

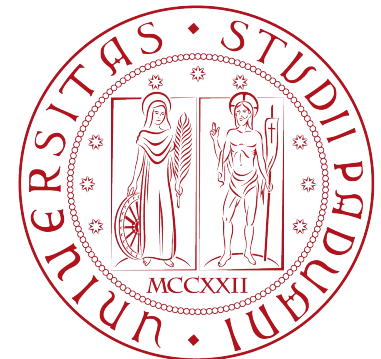
Presentation of Assignment 2

Quantum Information and Computing, A.Y. 2022/2023

Due date: 08/11/2022

Paolo Zinesi

paolo.zinesi@studenti.unipd.it



“EX2a_Zinesi_CODE.f90” tests the module “**checkpoint_mod**”, which contains different copies of the same checkpoint subroutine. Each copy is able to print a specific variable type and an optional string. All the copies are packed into a generic subroutine “**checkpoint**” using an interface.

```
! This interface defines the generic checkpoint subroutine.  
! The structure is similar for all the specific subroutines,  
! the only changes are in the type of numeric variable ("val") to print  
!  
! inputs:  
! - debug [logical]: Boolean value to enable/disable debugging  
! - str [character(len=*), optional]: optional string to print  
! - val [different types, optional]: optional numerical value to print
```

```
INTERFACE checkpoint  
  MODULE PROCEDURE STRcheckpoint  
  MODULE PROCEDURE Dcheckpoint, Acheckpoint  
  MODULE PROCEDURE Icheckpoint, Jcheckpoint, Kcheckpoint  
  MODULE PROCEDURE Ccheckpoint, CDcheckpoint  
END INTERFACE checkpoint
```

“EX2b_Zinesi_CODE.f90” tests the module “MatMul_mod”, which is a rewriting of “EX1c_Zinesi_CODE.f90” in a more elegant form.

Improvements with respect to the previous version:

- Elimination of subroutines' interfaces (FORTRAN generates them autonomously when importing the MatMul module).
- Addition of (switchable) checkpoints.
- Allocation of matrices at runtime using command-line arguments.

Documentation, comments, and compatibility of dimensions were already present in the previous version. Other checks on the command line arguments are included in this new version.

“EX2c_Zinesi_CODE.f90” tests the module “DCmatrix_mod”, which contains a double complex matrix derived type with functions and subroutines to manage it.

```
! declaration of derived type
TYPE DCmatrix
  INTEGER, DIMENSION(2) :: dim ! matrix dimensions
  DOUBLE COMPLEX, DIMENSION(:,,:), ALLOCATABLE :: elem ! matrix elements
END TYPE DCmatrix
```

Compilation with “-pedantic” option raises a warning, because “DOUBLE COMPLEX” is an extension of gfortran compiler. The compilations are performed with “-Wall -Wextra” options.

DCmatrix_mod.f90:23:27:

```
23 |           DOUBLE COMPLEX, DIMENSION(:,,:), ALLOCATABLE :: elem
    |                               1
```

Warning: GNU Extension: DOUBLE COMPLEX at (1)

- Initializer function **DCmatrix**: returns an allocated matrix with number of rows and columns specified in input, filled with an optional value. Positiveness of dimensions is checked.
- Ajoint operator **.ADJ.:** returns a DCmatrix that is the adjoint of the DCmatrix given as input.
- Trace **.TR.:** returns the trace of the input DCmatrix, if it is a square matrix, otherwise the program is halted.
- **writeMatFile**: write into a file a DCmatrix row by row. The format in which the complex numbers have to be written is specified in input.

```
! initializer interface
INTERFACE DCmatrix
|     MODULE PROCEDURE DCmatrix_init
END INTERFACE DCmatrix
```

DCmatrix - Testing



Successful allocation and computation of traces (debug= .FALSE.):

```
paolozinesi@MBP-di-Paolo Assignment2/EX2c » ./a.out 2 2
Trace input matrix = 1.00000E-01 +5.00000E-03 i
Trace adjoint matrix = 1.00000E-01 -5.00000E-03 i
paolozinesi@MBP-di-Paolo Assignment2/EX2c » ./a.out 200 200
Trace input matrix = 5.37340E+04 +2.68670E+03 i
Trace adjoint matrix = 5.37340E+04 -2.68670E+03 i
```

Error handling:

```
paolozinesi@MBP-di-Paolo Assignment2/EX2c » ./a.out 200 150
'MatTrace': Trying to compute trace of a non-square matrix
paolozinesi@MBP-di-Paolo Assignment2/EX2c » ./a.out 200 -200
Non-positive number given as matrix dimension
```

Possible improvements to the present module:

- Automatic deallocation of matrices.
- Redirection of error messages to a dedicated stream.