Quantum information and computing:
Report 8

Nicola Lonigro
University of Padua
(Dated: November 30, 2020)

The efficiency in the initialization of a state vector describing a separable state and a generic state are compared. The density matrix corresponding to a generic state vector of a 2 body system is computed and the reduced density matrices corresponding to the 2 single body systems are obtained by tracing over the full density matrix

I. THEORY

A density matrix is a matrix that describes the statistical state, whether pure or mixed, of a system in quantum mechanics. The probability for any outcome of any well-defined measurement upon a system can be calculated from the density matrix for that system. Given a state vector $|\psi\rangle$, its corresponding density matrix is given by

$$\rho = |\psi\rangle\langle\psi|$$

In the case of a statistical ensemble the density matrix will be given by

$$\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|$$

In the case of a separable state of N bodies in D dimensions, that is a state that can be written as a tensor product and so without entanglement, it is only required to store the D factors for each body and so only D*N coefficients are required. For a generic N body state the entire D^N coefficients are required to store the information regarding the entanglement of the state. In the case of a density matrix representing a many-body system it is possible to obtain the reduced density matrix referring to only a subsystem using

$$\rho_A = \sum_\beta \langle\beta|\rho_{AB}|\beta\rangle$$

where ρ_{AB} is the full density matrix of the system, $|\beta\rangle$ is the basis for sub-system B and ρ_A is the reduced density matrix referring to sub-system A.

II. CODE DEVELOPMENT

The code uses the same structure of previous exercises. In this case there are three possible optional flags: DEBUG,TIME and GENERIC.

```
CALL GET_COMMAND_ARGUMENT(1,input_file)
open (unit = 5, file = input_file)
CALL FATAL(  LEN_TRIM(input_file) == 0,      "main","Missing input file name" )

CALL GET_COMMAND_ARGUMENT(2,output_rad)
CALL FATAL(  LEN_TRIM(output_rad) == 0,      "main","Missing output file name" )
open (unit = 10,Access = 'append', file = TRIM(output_rad)//".txt")

CALL GET_COMMAND_ARGUMENT(3,call_flags)
IF (call_flags == "DEBUG")     D_FLAG = .TRUE.
IF (call_flags == "TIME")     Time_flag = .TRUE.
IF (call_flags == "GENERIC")     Generic_flag = .TRUE.
CALL GET_COMMAND_ARGUMENT(4,call_flags)
IF (call_flags == "DEBUG")     D_FLAG = .TRUE.
IF (call_flags == "TIME")     Time_flag = .TRUE.
IF (call_flags == "GENERIC")     Generic_flag = .TRUE.


IF (Time_flag .EQV. .FALSE.)  Generic_flag = .TRUE.
read(5,'(I10)',iostat=ierror) D,N
CALL FATAL( ierror > 0 ,      "main","Inputs are not integers" )

CALL FATAL(  D <= 0 ,      "main","Dimension must be non-negative" )
CALL FATAL(  N <= 0 ,      "main","Number of elements must be non-negative" )
```

If the DEBUG flag is used the code will print on screen the different relevant quantities during execution. If the TIME flag is used, the time required to allocate and initialize the separable state vector of a system with dimension D and N bodies is computed. If both the TIME and GENERIC flag are used the time required for generic state vectors is computed. The reason for this difference is that the time scales for the two cases are completely different and so it is necessary to use vectors of very different dimensions in the two cases to have similar initialization times. If the TIME flag is activated the program will not allocate and compute the density matrices

```
IF (Time_flag)call cpu_time(start)
DO i = 1,D*N
    wavefunction(i) =  COMPLEX(RAND(),RAND())
END DO
norm_wave = 0
DO i = 1,D*N
    norm_wave = norm_wave + wavefunction(i) * CONJG(wavefunction(i))
END DO
```

```
  wavefunction = wavefunction/SQRT(norm_wave)

  IF (Time_flag) call cpu_time (finish)
  !write(10,*)D,"  ", finish-start

  IF (Time_flag) call cpu_time(start_2)

  IF (Generic_flag) THEN
  DO i = 1,D**N
      N_body(i) = COMPLEX(RAND(),RAND())
  END DO
    norm_n_body = 0
    DO i = 1,D*N
      norm_n_body = norm_n_body + N_body(i) * CONJG(N_body(i))
    END DO
    N_body = N_body/SQRT(norm_n_body)
  ENDIF


  IF (Time_flag) call cpu_time (finish_2)
  IF (Time_flag) write(10,*)D,N, finish-start, finish_2 - start_2
```

If instead the TIME flag is not activated, the density matrix of the generic state is computed for a two body system

```
  IF (Time_flag .EQV. .FALSE.) THEN
    DO i = 1,D**2
      DO j = 1,D**2
        density(i,j) =  N_body(i)*CONJG(N_body(j))
      END DO
    END DO
  END IF
  IF (Time_flag .EQV. .FALSE.) CALL CHECKPOINT(D_FLAG,"After Allocation", D,N,wavefunction,N_body,density,subden_1,subden_2)
```

And the reduced density matrices for the two subsytems is then computed as shown in the theory section

```
IF (Time_flag .EQV. .FALSE.) THEN
    DO i = 1,D
      DO j = 1,D
        subden_2(i,j) = 0
        DO k = 1,D
          subden_2(i,j) =  subden_2(i,j) + density(i+D*(k-1),j+D*(k-1))
        END DO
      END DO
    END DO
    DO i = 1,D
      DO j = 1,D
        subden_1(i,j) = 0
        DO k = 1,D
          subden_1(i,j) =  subden_1(i,j) + density(k+D*(i-1),k+D*(j-1))
        END DO
      END DO
    END DO
END IF
```

