MASTER'S DEGREE IN PHYSICS
Academic Year 2020-2021

QUANTUM INFORMATION

Student: Giorgio Palermo
Student ID: 1238258
Date: October 20, 2020

EXERCISE 2

In this report I will review my solution to EX2, which is about the definition of new types, functions, subroutines and interfaces.

Theory

I based my solution of the proposed exercise on the definition of the `type`, `function`, `subroutine` and `interface` constructs.

Code Development

The basic brick of this program is the `dmatrix` type, which I defined as a new type containing a `double complex` matrix and some of its properties: shape, track and determinant.

The `InitUni` function is a `type(dmatrix)` function that calls the `clarnv` LAPACK subroutine to fill the matrix (`dmatrix%elem`) with random complex numbers. Since `clarnv` only works on scalar or vectors, I implemented a cycle to fill the matrix; I chose to loop over columns because this is the fastest algorithm since the matrix is stored column-wise. I decided that in my program the shape of a `dmatrix` has to be defined separately before the call to the initialization function, therefore I put a check at the beginning of it to verify that both dimensions are defined and positive.

The `Tr` function computes the trace summing over diagonal elements of a `dmatrix%elem` matrix given as input.

`Adj` is a `type(dmatrix)` function which aim is to compute the transposed conjugate of a `type(dmatrix)` input. To do this it copies an input `dmatrix` type element into a local new variable and computes the adjoint using the intrinsic elemental function `conjg()`; the transposition is then performed using the intrinsic `transpose()` function.

I assigned the `Adj` and the `InitUni` functions to two interface operators: `.Adj.` and `.Init.`.

All this is tested in a simple program, `DMatrixCODE`, which calls all the above mentioned functions and operators. More specifically, it defines and initializes a new `dmatrix` type, computes its adjoint, and writes both matrices on two separate text files, using the `MatToFile` subroutine.

Results

SCREENSHOTS OF THE OUTPUT FILES

Self evaluation

Writing this exercise I learned how to define new types, functions, subroutines and interface operators; I also learned to call external LAPACK functions and to compile the code including the linear algebra library.

I wonder if in REF TO TRACE LISTING function is sufficient to check for the dimensions of the matrix to be positive or it would be recommendable to check if the memory for the `dmatrix%elem` is already allocated, in order to avoid errors.