Assignment 2

Quantum Information and Computing Course 2022/2023

Massimo Colombo

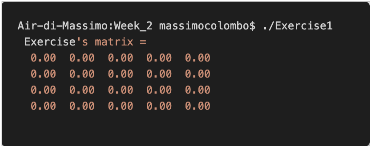
Department of Physics and Astronomy "Galileo Galilei"
Degree in Physics

November 7, 2022

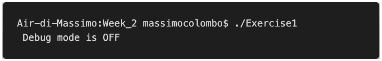
Exercise 1

Debugging subroutine: **print_matrix_debug**("matrix info, ... ", *debug*).
It contains an IF statement that depends on the logical variable *debug*.

- logical :: debug = .true.
- logical :: debug = .false.



```
Air-di-Massimo:Week_2 massimocolombo$ ./Exercise1
 Exercise's matrix =
  0.00  0.00  0.00  0.00  0.00
  0.00  0.00  0.00  0.00  0.00
  0.00  0.00  0.00  0.00  0.00
  0.00  0.00  0.00  0.00  0.00
```



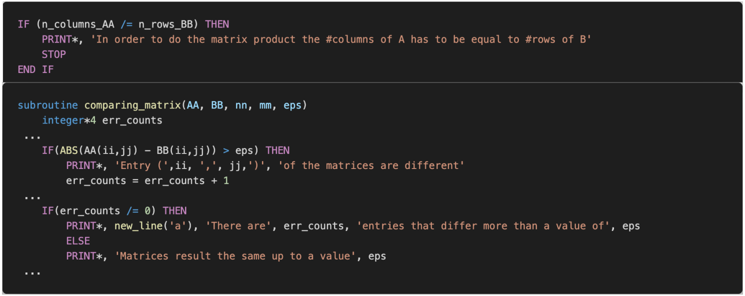
```
Air-di-Massimo:Week_2 massimocolombo$ ./Exercise1
 Debug mode is OFF
```

Fig. 1 STDOUT: Exercise1.f90 vs Debug variable.

Exercise 2, Part. 1

The second request is to implement a previous assignment's exercise in 5 points.

(1) DOCUMENTATION can be read in the Read.me file.
(2) COMMENTS can be read in the Exercise2.f90 code.
(3) PRE- and POST- CONDITIONS:



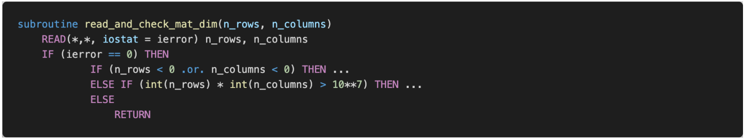
```
IF (n_columns_AA /= n_rows_BB) THEN
    PRINT*, 'In order to do the matrix product the #columns of A has to be equal to #rows of B'
    STOP
END IF
```

```
subroutine comparing_matrix(AA, BB, nn, mm, eps)
    integer*4 err_counts
 ...
    IF(ABS(AA(ii,jj) - BB(ii,jj)) > eps) THEN
        PRINT*, 'Entry (',ii, ',', jj,')', 'of the matrices are different'
        err_counts = err_counts + 1
 ...
    IF(err_counts /= 0) THEN
        PRINT*, new_line('a'), 'There are', err_counts, 'entries that differ more than a value of', eps
        ELSE
        PRINT*, 'Matrices result the same up to a value', eps
 ...
```

Fig.2 Pre- and Post- Conditions

Exercise 2, Part. 2

(4) ERROR HANDLING: Possible errors may arise from the incorrect matrix-dimension **STDIN**. To prevent this, the IOSTAT command was exploited together with IF statements.



```
subroutine read_and_check_mat_dim(n_rows, n_columns)
    READ(*,*, iostat = ierror) n_rows, n_columns
    IF (ierror == 0) THEN
            IF (n_rows < 0 .or. n_columns < 0) THEN ...
            ELSE IF (int(n_rows) * int(n_columns) > 10**7) THEN ...
            ELSE
                RETURN
```

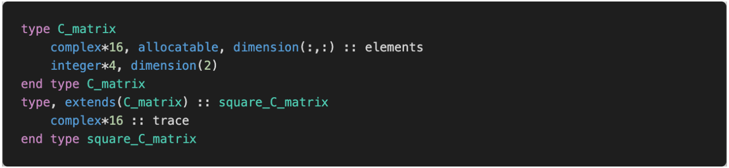
Fig.3 Error Handling.

- *IOSTAT* $= 0$, means the READ was executed flawlessly and all variables have received their input values.
- *IOSTAT* > 0, the previous READ has encountered some problem. A common problem is illegal data.
- *IOSTAT* < 0, it means the end of the input has reached. Under this circumstance, some or all of the variables in the READ may not receive input values.

(5) CHECKPOINTS: Logical variable *Debug* as in Ex.1 together with **PRINT**(debug) subroutines.

Exercise 3, Part. 1

The goal is to construct a Derived Type related to Complex Matrices.
It is possible to read in "Exercise3.f90" all the subroutines and functions that are associated with the Derived Types in Fig. 3.



```
type C_matrix
    complex*16, allocatable, dimension(:,:) :: elements
    integer*4, dimension(2)
end type C_matrix
type, extends(C_matrix) :: square_C_matrix
    complex*16 :: trace
end type square_C_matrix
```

Fig.4 Derived Types.

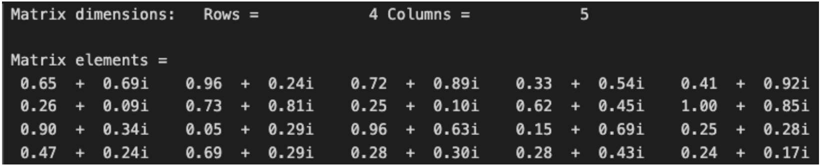
The choice of defining an extended type from the derived type C_matrix has been made to start understanding how the OOP feature works in Fortran.
It has been possible to explore the definition of subroutines and functions that act through **INHERITANCE** or distinctly on the type and its extended one.

Exercise 3, Part. 2

To conclude, a subroutine that prints in a readable form a complex matrix into a file, has been added to the test program:

print_C_matrix_intofile(mat, name, unit):

- "Mat, name, unit" corresponds to the matrix, its name, and the unit associated with the output file, requested by the **OPEN** and **WRITE** commands.
- The subroutine splits the real and imaginary part of each entry, adds the imaginary unit *i*, and adds the coherent sign between them".



```
Matrix dimensions:   Rows =          4 Columns =          5

Matrix elements =
 0.65  +  0.69i   0.96  +  0.24i   0.72  +  0.89i   0.33  +  0.54i   0.41  +  0.92i
 0.26  +  0.09i   0.73  +  0.81i   0.25  +  0.10i   0.62  +  0.45i   1.00  +  0.85i
 0.90  +  0.34i   0.05  +  0.29i   0.96  +  0.63i   0.15  +  0.69i   0.25  +  0.28i
 0.47  +  0.24i   0.69  +  0.29i   0.28  +  0.30i   0.28  +  0.43i   0.24  +  0.17i
```

Fig.5 Result.