Finally all the informations regarding the allocation time in case of the TIME flag or regarding the density matrices otherwise is printed to file

```
IF (Time_flag .EQV. .FALSE.) THEN
  write (10,*) "Separable state  ",wavefunction
  write (10,*) "Norm of separable state  ",SQRT(norm_wave)
  write (10,*) "Generic state  ",N_body
  write (10,*) "Norm of generic state  ",SQRT(norm_n_body)
  trace = 0
  DO i = 1,D**N
    trace = trace + density(i,i)
  END DO
  write (10,*) "Trace of density  ",trace
  write (10,*) "Density matrix    ",density
  trace = 0
  DO i = 1,D**N
    trace = trace + subden_1(i,i)
  END DO
  write (10,*) "Trace of density matrix of first system  ",trace
  write (10,*) "Density matrix of first system",subden_1
  trace = 0
  DO i = 1,D**N
    trace = trace + subden_2(i,i)
  END DO
  write (10,*) "Trace of density matrix of second system  ",trace
  write (10,*) "Density matrix of second system",subden_2
END IF
```

III. RESULTS

The code has been used to test the different initialization times for generic and separable states and in particular it was checked how the initialization times with N = 2 and increasing dimension D for Hilbert space of each body (Fig. 1) and in the case of qubits (D=2) and increasing N (Fig. 2). It is clear from the different scales on the x axis that separable states are much faster to initialize than generic states and that while for a separable state there is a symmetry between varying D and N (the number of elements is D*N), the scaling is much more sever for N in the generic case (the number of elements is D^N). Notice that in all cases the code stopped when crashing due to lack of memory. After fixing D = 2 and N = 2 the code has been tested to compute the reduced density matrices and in particular, initializing the a generic state with random complex values, the results computed were

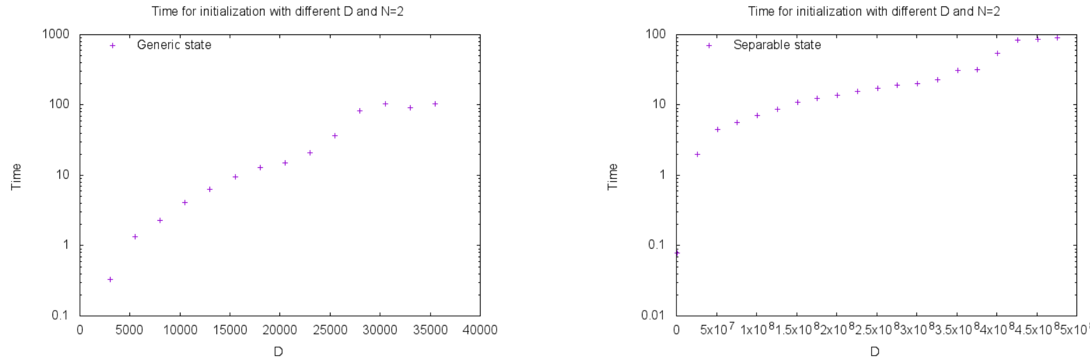

FIG. 1. Initialization time for a generic (left) and separable (right) state with N = 2 and varying D

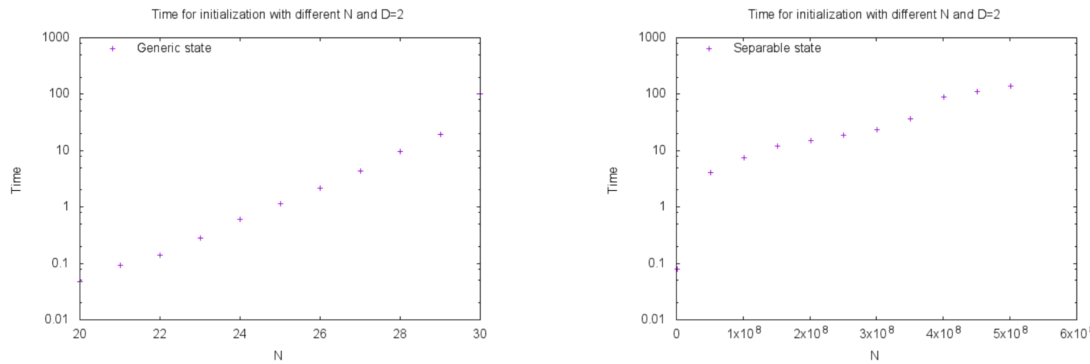

FIG. 2. Initialization time for a generic (left) and separable (right) state with D = 2 and varying N

$$\rho = \begin{pmatrix} 0.489 & 0.273 - i2.06E-003 & 0.218 - i0.276 & 0.194 + i0.113 \\ 0.273 + i2.060E-003 & 0.152 & 0.123 - i0.153 & 0.108 + i6.430E-002 \\ 0.218 + i0.276 & 0.123 + i0.153 & 0.253 & 2.300E-002 + i0.160 \\ 0.194 - i0.11 & 0.108 - i6.43E-002 & 2.30E-002 - i0.160 & 0.104 \end{pmatrix}$$

$$\rho_1 = \begin{pmatrix} 0.642 & 0.327 - i0.212 \\ 0.327 + i0.212 & 0.357 \end{pmatrix}$$

$$\rho_2 = \begin{pmatrix} 0.743 & 0.296 + i0.158 \\ 0.296 - i0.158 & 0.256 \end{pmatrix}$$

IV. SELF-EVALUATION

The difference between a generic and a separable state was tested by comparing the initialization times and it is clear how using separable states allows for significantly lower initialization times and memory resources needed. A code for computing the reduced density matrix of a system with 2 bodies and a generic dimension was implemented. The code also checks that the trace is equal to 1 for all the three density matrices. In the future the code could be generalized to system with N bodies and trying to check autonomously if the input state are separable or not