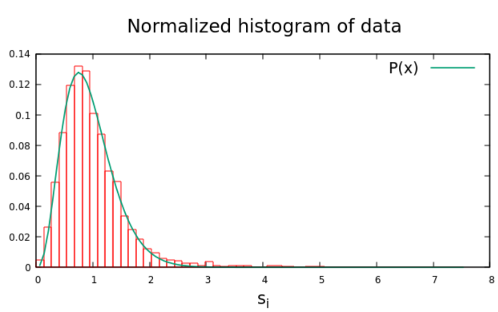


Figure 1: Normalized histogram of the normalized spacings between eigenvalues of the Hermitian matrix. The fitting function P(x) is of the type (4).

A similar plot, this time regarding the diagonal matrix, performed through the script `hist2.gnu`, is showed in figure 2. The parameters of the fitting function are in the following table. The errors are given by `GNUPLOT`.

| a | $\sigma(a)$ | $\alpha$ | $\sigma(\alpha)$ | b | $\sigma$(b) | $\beta$ | $\sigma(\beta)$ |
|---|---|---|---|---|---|---|---|
| 7.26372 | 3.88e+06 | -3.422 | 5560 | 0.35523 | 5.342e+05 | 0.0219 | 1.741e+04 |

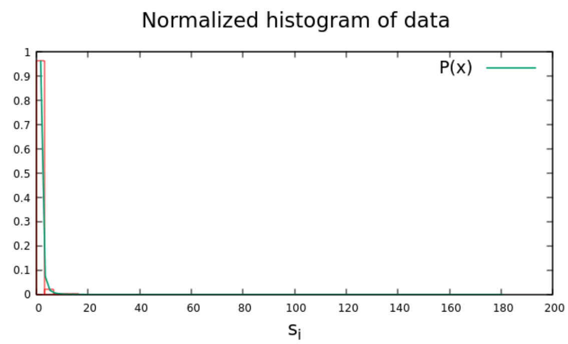


Figure 2: Normalized histogram of the normalized spacings between eigenvalues of the diagonal real matrix. The fitting function P(x) is of the type (4).

## N = 3000

Now the size $N$ of the square matrix is fixed to 3000. Three runs are performed. The fitted histogram of the normalized spacings is showed in figure 3. The errors on the fitting parameters are again given by `GNUPLOT`.

| a | $\sigma(a)$ | $\alpha$ | $\sigma(\alpha)$ | b | $\sigma$(b) | $\beta$ | $\sigma(\beta)$ |
|---|---|---|---|---|---|---|---|
| 1.2 | 0.7 | 2.75 | 0.20 | 3.0 | 0.4 | 1.27 | 0.09 |

A similar plot than before, regarding the diagonal matrices, is showed in figure 4.

| a | $\sigma(a)$ | $\alpha$ | $\sigma(\alpha)$ | b | $\sigma$(b) | $\beta$ | $\sigma(\beta)$ |
|---|---|---|---|---|---|---|---|
| 34 | 453 | -0.36 | 2.6 | 4 | 14 | 0.3 | 0.5 |

## Comments

The values of the parameters of the fitting function P(x), in the case of the Hermitian matrix, are reasonable, considering the theoretical ones, for both the sizes. However the values can improve, maybe varying the spacings (optional point), as the next section states. Moreover the parameters of the fit for the diagonal case have not theoretical values to compare with. However the errors given by `GNUPLOT` explodes, so one can think of using a different procedure.

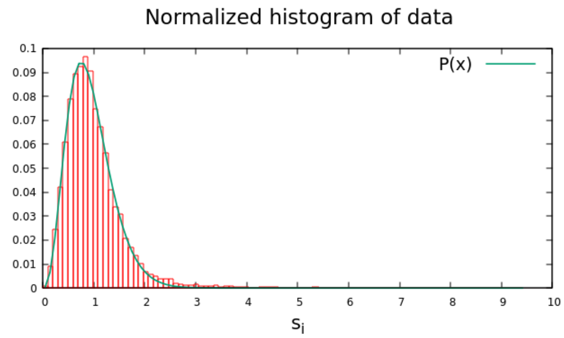


Figure 3: Normalized histogram of the normalized spacings between eigenvalues of the Hermitian matrix. The fitting function P(x) is of the type (4).

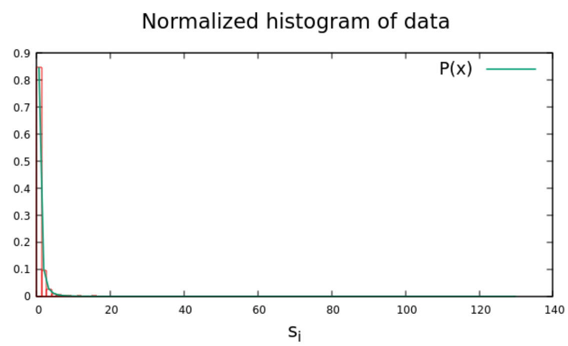


Figure 4: Normalized histogram of the normalized spacings between eigenvalues of the diagonal real matrix. The fitting function P(x) is of the type (4).

# Self-Evaluation

This assignment was the first occasion to me to practise with `LAPACK`, that is a very powerful tool and even (once get!) user friendly to implement. Improvements can be make, in terms of automation of the codes and in increasing the number of runs. Also the optional points could be implement in future, now they are not present due to lack of time.