Assignment 2

https://github.com/jchryssanthacopoulos/quantum_information/tree/main/assignment_2

Quantum Information and Computing
AA 2022–23

James Chryssanthacopoulos

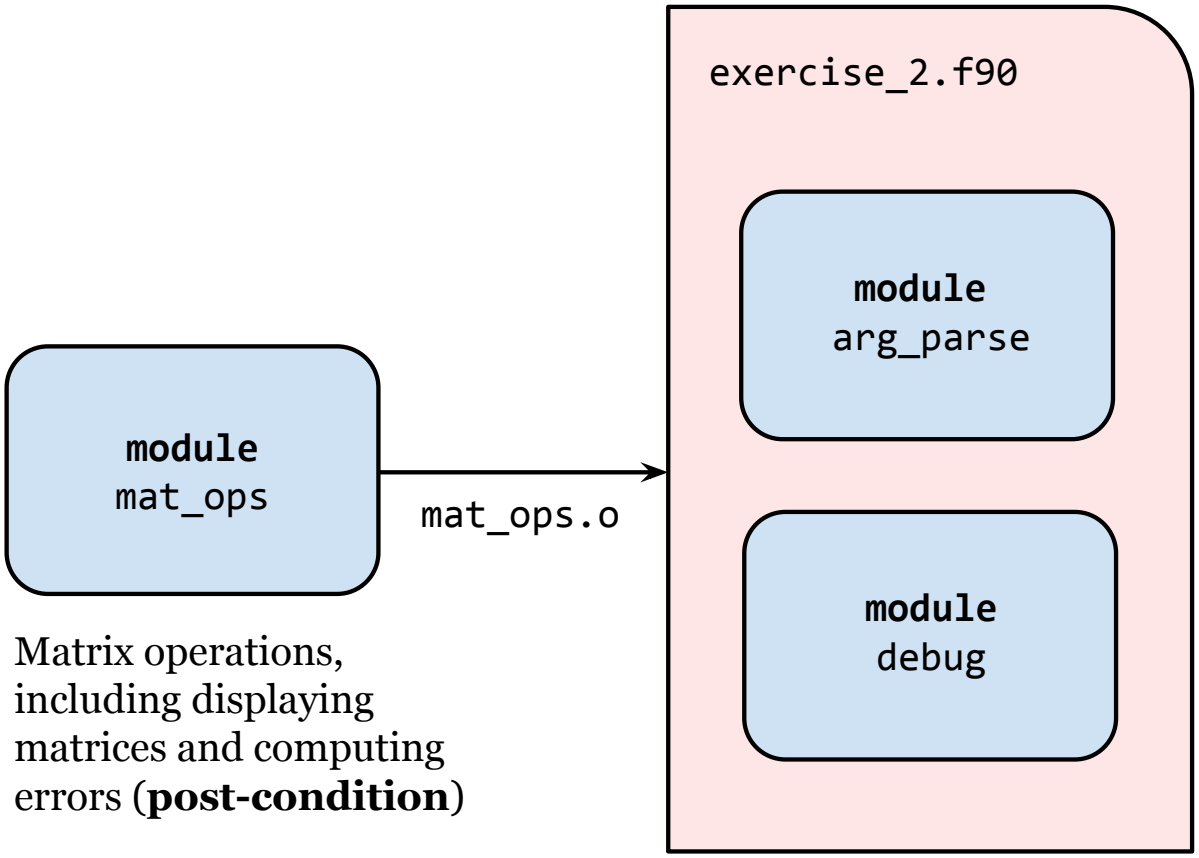
8 November 2022

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Exercise 1: Checkpoints

Exercise 2: Documentation (1)

Matrix multiplication program enhanced to be fault tolerant and easier to debug



Validates user-provided matrix dimensions are positive integers (**pre-condition**)

Debug checkpoints to display input, output matrices based on command-line flag

Matrix operations, including displaying matrices and computing errors (**post-condition**)

Exercise 2: Documentation (2)

Less room for error and more visibility into the results

```
 Running in verbose mode ...
 Matrix A =
   0.82   0.79
   0.80   0.99
 Matrix B =
   0.25   0.71
   0.48   0.96
 Product using matmul =
   0.58   1.35
   0.67   1.53
Elapsed time for matmul = 1.7000000000E-05
 Matrix using row-col-inner =
   0.58   1.35
   0.67   1.53
Max abs error for row-col-inner = 0.0000000000E+00
Elapsed time for row-col-inner = 3.0000000000E-06
 Matrix using inner-col-row =
   0.58   1.35
   0.67   1.53
Max abs error for inner-col-row = 0.0000000000E+00
Elapsed time for inner-col-row = 2.0000000000E-06
```

Verbose mode

```
# non-integers
$ compiled/exercise_2
 Enter number of rows, columns, and inner dimension:
a b c
 Dimensions need to be integers!

# non-positive integers
$ compiled/exercise_2
 Enter number of rows, columns, and inner dimension:
1, 2, -1
 Dimensions must be greater than zero!
```

Error checking

Exercise 3: Derived types (1)


```
module cmatrix

  type cmatrix
    N
    elems
    trace
  end type

  interface Trace
  interface operator (.Adj.)

contains
  function Init(nrows, ncols)
  subroutine Trace(M)
  function Adjoint(M)
  subroutine Del(M)

  + other utilities
```

cmatrix.o